



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE FÍSICA Y MATEMÁTICAS

Reporte de servicio social

María de los Ángeles Isidro Pérez

**Algoritmos para calcular los determinantes  
de matrices de Toeplitz de banda**

Proyecto de investigación IPN-SIP número 20130633  
Propiedades espectrales de matrices y operadores de Toeplitz  
Director del proyecto: Dr. Egor Maximenko

---

México, D.F.  
Mayo de 2014.

# Índice

Reporte Global	3
<b>1. Matrices de Toeplitz y de banda</b>	<b>7</b>
1.1. Matrices generales de Toeplitz . . . . .	7
1.2. Matrices generales de banda . . . . .	8
1.3. Matrices de Toeplitz de banda . . . . .	9
1.4. Formatos para almacenar matrices . . . . .	10
1.5. Matrices de Toeplitz simétricas . . . . .	15
<b>2. Método de eliminación de Gauss</b>	<b>18</b>
2.1. Método de eliminación Gauss con pivotes diagonales . . . . .	18
2.2. Método de eliminación Gauss con pivotes diagonales para matrices de banda . . . . .	22
<b>3. Modificación del algoritmo de Levinson–Durbin</b>	<b>29</b>
3.1. Ecuaciones de Yule–Walker . . . . .	29
3.2. Explicación de la idea de Levinson, para $n = 4$ . . . . .	30
3.3. Solución general de las ecuaciones de Yule–Walker . . . . .	31
3.4. Algoritmo de Levinson–Durbin modificado . . . . .	35
<b>4. Fórmula de Baxter–Schmidt</b>	<b>38</b>
4.1. Serie recíproca . . . . .	38
4.2. Fórmula de Baxter–Schmidt . . . . .	38
<b>5. Método de Widom</b>	<b>42</b>
5.1. Fórmula de Widom . . . . .	42
<b>6. Comparación de métodos</b>	<b>45</b>
6.1. Generador de matrices aleatorias . . . . .	45
6.2. Comparación de métodos . . . . .	47
<b>7. Programas en el lenguaje <i>Wolfram Mathematica</i></b>	<b>48</b>
Referencias	65

---

## Reporte Global

### Justificación

Las *matrices de Toeplitz* son un tipo especial de matrices, donde cada una de sus diagonales corresponde a un mismo valor. Dentro de este tipo de matrices están también las matrices de *Toeplitz de banda*, las cuales poseen información significativa sólo en algunas diagonales (las cuales forman una banda), y el resto de las entradas de la matriz son nulas. Son las *matrices de Toeplitz de banda* las que abordamos con mayor interés en este trabajo.

Matrices de este tipo surgen en métodos numéricos para resolver problemas de frontera, en el análisis de procesos estocásticos estacionarios, en algunos modelos de mecánica cuántica y en el procesamiento de imágenes. Dados estos antecedentes de aplicaciones, nos interesa estudiar las propiedades de dichas matrices. En este trabajo se estudian varios métodos para calcular los determinantes de las matrices de Toeplitz de banda, métodos que son aplicables específicamente a este tipo de matrices y que son más eficientes respecto a los métodos tradicionales para el cálculo de determinantes. En el campo numérico y de investigación es de gran importancia eficientar los métodos conocidos, así como estudiar nuevos métodos; por ello nos enfocamos en estudiar métodos que logren una eficiencia mayor a los métodos clásicos.

### Objetivos

- ✦ Estudiar varios métodos para almacenar matrices de banda (no necesariamente de Toeplitz), y elegir un formato apropiado para cada tipo de matrices.
- ✦ Estudiar, modificar y programar el algoritmo de eliminación gaussiana para aplicarlo a matrices de banda y en el cálculo de sus determinantes.
- ✦ Estudiar las fórmulas recursivas de Levinson–Durbin.
- ✦ Modificar y programar el algoritmo de Levinson–Durbin para calcular los determinantes de matrices de Toeplitz de banda.
- ✦ Estudiar y programar la fórmula de Baxter–Schmidt para matrices de Toeplitz “casi triangulares”.
- ✦ Estudiar y programar la fórmula de Widom para calcular los determinantes de matrices de Toeplitz de banda.
- ✦ Calcular y comparar la complejidad teórica de los algoritmos programados.
- ✦ Comparar las funciones programadas para determinar el método más eficiente numéricamente.

## Marco Teórico

El análisis numérico y el álgebra, son áreas sumamente utilizadas y estudiadas, en este campo se requieren métodos nuevos que se especialicen y logren mayor eficiencia para ciertas matrices. Las matrices de Toeplitz tienen diversas aplicaciones numéricas, lo cual ha requerido el estudio de nuevos métodos para el cálculo de los determinantes de éstas, el objetivo de los nuevos métodos es ser más eficientes que los comúnmente utilizados, esto es, que realicen menos operaciones, empleen menos recursos o información y por supuesto que sean más rápidos en cuanto a tiempo. Cuando nos referimos a matrices Toeplitz de banda, hablamos de matrices cuya información relevante está concentrada en una lista de coeficientes, lo cual se puede aprovechar para ahorrar memoria al momento de almacenar estas matrices.

Los métodos estudiados en el presente trabajo, poseen ventajas respecto a los métodos más conocidos y utilizados para el cálculo de determinantes, como el método de Gauss y algunas variantes. El número de operaciones que ejecutan estos métodos son menores al número de operaciones empleadas por el método de Gauss, por ejemplo; y cada método puede funcionar mejor que los otros dependiendo de las matrices que empleemos.

El número de operaciones que realiza cada método depende de dos características: tamaño de la matriz  $n$  y ancho de la banda  $w$ . El método tradicional de Gauss para el cálculo de determinantes tiene una complejidad teórica de orden  $n^3$ , es decir, el número de operaciones que realiza es aproximadamente  $n^3$ ; el algoritmo modificado de Levinson-Durbin realiza  $nw + n^2$  operaciones, aproximadamente; la modificación del método de Gauss tiene una complejidad de  $nw^2$ ; la complejidad de la fórmula de Baxter-Schmidt es del orden de  $nw + w^3$ ; y finalmente la complejidad de la fórmula de Widom, a diferencia de los demás métodos aquí estudiados, no depende del tamaño de la matriz  $n$ , sólo depende del ancho de banda  $w$ , esta complejidad es aproximadamente  $2^w$ , este método, por ejemplo, tiene ventaja para matrices grandes con anchos de banda pequeños.

El algoritmo de Levinson-Durbin modificado para matrices de Toeplitz de banda simétricas utiliza relaciones recursivas con matrices Toeplitz. La fórmula de Baxter-Schmidt está basada en el teorema de Jacobi para los menores de la matriz adjunta clásica, aplicado a matrices de Toeplitz de banda. La fórmula de Widom combina ideas de variable compleja y combinatoria; expresa el determinante como una combinación lineal de  $w_M^n$  y  $C_M$  y su complejidad, como se ha mencionado, no depende del tamaño de matriz.

Las propiedades de las matrices estudiadas permiten utilizar las ventajas de cada método con respecto a otros más conocidos y con ello se logra una mayor eficiencia, utilizando un método especializado según el tipo de matriz. En el campo del análisis numérico se busca optimizar: lograr mayor eficiencia y utilizar la menor cantidad de recursos posible.

## Desarrollo

Para poder desarrollar los objetivos planteados requerí primeramente familiarizarme con los temas a desarrollar: estudié sobre ellos en los libros que se muestran como referencias al final del presente trabajo.

La primera parte que se llevó a cabo fue estudiar los métodos para almacenar matrices de banda; después elegí un formato adecuado para cada tipo de matrices, ya que estos formatos serían utilizados posteriormente para el desarrollo de los temas.

En la modificación del algoritmo de eliminación de Gauss, trabajé con el formato elegido para almacenar matrices de banda: el trabajo principal en este punto fue determinar las entradas en el formato elegido que se modifican al realizar cada una de las iteraciones del método. Posteriormente, después de aplicar el método de eliminación, determiné la posición en que se ubicaban las entradas que pertenecían a la diagonal principal, lo cual se requiere para calcular el determinante.

Para el método modificado del algoritmo de Levinson–Durbin, estudié primero el algoritmo original el cual sirve para resolver sistemas de ecuaciones lineales; utilicé la idea de este algoritmo para hacer la reducción de la matriz a una triangular, con lo cual el determinante de la matriz está dado por el producto de sus elementos en la diagonal principal.

La fórmula de Baxter–Schmidt consta de analizar dos partes: la primera consiste en los coeficientes de la serie recíproca, lo cual se programó antes del método; la segunda, consiste en aplicar dicha fórmula, lo cual requiere el cálculo de los coeficientes de la serie recíproca. Estudié también la deducción de la fórmula.

La fórmula de Widom resultó ser un método muy eficiente cuando se trata de matrices de banda pequeña: cuando la banda es muy grande el tiempo requerido para realizar el cálculo es superior a los otros métodos, contrario a lo que ocurre con bandas pequeñas. En este método requerí utilizar la función de precisión numérica para obtener resultados adecuados.

Realicé la comparación de los métodos estudiados y también de la función interna para calcular el determinante de *Wolfram Mathematica*, que es en donde desarrollé todos los programas. En este caso realicé una comparación numérica de eficiencia respecto al tiempo que tarda cada método. También realicé una comparación teórica.

Finalmente redacté los apuntes donde plasmo los resultados obtenidos, así como la metodología, algoritmos y programas realizados.

## Conclusiones

Realizar mi servicio social en un proyecto de investigación, en el que tenía que desarrollar un tema propio, me permitió aprender sobre: la estructura de un proyecto y una investigación, establecer objetivos, usar los objetivos como guía para desarrollar la investigación, analizar problemas, la utilización de métodos numéricos en el álgebra lineal y nuevos algoritmos.

El apoyo por parte del director del proyecto, fue de suma importancia en los temas que yo no dominaba totalmente, gracias a lo cual pude desarrollar los temas y programas de manera exitosa. En el proyecto de investigación han participado también varios estudiantes y, aunque se trabaja principalmente de forma independiente, los aportes y experiencia de los demás compañeros permiten mejorar el trabajo, como apoyo para instalar o utilizar un programa, cuestiones sobre los temas que han desarrollado y retroalimentación, lo que me permitió valorar la importancia de un equipo con ideas y conocimientos distintos a los míos.

Durante mi servicio adquirí nuevos conocimientos que me permitieron desarrollar satisfactoriamente la investigación, desde comandos que no conocía en el lenguaje *Wolfram*, así como mejorar mi capacidad de análisis, aprendí nuevos conceptos de álgebra y métodos numéricos, mejoré mi escritura de textos en  $\text{\LaTeX}$  y amplié los conocimientos que tenía. A través del estudio de textos científicos, libros y artículos, así como del análisis de los métodos estudiados, aprendí sobre algoritmos: cómo las ideas se convierten en algoritmos, los algoritmos en pro-

gramas y los programas en información; esta información puede ser utilizada para comparar y con ello se pueden apreciar las deducción teóricas, en este caso, sobre la eficiencia de los métodos. También realicé modificaciones a algoritmos para que realizaran tareas más específicas, con lo que se logró una mayor eficiencia.

Para el desarrollo del tema de investigación que realicé, requerí conocimientos previos de álgebra, combinatoria, métodos numéricos, análisis y series. Me resultó de gran ayuda conocer el lenguaje Wolfram y saber redactar textos en  $\text{\LaTeX}$ , estos conocimientos no son universales en la formación como ingenieros, sin embargo, son de suma importancia en el desarrollo de temas matemáticos.

Por medio de este trabajo quienes trabajan en investigación de matrices de Toeplitz, podrán revisar algunas ideas y algoritmos para el cálculo de determinantes y, los estudiantes de álgebra lineal y métodos numéricos podrán contar con un apoyo en su estudio, en la realización de programas, análisis de algoritmos y temas de álgebra lineal.

Este reporte está publicado en internet:

[http://esfm.egormaximenko.com/toeplitz\\_matrices.html](http://esfm.egormaximenko.com/toeplitz_matrices.html)

[http://esfm.egormaximenko.com/students/  
Isidro\\_Perez\\_2014\\_social\\_service\\_es.pdf](http://esfm.egormaximenko.com/students/Isidro_Perez_2014_social_service_es.pdf)

Espero que el trabajo sea útil en las investigaciones sobre matrices de Toeplitz y para los estudiantes que estudian álgebra lineal numérica.

# 1. Matrices de Toeplitz y de banda

En esta primera sección definiremos los tipos de matrices con las que trabajaremos, revisaremos formatos para almacenar estas matrices y estudiaremos los algoritmos para cambiar de un formato a otro.

## 1.1. Matrices generales de Toeplitz

**Definición 1.** Definimos a una *matriz de Toeplitz* de orden  $n$  o tamaño  $n \times n$   $T_n(a)$  de la siguiente forma:

$$T_n(a) = [a_{j-k}]_{j,k=1}^n = \begin{pmatrix} a_0 & a_{-1} & \dots & a_{-n+1} \\ a_1 & a_0 & \dots & a_{-n+2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \dots & a_0 \end{pmatrix}$$

**Ejemplo 1.** Una matriz de Toeplitz (o matriz de Toeplitz completa) de tamaño  $n = 6$  es de la siguiente forma

$$T_6(a) = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & a_{-3} & a_{-4} & a_{-5} \\ a_1 & a_0 & a_{-1} & a_{-2} & a_{-3} & a_{-4} \\ a_2 & a_1 & a_0 & a_{-1} & a_{-2} & a_{-3} \\ a_3 & a_2 & a_1 & a_0 & a_{-1} & a_{-2} \\ a_4 & a_3 & a_2 & a_1 & a_0 & a_{-1} \\ a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{pmatrix}.$$

Posteriormente estudiaremos formatos para almacenar este tipo de matrices.

**Nota:** En el presente trabajo se consideran únicamente matrices reales, es decir, matrices cuyas entradas son números reales.

## 1.2. Matrices generales de banda

Vamos a usar la siguiente terminología para las diagonales paralelas a la diagonal principal:

**Definición 2.** ( $q$ -ésima diagonal de una matriz cuadrada). Sea  $A \in \mathcal{M}_{n \times n}(\mathbb{R})$  y sea  $q \in \{-n + 1, \dots, n - 1\}$ . Si  $j, k \in \{1, \dots, n\}$  y  $j - k = q$ , entonces decimos que la entrada  $(j, k)$  está en la  $q$ -ésima diagonal de la matriz  $A$ .

Si en una matriz cuadrada todas las entradas no nulas están en las diagonales  $-r, -r + 1, \dots, s$ , con  $r, s < n - 1$  entonces se dice que la matriz es *de banda*.

**Definición 3.** Se dice que una matriz  $A$  es *de banda* si  $A_{j,k} = 0$  cuando  $j - k \leq -r$  o bien  $j - k \leq s$ . Aquí  $r$  representa el *semiancho superior*, es decir el número de diagonales no triviales sobre la diagonal principal, y  $s$  representa el *semiancho inferior*, es decir, el número de diagonales por debajo de la diagonal principal. El número  $w = r + s + 1$  es el *ancho* (del inglés *width*) de la banda.

**Ejemplo 2.** La siguiente matriz corresponde a  $n = 5, r = 1, s = 2$ :

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & 0 & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} & 0 \\ 0 & A_{4,2} & A_{4,3} & A_{4,4} & A_{4,5} \\ 0 & 0 & A_{5,3} & A_{5,4} & A_{5,5} \end{pmatrix}.$$

En este caso el ancho de banda es  $w = 4$ .

También las matrices de banda se guardarán en un formato especial: sólo guardaremos las entradas de la banda, puesto que el resto son ceros. De esta manera se ahorra espacio (memoria) y tiempo para crear estas matrices.

Por ejemplo, la matriz del Ejemplo 2 se guardaría como:

$$B = \begin{pmatrix} A_{1,1} & A_{1,2} & & & \\ A_{2,1} & A_{2,2} & A_{2,3} & & \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} & \\ A_{4,2} & A_{4,3} & A_{4,4} & A_{4,5} & \\ A_{5,3} & A_{5,4} & A_{5,5} & & \end{pmatrix}.$$

### 1.3. Matrices de Toeplitz de banda

**Definición 4.** Sea  $b$  un polinomio de Laurent:

$$b(t) = \sum_{j=-r}^s b_j t^j, \quad r, s \in \mathbb{N} \cup \{0\}.$$

Una *matriz de Toeplitz de banda generada por un polinomio de Laurent*, es una matriz de Toeplitz con  $b_{j-k} = 0$  cuando  $j - k < -r$  o bien  $j - k > s$ , y los coeficientes del polinomio de Laurent son los elementos en la banda  $b_s \dots b_0 \dots b_{-r}$ , donde  $r$  representa el semiancho superior (número de diagonales sobre la diagonal principal) y  $s$  representa el semiancho inferior (número de diagonales bajo la diagonal principal), con  $r, s < n - 1$ , donde  $n$  es el tamaño de la matriz. La denotamos por  $T_n(b)$  y tiene la siguiente forma:

$$T_n(b) = \begin{pmatrix} b_0 & b_{-1} & \dots & b_{-r} & 0 & \dots \\ b_1 & b_0 & \dots & b_{-r+1} & b_{-r} & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ b_s & b_{s-1} & \dots & b_0 & b_{-1} & \dots \\ 0 & b_s & \dots & b_1 & b_0 & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{pmatrix}$$

**Ejemplo 3.** Sea  $b(t) = b_{-3}t^{-3} + b_{-2}t^{-2} + b_{-1}t^{-1} + b_0 + b_1t^1 + b_2t^2$ , donde  $r = 3$  y  $s = 2$ , para  $n = 6$  entonces,

$$T_6(b) = \begin{pmatrix} b_0 & b_{-1} & b_{-2} & b_{-3} & 0 & 0 \\ b_1 & b_0 & b_{-1} & b_{-2} & b_{-3} & 0 \\ b_2 & b_1 & b_0 & b_{-1} & b_{-2} & b_{-3} \\ 0 & b_2 & b_1 & b_0 & b_{-1} & b_{-2} \\ 0 & 0 & b_2 & b_1 & b_0 & b_{-1} \\ 0 & 0 & 0 & b_2 & b_1 & b_0 \end{pmatrix}.$$

## 1.4. Formatos para almacenar matrices

Estos tipos de matrices se guardarán en un formato especial, donde lo único que se almacena son los coeficientes del polinomio de Laurent o coeficientes de la banda. En el caso de la matriz general de Toeplitz requerimos la información de los coeficientes  $a_{-n+1} \dots a_{n-1}$ . En el ejemplo 1, se almacena únicamente la lista de los coeficientes

$$a = ( a_{-5} \ a_{-4} \ a_{-3} \ a_{-2} \ a_{-1} \ a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 )$$

Con esta lista de coeficientes se puede generar la matriz completa de tamaño  $6 \times 6$ . La matriz del ejemplo 4 se guarda de la siguiente forma:

$$b = ( b_{-3} \ b_{-2} \ b_{-1} \ b_0 \ b_1 \ b_2 ),$$

**Nota:** Este segundo formato requiere indicar también el tamaño de la matriz  $n$  y el semiancho de banda  $r$ , puesto que el mismo polinomio puede generar matrices de distintos tamaños, en este caso  $r = 3$  y  $n = 6$ . En el caso de una matriz de Toeplitz completa basta con conocer la lista de coeficientes, ya que con ésta se puede calcular el tamaño de la matriz.

Usaremos la siguiente notación:

$T$ : Formato de Toeplitz.

$TB$ : Formato Toeplitz de banda.

$B$ : Formato de banda.

$M$ : Formato de matriz general.

$S$ : Indica que se está trabajando con matrices simétricas.

La idea del algoritmo 1 se puede observar en la matriz del ejemplo 1. Los coeficientes en cada diagonal son iguales por lo tanto basta con almacenar estos elementos, ya que con ellos se puede generar una matriz general o de Toeplitz completa.

$$T = ( a_{-5} \ a_{-4} \ a_{-3} \ a_{-2} \ a_{-1} \ a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 ) \mapsto M = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & a_{-3} & a_{-4} & a_{-5} \\ a_1 & a_0 & a_{-1} & a_{-2} & a_{-3} & a_{-4} \\ a_2 & a_1 & a_0 & a_{-1} & a_{-2} & a_{-3} \\ a_3 & a_2 & a_1 & a_0 & a_{-1} & a_{-2} \\ a_4 & a_3 & a_2 & a_1 & a_0 & a_{-1} \\ a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{pmatrix}.$$

Posteriormente podemos ver la relación que tienen los elementos de  $M$  con respecto a los elementos de  $T$ .

$$M_{1,1} = T_6 = a_0 \quad M_{4,4} = T_6 = a_0 \quad M_{3,1} = T_8 = a_2 \quad M_{6,4} = T_8 = a_2$$

Esta relación se puede establecer de la siguiente forma:

$$M_{j,k} = T_{n+j-k} = a_{j-k}$$

Notemos también que  $n$ , el tamaño de la matriz, se puede determinar a partir de la longitud de  $T$ , dado que  $\text{longitud}(T) = 2n - 1$ . Por lo tanto, para almacenar matrices de Toeplitz, basta con guardar las entradas de la primera fila y la primera columna.

---

**Algoritmo 1**  $T \rightarrow M$ 


---

**Entrada:**  $a$ : coeficientes del polinomio de Laurent o coeficientes de la matriz de Toeplitz;

**Salida:**  $M$ : matriz de Toeplitz guardada en el formato general;

- 1:  $l \leftarrow$  longitud de  $a$ ;
  - 2:  $n \leftarrow (l + 1)/2$ ;
  - 3:  $M \leftarrow$  matriz de ceros  $n \times n$ ;
  - 4: **para**  $j = 1 \dots n$  **hacer**
  - 5:     **para**  $k = 1 \dots n$  **hacer**
  - 6:          $M[j, k] \leftarrow a[n + j - k]$ ;
  - 7:     **fin para**
  - 8: **fin para**
- 

El Algoritmo 1 hace la transformación de la lista de coeficientes de una matriz de Toeplitz a una matriz general.

La idea del Algoritmo 2 se muestra con el siguiente esquema (numeramos todos los índices a partir de 1):

$$b = ( b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 ), \quad r = 3, \quad n = 6$$

$$\mapsto c = \left[ \underbrace{c_1}_0 \ \underbrace{c_2}_0 \ \underbrace{c_3}_{b_1} \ \underbrace{c_4}_{b_2} \ \underbrace{c_5}_{b_3} \ \underbrace{c_6}_{b_4} \ \underbrace{c_7}_{b_5} \ \underbrace{c_8}_{b_6} \ \underbrace{c_9}_0 \ \underbrace{c_{10}}_0 \ \underbrace{c_{11}}_0 \right]$$

$$\mapsto M = \begin{pmatrix} c_6 & c_5 & c_4 & c_3 & c_2 & c_1 \\ c_7 & c_6 & c_5 & c_4 & c_3 & c_2 \\ c_8 & c_7 & c_6 & c_5 & c_4 & c_3 \\ c_9 & c_8 & c_7 & c_6 & c_5 & c_4 \\ c_{10} & c_9 & c_8 & c_7 & c_6 & c_5 \\ c_{11} & c_{10} & c_9 & c_8 & c_7 & c_6 \end{pmatrix}.$$

Este formato Toeplitz banda se transforma en el formato de una matriz completa, dado  $r = 3$  y  $n = 6$ , queda de la siguiente forma:

$$b = ( b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 ) \quad \mapsto \quad M = \begin{pmatrix} b_4 & b_3 & b_2 & b_1 & 0 & 0 \\ b_5 & b_4 & b_3 & b_2 & b_1 & 0 \\ b_6 & b_5 & b_4 & b_3 & b_2 & b_1 \\ 0 & b_6 & b_5 & b_4 & b_3 & b_2 \\ 0 & 0 & b_6 & b_5 & b_4 & b_3 \\ 0 & 0 & 0 & b_6 & b_5 & b_4 \end{pmatrix}$$

Notemos que el algoritmo 2 consiste en convertir el formato Toeplitz banda al formato Toeplitz y posteriormente aplicar el algoritmo 1.

---

**Algoritmo 2** TB  $\rightarrow$  M
 

---

**Entrada:**  $b$ : coeficientes del polinomio de Laurent;  $r$ : semiancho superior de la banda;  $n$ : tamaño de la matriz;

**Salida:**  $T$ : matriz de Toeplitz de banda guardada en el formato general M;

```

1:  $l \leftarrow$  longitud de  $b$ ;
2:  $c \leftarrow$  lista de ceros de tamaño  $2n - 1$ ;
3: para  $i = 1 \dots l$  hacer
4:    $p \leftarrow n - r - 1 + i$ ;
5:    $c[p] = b[i]$ ;
6: fin para
7:  $T \leftarrow$  matriz de ceros  $n \times n$ ;
8: para  $j = 1 \dots n$  hacer
9:   para  $k = 1 \dots n$  hacer
10:     $T[j, k] \leftarrow c[n + j - k]$ ;
11:  fin para
12: fin para

```

---

La función del Algoritmo 3 puede observarse en el siguiente ejemplo:

**Ejemplo 4.** Dada una matriz  $7 \times 7$  en el formato de banda, con  $r = 3$  y  $s = 1$ , construimos la matriz completa:

$$B = \begin{pmatrix} B_{1,1} & B_{1,2} & B_{1,3} & B_{1,4} & & & \\ B_{2,1} & B_{2,2} & B_{2,3} & B_{2,4} & B_{2,5} & & \\ B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & B_{3,5} & & \\ B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & B_{4,5} & & \\ B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} & & & \\ B_{6,1} & B_{6,2} & B_{6,3} & & & & \\ B_{7,1} & B_{7,2} & & & & & \end{pmatrix} \mapsto A = \begin{pmatrix} B_{1,1} & B_{1,2} & B_{1,3} & B_{1,4} & 0 & 0 & 0 \\ B_{2,1} & B_{2,2} & B_{2,3} & B_{2,4} & B_{2,5} & 0 & 0 \\ 0 & B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & B_{3,5} & 0 \\ 0 & 0 & B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & B_{4,5} \\ 0 & 0 & 0 & B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} \\ 0 & 0 & 0 & 0 & B_{6,1} & B_{6,2} & B_{6,3} \\ 0 & 0 & 0 & 0 & 0 & B_{7,1} & B_{7,2} \end{pmatrix}$$

En este caso, vemos que el arreglo  $B$  se puede transformar en la matriz  $A$ , por ejemplo:

$$A_{6,2} = 0, \quad A_{6,6} = B_{6,2}.$$

$A$  resulta de aplicarle una transformación a  $B$ . En el siguiente algoritmo se establece la transformación del arreglo  $B$  a la matriz  $A$ .

**Algoritmo 3**  $B \rightarrow M$ **Entrada:**  $B$  : matriz dada en forma de banda.**Salida:**  $A$  : matriz completa.

- 1:  $n \leftarrow$  número de filas de  $B$ ;
- 2:  $A \leftarrow$  matriz de ceros  $n \times n$ ;
- 3: **para**  $i = 1 \dots n$  **hacer**
- 4:    $a \leftarrow \min\{1, i - s\}$ ;
- 5:    $b \leftarrow \min\{i + r, n\}$ ;
- 6:    $A[i, a : b] \leftarrow B[i, *]$ ;
- 7: **fin para**

En el Algoritmo 3 la notación  $B[i, *]$  significa toda la fila  $i$  del arreglo  $B$ .

**Observación 1.** Estos formatos, para guardar matrices de banda y matrices de Toeplitz de banda, son utilizados para los algoritmos que estudiamos. Por ello es importante también convertir el formato de Toeplitz banda a formato de banda.

**Ejemplo 5.** Para una matriz de Toeplitz de banda, con  $r = 2$ ,  $s = 4$  y  $n = 8$ , se representa a continuación la transformación de formato de Toeplitz de banda a formato de banda,

$$b = \begin{pmatrix} b_{-2} & b_{-1} & b_0 & b_1 & b_2 & b_3 & b_4 \end{pmatrix} \mapsto B = \begin{pmatrix} b_0 & b_{-1} & b_{-2} & & & & & \\ b_1 & b_0 & b_{-1} & b_{-2} & & & & \\ b_2 & b_1 & b_0 & b_{-1} & b_{-2} & & & \\ b_3 & b_2 & b_1 & b_0 & b_{-1} & b_{-2} & & \\ b_4 & b_3 & b_2 & b_1 & b_0 & b_{-1} & b_{-2} & \\ b_4 & b_3 & b_2 & b_1 & b_0 & b_{-1} & b_{-2} & \\ b_4 & b_3 & b_2 & b_1 & b_0 & b_{-1} & & \\ b_4 & b_3 & b_2 & b_1 & b_0 & & & \end{pmatrix}$$

En el ejemplo 5 observamos la transformación del formato Toeplitz banda al formato de Banda, esta transformación se establece en el Algoritmos 4.

---

**Algoritmo 4**  $TB \rightarrow B$ .

---

**Entrada:**  $b$  : coeficientes del polinomio de Laurent;  $n$ : tamaño de la matriz;  $r$ : semiancho de banda;

**Salida:**  $B$  : Matriz de banda;

1:  $l \leftarrow$  longitud de  $b$ ;

2:  $B \leftarrow$  matriz de  $n$  arreglos.

3: **para**  $i = 1 \dots n$  **hacer**

4:    $c \leftarrow \max\{1, i + r + 1 - n\}$ ;

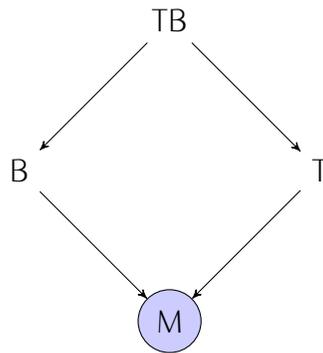
5:    $d \leftarrow \min\{l, r + 1\}$ ;

6:    $B[i] = b[d : c : -1]$ ; // el arreglo  $b[c : d]$  invertido

7: **fin para**

---

En el siguiente esquema observamos la forma en que se pueden transformar los formatos, las flechas nos indican cómo se pueden efectuar estas transformaciones.



## 1.5. Matrices de Toeplitz simétricas

Otro caso es el de las matrices simétricas, específicamente las matrices de Toeplitz simétricas o bien las matrices de Toeplitz simétricas de banda. En estos caso al ser simétricas, basta con almacenar los elementos de la banda en la primera fila, como podemos observar en los siguientes ejemplos.

**Ejemplo 6.** Sea una matriz de Toeplitz simétrica de tamaño  $n = 6$ . Podemos almacenar la información suficiente de la matriz utilizando solamente la información de la primera fila, o bien, la primera columna, ya que el resto de la matriz puede generarse con esta información.

$$TS = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{pmatrix} \mapsto M = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\ a_2 & a_1 & a_2 & a_3 & a_4 & a_5 \\ a_3 & a_2 & a_1 & a_2 & a_3 & a_4 \\ a_4 & a_3 & a_2 & a_1 & a_2 & a_3 \\ a_5 & a_4 & a_3 & a_2 & a_1 & a_2 \\ a_6 & a_4 & a_3 & a_2 & a_1 & a_2 \end{pmatrix}$$

---

**Algoritmo 5**  $TS \rightarrow M$ .

---

**Entrada:**  $a_f$  : elementos de la primera fila;

**Salida:**  $M$  : Matriz completa;

- 1:  $n \leftarrow$  longitud de  $a_f$ ;
  - 2:  $long \leftarrow 2n - 1$ ;
  - 3:  $a \leftarrow$  lista de ceros de tamaño  $long$ ;
  - 4:  $a[n + 1 : long] \leftarrow a_f[n - 1 : 1 : -1]^*$ ;
  - 5:  $a[1 : n] \leftarrow a_f$ ;
  - 6:  $M \leftarrow$  matriz de ceros de tamaño  $n \times n$ ;
  - 7: **para**  $j = 1 \dots n$  **hacer**
  - 8:   **para**  $k = 1 \dots n$  **hacer**
  - 9:      $M[j, k] \leftarrow a[n + j - k]$ ;
  - 10:   **fin para**
  - 11: **fin para**
- 

\* La notación  $a_f[n - 1 : 1 : -1]$  representa los  $n - 1$  primeros elementos de la lista  $a_f$  empezando por el elemento  $n - 1$  y terminado con el primer elemento.

**Ejemplo 7.** Sea una matriz de Toeplitz simétrica de banda de tamaño  $n = 7$ , con  $r = 2$ . En este caso tenemos que la información relevante de la matriz, la que nos interesa almacenar, son las tres entradas significativas y el tamaño de la matriz. Esa información es suficiente para generar la matriz completa.

$$TBS = (b_1 \ b_2 \ b_3) \mapsto M = \begin{pmatrix} b_1 & b_2 & b_3 & 0 & 0 & 0 & 0 \\ b_2 & b_1 & b_2 & b_3 & 0 & 0 & 0 \\ b_3 & b_2 & b_1 & b_2 & b_3 & 0 & 0 \\ 0 & b_3 & b_2 & b_1 & b_2 & b_3 & 0 \\ 0 & 0 & b_3 & b_2 & b_1 & b_2 & b_3 \\ 0 & 0 & 0 & b_3 & b_2 & b_1 & b_2 \end{pmatrix}$$

El algoritmo 6 realiza la transformación de formato TBS al fomato de matriz general  $M$ .

---

**Algoritmo 6** TBS  $\rightarrow$   $M$ .

---

**Entrada:**  $b_f$  : elementos de la banda en la primera fila;  $n$  : tamaño de la matriz;

**Salida:**  $M$  : Matriz completa;

- 1:  $r \leftarrow$  longitud de  $b_f - 1$ ;
  - 2:  $long \leftarrow 2n - 1$ ;
  - 3:  $b \leftarrow$  lista de ceros de tamaño  $long$ ;
  - 4:  $b[n + r : n] \leftarrow b_f$ ;
  - 5:  $b[n : n + r] \leftarrow \text{reverso}[b_f]$ ;
  - 6:  $M \leftarrow$  matriz de ceros de tamaño  $n \times n$ ;
  - 7: **para**  $j = 1 \dots n$  **hacer**
  - 8:     **para**  $k = 1 \dots n$  **hacer**
  - 9:          $M[j, k] \leftarrow b[n + j - k]$ ;
  - 10:     **fin para**
  - 11: **fin para**
- 

También nos interesa convertir el formato TBS al formato de banda, esto lo hacemos aplicando a la tranformación de los algoritmos anteriores el algoritmo 4.

**Ejemplo 8.** Para una matriz de Toeplitz de banda, de tamaño  $n = 8$  y con  $r = 3$ . En este caso podemos generar la matriz en formato de Banda, como se muestra a continuación. Sólo se requiere la información de los elementos en la parte superior de la banda y el tamaño de la matriz.

$$\text{TBS} = \begin{pmatrix} b_0 & b_1 & b_2 & b_3 \end{pmatrix} \mapsto B = \begin{pmatrix} b_0 & b_1 & b_2 & b_3 & & & & \\ b_1 & b_0 & b_1 & b_2 & b_3 & & & \\ b_2 & b_1 & b_0 & b_1 & b_2 & b_3 & & \\ b_3 & b_2 & b_1 & b_0 & b_1 & b_2 & b_3 & \\ b_3 & b_2 & b_1 & b_0 & b_1 & b_2 & b_3 & \\ b_3 & b_2 & b_1 & b_0 & b_1 & b_2 & & \\ b_3 & b_2 & b_1 & b_0 & b_1 & & & \\ b_3 & b_2 & b_1 & b_0 & & & & \end{pmatrix}$$

El algoritmo 7 realiza esta transformación.

---

**Algoritmo 7** TBS  $\rightarrow$  B.

**Entrada:**  $b_f$  : elementos de la banda en la primera fila;  $n$  : tamaño de la matriz;

**Salida:**  $B$  : Matriz de banda;

- 1:  $r \leftarrow$  longitud de  $b_f - 1$ ;
  - 2:  $long \leftarrow 2r + 1$ ;
  - 3:  $b \leftarrow$  lista de ceros de tamaño  $long$ ;
  - 4:  $b[r + 2 : long] \leftarrow [b_f[r : 1 : -1]]$ ;
  - 5:  $b[1 : r + 1] \leftarrow b_f$ ;
  - 6:  $B \leftarrow$  matriz de  $n$  arreglos;
  - 7: **para**  $i = 1 \dots n$  **hacer**
  - 8:    $c \leftarrow \max\{1, i + r + 1 - n\}$ ;
  - 9:    $d \leftarrow \min\{l, r + 1\}$ ;
  - 10:    $B[i] = b[d : c : -1]$ ;
  - 11: **fin para**
- 

Una vez que conocemos los formatos que utilizaremos en este trabajo, podemos comenzar el análisis de los métodos para calcular los determinantes de matrices de Toeplitz de banda.

Se podrá notar en cada tema que el algoritmo utiliza alguno de los formatos que estudiamos en este primer capítulo. Y para realizar la comparación de métodos se usa la transformación de un formato a otro, con la finalidad de utilizar las mismas matrices con cada uno de los métodos.

## 2. Método de eliminación de Gauss

El método de eliminación de Gauss, es un método muy conocido para resolver sistemas de ecuaciones lineales, este método realiza una transformación a una matriz inicial a una matriz triangular, después de realizar esta transformación es sencillo calcular su determinante como el producto de sus elementos en la diagonal. En esta sección estudiaremos este método, específicamente la eliminación de Gauss con pivotes diagonales y realizaremos modificaciones al algoritmo, para obtener una mayor eficiencia. Para matrices de banda este método se puede optimizar utilizando el formato de banda, visto en la sección anterior.

### 2.1. Método de eliminación Gauss con pivotes diagonales

Primero recordemos el algoritmo de Gauss con pivotes diagonales para matrices completas (no de banda):

---

**Algoritmo 8** Gauss con pivotes diagonales

---

**Entrada:**  $A$ : matriz completa;

**Salida:**  $G$ : matriz reducida;

- 1:  $n \leftarrow$  número de filas de  $A$ ;
  - 2:  $G \leftarrow$  una copia de  $A$ ;
  - 3: **para**  $p = 1 \dots n - 1$  **hacer**
  - 4:   **para**  $i = p + 1 \dots n$  **hacer**
  - 5:      $f \leftarrow -G[i, p]/G[p, p]$ ;
  - 6:      $G[i, p] \leftarrow 0$ ;
  - 7:      $G[i, p + 1 : n] \leftarrow G[i, p + 1 : n] + f * G[p, p + 1 : n]$ ;
  - 8:   **fin para**
  - 9: **fin para**
- 

Para calcular el determinante de una matriz utilizando el algoritmo 8, se puede usar el siguiente algoritmo.

---

**Algoritmo 9** Gauss con pivotes diagonales para calcular el determinante.

---

**Entrada:**  $A$ : matriz completa;

**Salida:** det: determinante de  $A$ ;

- 1:  $n \leftarrow$  número de filas de  $A$ ;
  - 2:  $R \leftarrow$  matriz  $A$  reducida con eliminación de Gauss;
  - 3:  $\text{det} \leftarrow \prod_{i=1}^n R[i, i]$ ;
-

Si denotamos por  $G^{(p)}$  a la matriz obtenida después de  $p$  pasos del algoritmo, entonces el algoritmo anterior se puede escribir matemáticamente de la siguiente manera:

$$G^{(0)} = A;$$

$$G_{i,p}^{(p)} = 0 \quad (1 \leq p \leq n-1, p+1 \leq i \leq n);$$

$$f_{p,i} = -\frac{G_{i,p}^{(p-1)}}{G_{p,p}^{(p-1)}} \quad (1 \leq p \leq n-1, p+1 \leq i \leq n);$$

$$G_{i,j}^{(p)} = G_{i,j}^{(p-1)} + f_{p,j} G_{p,j}^{(p-1)} \quad (1 \leq p \leq n-1, p+1 \leq i \leq n, p+1 \leq j \leq n).$$

**Ejemplo 9.** Hay matrices para las cuales este algoritmo no se puede aplicar, por ejemplo

$$A = \begin{pmatrix} 1 & -2 & 1 \\ 1 & -2 & 2 \\ 1 & -2 & 3 \end{pmatrix}.$$

Después del primer paso del algoritmo de Gauss obtenemos la matriz

$$G^{(1)} = \begin{pmatrix} 1 & -2 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \end{pmatrix},$$

en la cual la entrada  $(2, 2)$  es igual a cero.

Hay varias modificaciones del algoritmo de eliminación de Gauss que usan estrategias de *pivoteo* e *intercambio de renglones*, pero en este texto nos restringimos al algoritmo de eliminación de Gauss con pivotes diagonales (sin pivoteo).

Para caracterizar las matrices cuadradas a las cuales se puede aplicar el algoritmo anterior (algoritmo 8), introducimos el siguiente concepto.

**Definición 5** (menores de esquina). Sea  $A \in \mathcal{M}_{n \times n}(\mathbb{R})$ . Entonces para cada  $p \in \{1, \dots, n\}$  denotemos por  $\Delta_p(A)$  al menor de  $A$  ubicado en la intersección de los primeros  $k$  renglones con las primeras  $k$  columnas:

$$\Delta_k(A) := \det(A[1:p, 1:p]).$$

Además, pongamos

$$\Delta_0(A) := 1.$$

Los menores de esta forma se llaman *menores de esquina* o *menores principales líderes*.

**Teorema 1.** Sea  $A \in \mathcal{M}_{n \times n}(\mathbb{R})$ . Entonces las siguientes condiciones son equivalentes:

- (a) Para cada  $p \in \{1, \dots, n-1\}$ ,  $\Delta_p(A) \neq 0$ .
- (b) A la matriz  $A$  se puede aplicar el algoritmo de eliminación de Gauss con pivotes diagonales, es decir, para cada  $p \in \{1, \dots, n-1\}$  después de aplicar  $p-1$  pasos del algoritmo la entrada  $(p, p)$  es distinta de cero.

*Idea de la demostración.* Supongamos que se cumple (a) y demostremos (b) por inducción sobre  $p$ . Supongamos que el algoritmo ya realizó los primeros  $p-1$  pasos, denotemos la matriz obtenida por  $G^{(p-1)}$  y demostremos que  $G_{p,p}^{(p-1)} \neq 0$ .

Se sabe que al sumar a un renglón de la matriz cualquier otro renglón multiplicado por un escalar, el determinante de la matriz no se cambia. Por eso

$$\Delta_p(A) = \Delta_p(G^{(p-1)}). \quad (1)$$

El siguiente esquema muestra la situación en el caso particular  $n = 5$ ,  $p = 3$ . La submatriz de  $G^{(2)}$  sombreada con color verde se obtiene de la submatriz correspondiente de  $A$  al aplicar ciertas operaciones elementales que no cambian el valor del determinante. Por eso  $\Delta_3(A) = \Delta_3(G^{(2)})$ .

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} & A_{2,5} \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} & A_{3,5} \\ A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} & A_{4,5} \\ A_{5,1} & A_{5,2} & A_{5,3} & A_{5,4} & A_{5,5} \end{pmatrix} \xrightarrow{\text{dos pasos del Algoritmo 8}} G^{(2)} = \begin{pmatrix} G_{1,1}^{(2)} & G_{1,2}^{(2)} & G_{1,3}^{(2)} & G_{1,4}^{(2)} & G_{1,5}^{(2)} \\ 0 & G_{2,2}^{(2)} & G_{2,3}^{(2)} & G_{2,4}^{(2)} & G_{2,5}^{(2)} \\ 0 & 0 & G_{3,3}^{(2)} & G_{3,4}^{(2)} & G_{3,5}^{(2)} \\ 0 & 0 & G_{4,3}^{(2)} & G_{4,4}^{(2)} & G_{4,5}^{(2)} \\ 0 & 0 & G_{5,3}^{(2)} & G_{5,4}^{(2)} & G_{5,5}^{(2)} \end{pmatrix}.$$

Por otro lado, la submatriz  $G^{(p-1)}[1 : p, 1 : p]$  es triangular superior, por eso su determinante es el producto de  $G_{1,1}^{(p-1)}, \dots, G_{p,p}^{(p-1)}$ . Combinando este hecho con la fórmula (1) obtenemos que

$$\Delta_p(A) = G_{1,1}^{(p-1)} \cdots G_{p,p}^{(p-1)} \quad (2)$$

Por la hipótesis sabemos que  $\Delta_p(A) \neq 0$ , por lo tanto el lado derecho de (2) todos los factores son distintos de cero, en particular,  $G_{p,p}^{(p-1)}$ .

La implicación (b) $\Rightarrow$ (a) se sigue de la fórmula (2). □

En toda esta sección suponemos que las matrices satisfacen las condiciones del Teorema 1.

Supongamos que la matriz original está dada en el formato de banda y modificamos el algoritmo 8 de manera que funcione en el formato de banda, sin construir la matriz completa. En el caso de que la matriz esté dada en formato de banda, como vimos en el capítulo anterior, no se tendrá que construir la matriz completa.

La idea de la modificación del algoritmo de Gauss es trabajar con la matriz dada en el formato de banda, es decir, no necesitamos la matriz completa para realizar el algoritmo, dado que la matriz completa tiene información no relevante para el cálculo del determinante. Al no realizar operaciones innecesarias utilizando los ceros, esta modificación del algoritmo resulta más eficiente que el algoritmo tradicional.

## 2.2. Método de eliminación Gauss con pivotes diagonales para matrices de banda

**Ejemplo 10.** Sea  $B$  una matriz dada en formato de banda, de tamaño  $n = 5$  y ancho de banda  $w = 4$ , con  $r = 1$  y  $s = 2$ , realizamos las primeras dos iteraciones del método de eliminación de Gauss.

$$B = \begin{pmatrix} B_{1,1} & B_{1,2} & & & \\ B_{2,1} & B_{2,2} & B_{2,3} & & \\ B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & \\ B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & \\ B_{5,1} & B_{5,2} & B_{5,3} & & \end{pmatrix} \xrightarrow{\text{Equivalente a}} \begin{pmatrix} B_{1,1} & B_{1,2} & 0 & 0 & 0 \\ B_{2,1} & B_{2,2} & B_{2,3} & 0 & 0 \\ B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & 0 \\ 0 & B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} \\ 0 & 0 & B_{5,1} & B_{5,2} & B_{5,3} \end{pmatrix}$$

**Primera iteración:** Eliminación en la primera columna. Pivote:  $B_{1,1}$

$$\begin{pmatrix} B_{1,1} & B_{1,2} & & & \\ 0 & B'_{2,2} & B_{2,3} & & \\ 0 & B'_{3,2} & B_{3,3} & B_{3,4} & \\ B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & \\ B_{5,1} & B_{5,2} & B_{5,3} & & \end{pmatrix} \xrightarrow{\text{Equivalente a}} \begin{pmatrix} B_{1,1} & B_{1,2} & 0 & 0 & 0 \\ 0 & B'_{2,2} & B_{2,3} & 0 & 0 \\ 0 & B'_{3,2} & B_{3,3} & B_{3,4} & 0 \\ 0 & B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} \\ 0 & 0 & B_{5,1} & B_{5,2} & B_{5,3} \end{pmatrix}$$

**Segunda iteración:** Eliminación en la segunda columna. Pivote:  $B'_{2,2}$

$$\begin{pmatrix} B_{1,1} & B_{1,2} & & & \\ 0 & B'_{2,2} & B_{2,3} & & \\ 0 & 0 & B'_{3,3} & B_{3,4} & \\ 0 & B'_{4,2} & B_{4,3} & B_{4,4} & \\ B'_{5,1} & B_{5,2} & B_{5,3} & & \end{pmatrix} \xrightarrow{\text{Equivalente a}} \begin{pmatrix} B_{1,1} & B_{1,2} & 0 & 0 & 0 \\ 0 & B'_{2,2} & B_{2,3} & 0 & 0 \\ 0 & 0 & B'_{3,3} & B_{3,4} & 0 \\ 0 & 0 & B'_{4,2} & B_{4,3} & B_{4,4} \\ 0 & 0 & B'_{5,1} & B_{5,2} & B_{5,3} \end{pmatrix}$$

Como podemos observar en el ejemplo, el algoritmo de Gauss sin modificación (derecha), realiza operaciones en todos los elementos de la columna aunque estos sean ceros al finalizar cada iteración, se realizan operaciones que pueden ser evitadas, ya que no alteran el resultado final. El algoritmo modificado de Gauss (izquierda), por otra parte, sólo incluye a las entradas significativas.

Ahora nos interesa conocer cómo realizar esta modificación al algoritmo. La cual básicamente consiste en cambiar los índices de las entradas que se modifican, por los adecuados para el formato de banda.

Consideremos lo siguiente:

1. El primer ciclo en el algoritmo de Gauss, permanece igual.
2. El segundo ciclo sí se modifica. En lugar de modificar todos elementos de la columna después de pivote, sólo se modifican los pertenecientes a la banda, sabemos que el resto de elemetos son cero. En conclusión, sólo se modifican como máximo las  $s$  siguientes filas bajo la fila del pivote, o bien las filas restantes.
3. Para modificar la fila donde se efectúa la eliminación sólo se requiere utilizar el resto de la fila del pivote (elementos de la banda), sin el pivote, porque sabemos que la entrada que se modifica sumando el producto del pivote por un escalar debe ser cero, así que lo mejor es asignar a esta entrada el valor de cero, así evitaremos errores de redondeo en estas entradas.
4. Se debe considerar el primer y último elemento de cada fila que se modifica en cada iteración, sólo se trabajará con las entradas que se modifican.
5. Se establece la posición de cada entrada que se elimina de acuerdo a la posición en el formato de banda.

Para analizar las entradas que se modifican podemos establecer índices que nos interesan en el siguiente ejemplo.

**Ejemplo 11.** Sea una matriz de banda de tamaño  $n = 7$ , con  $r = 2$  y  $s = 3$ .

$$\begin{pmatrix} B_{1,1} & B_{1,2} & B_{1,3} & 0 & 0 & 0 & 0 \\ B_{2,1} & B_{2,2} & B_{2,3} & B_{3,4} & 0 & 0 & 0 \\ B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & B_{3,5} & 0 & 0 \\ B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & B_{4,5} & B_{4,6} & 0 \\ 0 & B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} & B_{5,5} & B_{5,6} \\ 0 & 0 & B_{6,1} & B_{6,2} & B_{6,3} & B_{6,4} & B_{6,5} \\ 0 & 0 & 0 & B_{7,1} & B_{7,2} & B_{7,3} & B_{7,4} \end{pmatrix} \xrightarrow{\text{Equivalente a}} \begin{pmatrix} B_{1,1} & B_{1,2} & B_{1,3} & & & & \\ B_{2,1} & B_{2,2} & B_{2,3} & B_{3,4} & & & \\ B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & B_{3,5} & & \\ B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & B_{4,5} & B_{4,6} & \\ B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} & B_{5,5} & B_{5,6} & \\ B_{6,1} & B_{6,2} & B_{6,3} & B_{6,4} & B_{6,5} & & \\ B_{7,1} & B_{7,2} & B_{7,3} & B_{7,4} & & & \end{pmatrix}$$

Los índices  $j_1$ ,  $j_2$ ,  $j_3$  y  $j_4$  serán de apoyo para determinar las entradas que se modifican. Para el caso de la parte superior, con  $i \leq s$  y  $r + i \leq n$ , donde  $i$  es el renglón del pivote, tenemos que estos índices están dados por:

$$j_1 = i, \quad j_2 = j_1 + r, \quad j_3 = i, \quad \text{y} \quad j_4 = j_3 + r.$$

$$\begin{pmatrix} B_{1,j_1} & B_{1,2} & B_{1,j_2} & 0 & 0 & 0 & 0 \\ B_{2,j_3} & B_{2,2} & B_{2,j_4} & B_{2,4} & 0 & 0 & 0 \\ B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & B_{3,5} & 0 & 0 \\ B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & B_{4,5} & B_{4,6} & 0 \\ 0 & B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} & B_{5,5} & B_{5,6} \\ 0 & 0 & B_{6,1} & B_{6,2} & B_{6,3} & B_{6,4} & B_{6,5} \\ 0 & 0 & 0 & B_{7,1} & B_{7,2} & B_{7,3} & B_{7,4} \end{pmatrix}$$

En este ejemplo, para la primera iteración, se tiene que:

$$j_1 = i = 1, \quad j_2 = j_1 + r = 1 + 2 = 3, \quad j_3 = i = 1, \quad \text{y} \quad j_4 = j_3 + r = 1 + 2 = 3.$$

Para los elementos en el centro, con  $s + 1 \leq i \leq n - 1$  y  $r + i \leq n$ , donde  $k$  indica el renglón de eliminación.

$$j_1 = s + 1, \quad j_2 = j_1 + r, \quad j_3 = j_1 + i - k, \quad \text{y} \quad j_4 = j_3 + r.$$

$$\begin{pmatrix} B_{1,1} & B_{1,2} & B_{1,3} & 0 & 0 & 0 & 0 \\ 0 & B'_{2,2} & B'_{2,3} & B_{2,4} & 0 & 0 & 0 \\ 0 & 0 & B'_{3,3} & B'_{3,4} & B_{3,5} & 0 & 0 \\ 0 & 0 & 0 & B'_{4,j_1} & B'_{4,5} & B_{4,j_2} & 0 \\ 0 & 0 & 0 & 0 & B'_{5,4} & B'_{5,5} & B_{5,6} \\ 0 & 0 & 0 & B'_{6,j_3} & B'_{6,3} & B_{6,j_4} & B_{6,5} \\ 0 & 0 & 0 & B_{7,1} & B_{7,2} & B_{7,3} & B_{7,4} \end{pmatrix}$$

En esta iteración los índices son los siguientes:

$$j_1 = s + 1 = 4, \quad j_2 = j_1 + r = 6, \quad j_3 = j_1 + i - k = 2, \quad \text{y} \quad j_4 = j_3 + r = 4.$$

Para los elementos en las últimas filas, con  $s + 1 \leq i \leq n - 1$  y  $r + i > n$ .

$$j_1 = s + 1, \quad j_2 = j_1 + n - i, \quad j_3 = j_1 + i - k, \quad \text{y} \quad j_4 = j_3 + n - i.$$

$$\begin{pmatrix} B_{1,1} & B_{1,2} & B_{1,3} & 0 & 0 & 0 & 0 \\ 0 & B'_{2,2} & B'_{2,3} & B_{2,4} & 0 & 0 & 0 \\ 0 & 0 & B'_{3,3} & B'_{3,4} & B_{3,5} & 0 & 0 \\ 0 & 0 & 0 & B'_{4,4} & B'_{4,5} & B_{4,6} & 0 \\ 0 & 0 & 0 & 0 & B'_{5,4} & B'_{5,5} & B_{5,6} \\ 0 & 0 & 0 & 0 & 0 & B'_{6,j_1} & B'_{6,j_2} \\ 0 & 0 & 0 & 0 & 0 & B'_{7,j_3} & B'_{7,j_4} \end{pmatrix}$$

En la última iteración, por ejemplo, los índices quedan de la siguiente forma:

$$j_1 = s + 1 = 4, \quad j_2 = j_1 + n - i = 5, \quad j_3 = s + 1 + i - k = 3, \quad \text{y} \quad j_4 = j_3 + n - i = 4.$$

Se pueden generalizar los índices, para incluir los tres casos en uno.

Sea  $B$  una matriz de tamaño  $n$ ,  $i$  indica el renglón del pivote y  $k$  el renglón que se está modificando, tenemos entonces, la matriz en la siguiente forma:

$$B = \begin{pmatrix} B_{1,1} & \dots & B_{1,r+1} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & B'_{i,j_1} & \dots & B'_{i,j_2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & B'_{k,j_3} & \dots & B'_{k,j_4} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & B_{n,1} & \dots & B_{n,s+1} \end{pmatrix}$$

Los índices se pueden escribir como sigue:

$$j_1 = \min\{i, s + 1\}, \quad j_2 = \min\{j_1 + r, j_1 + n - i\}, \quad j_3 = \min\{i, s + i - k + 1\}, \quad \text{y} \quad j_4 = \min\{j_3 + r, j_3 + n - i\}.$$

Para comprender mejor esta idea se programó un algoritmo que permite visualizar: los elementos en la matriz que se modifican en una iteración dada, los renglones que cambian y los elementos utilizados del renglón donde se encuentra el pivote. A continuación presenta el algoritmo.

En el algoritmo 10 se muestra en la matriz con letras  $m$ , las entradas en la matriz que se modifican en la iteración  $i$ , las  $f$  indican los elementos en la fila del pivote, los ceros indican que ese elemento no se utilizó en la iteración.

---

**Algoritmo 10** Localización de la parte de la matriz que participa en un paso del método de Gauss.

---

**Entrada:**  $n$ : tamaño de la matriz;  $r$ : semiancho superior;  $s$ : semiancho inferior;  $i$ : renglón del pivote;

**Salida:**  $C$ : matriz en la que se indican los elementos que se modifican;

- 1:  $j_1 \leftarrow \min\{i, s + 1\}$ ;
  - 2:  $j_2 \leftarrow \min\{j_1 + r, j + n - i\}$ ;
  - 3:  $C \leftarrow$  longitud de la primera fila de  $G$ ;
  - 4:  $C[i, j_1 : j_2] \leftarrow f$
  - 5: **para**  $k = i + 1 \dots \min\{i + s, n\}$  **hacer**
  - 6:    $j_3 \leftarrow \min\{s + i - k + 1, i\} + \max\{0, i - (p + 1)\}$ ;
  - 7:    $j_4 \leftarrow \min\{j_3 + r, j_3 + n - i\}$ ;
  - 8:    $C[k, j_3 : j_4] \leftarrow m$ ;
  - 9: **fin para**
- 

**Ejemplo 12.** En la siguiente matriz se muestran las entradas que se modifican, al hacer la cuarta iteración de eliminación en una matriz de tamaño  $n = 8$ , con  $r = 2$  y  $s = 3$ .

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & f & f & f & 0 & 0 \\ 0 & 0 & 0 & m & m & m & 0 & 0 \\ 0 & 0 & 0 & m & m & m & 0 & 0 \\ 0 & 0 & 0 & m & m & m & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

El algoritmo 11 realiza la eliminación de Gauss, utilizando la modificación anteriormente explicada, con el formato de banda.

---

**Algoritmo 11** Eliminación de Gauss modificado para matrices de banda.

---

**Entrada:**  $B$ : matriz en formato de banda;

**Salida:**  $G$ : matriz reducida en formato de banda;

```

1:  $G \leftarrow$  copia de  $B$ ;
2:  $n \leftarrow$  número de filas de  $G$ ;
3:  $r \leftarrow$  longitud de la primera fila de  $G$ ;
4:  $s \leftarrow$  longitud de la última fila de  $G$ ;
5: para  $i = 1 \dots n - 1$  hacer
6:    $j_1 \leftarrow \min\{i, s + 1\}$ ;
7:    $j_2 \leftarrow \min\{j_1 + r, j_1 + n - i\}$ ;
8:   para  $k = i + 1 \dots \min\{i + s, n\}$  hacer
9:      $j_3 \leftarrow \min\{s + i - k + 1, i\}$ ;
10:     $j_4 \leftarrow \min\{j_3 + r, j_3 + n - i\}$ ;
11:     $m \leftarrow -G[k, j_3]/G[i, j_1]$ ;
12:     $G[k, j_3] \leftarrow 0$ ;
13:     $G[k, j_3 + 1 : j_4] += mG[i, j_1 + 1 : j_2]$ ;
14:   fin para
15: fin para

```

---

Después de aplicar el algoritmo de eliminación de Gauss modificado, obtenemos una matriz triangular, pero dada en formato de banda. Por lo tanto, el determinante de la matriz está dado por el producto de los elementos en su diagonal.

**Ejemplo 13.** Calcular el determinante de la matriz  $B$  dada en formato de banda, con  $n = 6$ ,  $r = 2$  y  $s = 3$ .

$$\left( \begin{array}{cccccc} B_{1,1} & B_{1,2} & B_{1,3} & & & \\ B_{2,1} & B_{2,2} & B_{2,3} & B_{3,4} & & \\ B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & B_{3,5} & \\ B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & B_{4,5} & B_{4,6} \\ B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} & B_{5,5} & \\ B_{6,1} & B_{6,2} & B_{6,3} & B_{6,4} & & \end{array} \right) \xrightarrow{\text{Después de la eliminación}} \left( \begin{array}{cccccc} B_{1,1} & B_{1,2} & B_{1,3} & & & \\ 0 & B'_{2,2} & B'_{2,3} & B_{3,4} & & \\ 0 & 0 & B'_{3,3} & B'_{3,4} & B_{3,5} & \\ 0 & 0 & 0 & B'_{4,4} & B'_{4,5} & B_{4,6} \\ 0 & 0 & 0 & B'_{5,4} & B_{5,5} & \\ 0 & 0 & 0 & B'_{6,4} & & \end{array} \right)$$

Después de realizar la eliminación, en el formato de banda nos queda algo parecido a una matriz triangular, para encontrar el determinante de la matriz, sólo basta determinar los índices correspondientes a la diagonal principal, que son los sombreados en verde. Notemos que la posición de estos elementos está dada por el mínimo entre  $i$  y  $s + 1$  ( $\min\{i, s + 1\}$ ). De esta forma, el determinante de matrices en formato de banda, después de aplicar la modificación de la eliminación de Gauss, es:

$$\det(B) = B_{1,1} \prod_{i=2}^n B'[i, \min\{i, s + 1\}].$$

En este ejemplo el determinante es:

$$\det(B) = B_{1,1} B'_{2,2} B'_{3,3} B'_{4,4} B'_{5,4} B'_{6,4}.$$

**Algoritmo 12** Determinante con eliminación de Gauss modificado.

**Entrada:**  $B$ : matriz en formato de banda;

**Salida:**  $d$ : determinante de la matriz  $B$ ;

- 1:  $s \leftarrow$  semiancho inferior;
- 2:  $n \leftarrow$  número de filas de  $B$ ;
- 3:  $T \leftarrow$  Matriz  $B$  reducida con el algoritmo modificado de Gauss;
- 4:  $d \leftarrow \prod_{i=1}^n T[i, \min\{i, p + 1\}]$ ;

Con la modificación del algoritmo de eliminación de Gauss, se eficienta el método clásico al evitar operaciones no necesarias. Por ello este algoritmo modificado, tarda menos tiempo en calcular determinantes de matrices de banda que el algoritmo original.

Continuaremos el estudio de los métodos para calcular los determinantes matrices de Toeplitz, con el método modificado del algoritmo de Levinson–Durbin, para resolver sistemas de ecuaciones lineales. En el siguiente algoritmo, como con la eliminación de Gauss, utilizamos un método conocido para obtener un método más eficiente para el cálculo de determinantes de matrices de Toeplitz de banda.

### 3. Modificación del algoritmo de Levinson–Durbin

En esta sección estudiaremos el algoritmo de Levinson–Durbin para resolver el problema de Yule–Walker, posteriormente, realizamos modificaciones al algoritmo, como en la sección anterior, en este caso para calcular determinantes de matrices de Toeplitz simétricas y matrices de Toeplitz simétricas de banda.

#### 3.1. Ecuaciones de Yule–Walker

Estas ecuaciones surgen en la teoría de procesos estocásticos, más precisamente, para calcular ciertos parámetros en el modelo autorregresivo.

**Definición 6** (Ecuaciones de Yule–Walker). Sea  $T_n$  una matriz de Toeplitz simétrica de orden  $n$  con la primera columna  $t_0, \dots, t_{n-1}$ , y sea  $U$  el vector de longitud  $n$ , con entradas  $t_1, \dots, t_n$ . Se busca un vector  $x$  de longitud  $n$  que satisfaga el siguiente sistema de ecuaciones:

$$T_n x = U,$$

equivalentemente,

$$\begin{pmatrix} t_0 & t_1 & \dots & t_{n-1} \\ t_1 & t_0 & \dots & t_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & \dots & t_0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix}.$$

Una técnica para resolver este problema es el algoritmo de Levinson–Durbin.

### 3.2. Explicación de la idea de Levinson, para $n = 4$

**Ejemplo 14.** Para  $n = 4$ , los datos iniciales son algunos números  $t_0, t_1, t_2, t_3, t_4$ , y se tiene el problema

$$T_4x = U \quad \mapsto \quad \begin{pmatrix} t_0 & t_1 & t_2 & t_3 \\ t_1 & t_0 & t_1 & t_2 \\ t_2 & t_1 & t_0 & t_1 \\ t_3 & t_2 & t_1 & t_0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix}.$$

Hacemos el cambio lineal de variables  $x = Cy$  de una forma especial con el propósito de eliminar la última columna del sistema, excepto la entrada  $(4, 4)$ :

$$T_4C = \begin{pmatrix} t_0 & t_1 & t_2 & t_3 \\ t_1 & t_0 & t_1 & t_2 \\ t_2 & t_1 & t_0 & t_1 \\ t_3 & t_2 & t_1 & t_0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -\alpha_3 \\ 0 & 1 & 0 & -\alpha_2 \\ 0 & 0 & 1 & -\alpha_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} t_0 & t_1 & t_2 & 0 \\ t_1 & t_0 & t_1 & 0 \\ t_2 & t_1 & t_0 & 0 \\ t_3 & t_2 & t_1 & q \end{pmatrix}.$$

Aquí  $\alpha_1, \alpha_2, \alpha_3$  son las incógnitas y se eligen de tal manera que las entradas  $(1, 4), (2, 4), (3, 4)$  del producto sean nulas. En otras palabras, las incógnitas  $\alpha_1, \alpha_2, \alpha_3$  deben satisfacer el sistema de ecuaciones:

$$\begin{aligned} -t_0\alpha_3 - t_1\alpha_2 - t_2\alpha_1 + t_3 &= 0; \\ -t_1\alpha_3 - t_0\alpha_2 - t_1\alpha_1 + t_2 &= 0; \\ -t_2\alpha_3 - t_1\alpha_2 - t_0\alpha_1 + t_1 &= 0. \end{aligned}$$

Escribimos este sistema en forma matricial, empezando con la última ecuación:

$$\begin{pmatrix} t_0 & t_1 & t_2 \\ t_1 & t_0 & t_1 \\ t_2 & t_1 & t_0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}. \quad (3)$$

Después del cambio  $x = Cy$  para las incógnitas nuevas se obtiene el sistema

$$T_4x = T_4Cy = \begin{pmatrix} t_0 & t_1 & t_2 & 0 \\ t_1 & t_0 & t_1 & 0 \\ t_2 & t_1 & t_0 & 0 \\ t_3 & t_2 & t_1 & q_4 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix}.$$

En particular, las incógnitas  $y_1$ ,  $y_2$  y  $y_3$  deben satisfacer el sistema

$$\begin{pmatrix} t_0 & t_1 & t_2 \\ t_1 & t_0 & t_1 \\ t_2 & t_1 & t_0 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}. \quad (4)$$

Notemos que los sistemas (3) y (4) coinciden. Más aún, es un problema idéntico al inicial, pero de orden  $n - 1$ . Después de resolver (4) la incógnita  $y_4$  se expresa fácilmente en términos de  $y_1$ ,  $y_2$  y  $y_3$ :

$$y_4 = \frac{t_4 - (t_3 y_1 + t_2 y_2 + t_1 y_3)}{q_4},$$

donde

$$q_4 = t_0 - t_1 \alpha_1 - t_2 \alpha_2 - t_3 \alpha_3.$$

Este razonamiento muestra que el sistema de ecuaciones de Yule–Walker se puede resolver de manera *recursiva*, cuando para resolver el problema de orden  $n$  se forma y se resuelve un problema de orden  $n - 1$ . Aún más cómodo es organizar los cálculos de manera *iterativa*, como se muestra en la siguiente subsección.

### 3.3. Solución general de las ecuaciones de Yule–Walker

El algoritmo de Levinson–Durbin es un proceso iterativo para resolver las ecuaciones de Yule–Walker. primero se resuelve el sistema de orden 1, el resultado se usa para resolver el sistema de orden 2, el resultado se usa para resolver el sistema de orden 3, etc. Para cada  $n \in \{1, 2, 3, \dots\}$ , el cambio de variable  $x = Cy$  será de la forma

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 & -\alpha_{n-1} \\ 0 & 1 & \dots & 0 & -\alpha_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\alpha_1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}$$

La matriz  $C$  tiene que satisfacer lo siguiente:

$$\begin{pmatrix} t_0 & t_1 & \dots & t_{n-1} \\ t_1 & t_0 & \dots & t_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-2} & t_{n-3} & \dots & t_1 \\ t_{n-1} & t_{n-2} & \dots & t_0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \dots & 0 & -\alpha_{n-1} \\ 0 & 1 & \dots & 0 & -\alpha_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\alpha_1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} = \begin{pmatrix} t_0 & t_1 & \dots & 0 \\ t_1 & t_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-2} & t_{n-3} & \dots & 0 \\ t_{n-1} & t_{n-2} & \dots & q_n \end{pmatrix}$$

De lo anterior se obtiene que,

$$q_n = t_0 - \sum_{i=1}^{n-1} t_i \alpha_i.$$

Después de cambio de variable, tenemos el siguiente sistema:

$$T_n C y = \begin{pmatrix} t_0 & t_1 & \dots & 0 \\ t_1 & t_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-2} & t_{n-3} & \dots & 0 \\ t_{n-1} & t_{n-2} & \dots & q_n \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_{n-1} \\ t_n \end{pmatrix}$$

De aquí se tiene que,

$$y_n = \frac{t_n - \sum_{i=1}^{n-1} t_{n-i} y_i}{q_n}$$

Para obtener la solución del sistema de ecuaciones inicial, se empieza por calcular  $y_1$  y  $y_2$ , posteriormente se construye la solución del sistema. Como vimos en el ejemplo anterior, los sistemas para  $y$  y  $\alpha$  son equivalentes, por lo tanto,

$$\alpha_n = y_n$$

En cada iteración se resuelve el sistema para cada valor de  $n$ . Para  $n = 1$ , la solución es inmediata,  $x_1 = y_1$ . Para  $n = 2$ ,  $\alpha_1 = y_1$ ,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & -\alpha_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

Para  $n = 3$ , se restablecen los valores  $y$ ,

$$y_1 \leftarrow x_1 \quad y \quad y_2 \leftarrow x_2$$

Por la equivalencia de los sistemas, tenemos,

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

Se calcula el valor de  $y_3$ , se hace otra iteración del algoritmo y se obtiene la solución para el problema de Yule-Walker de tamaño  $n = 3$ .

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\alpha_2 \\ 0 & 1 & -\alpha_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Para  $n$  arbitrario, se reasignan valores para cada  $y$ .

$$y_1 \leftarrow x_1, \quad y_2 \leftarrow x_2, \quad \dots, \quad y_{n-1} \leftarrow x_{n-1}$$

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Se calcula  $y_n$ . Y finalmente la solución del sistema inicial es:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 & -\alpha_{n-1} \\ 0 & 1 & \dots & 0 & -\alpha_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\alpha_1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}$$

A continuación se muestra el algoritmo de Levinson–Durbin para resolver el problema de Yule–Walker.

---

**Algoritmo 13** Levinson–Durbin para resolver Yule–Walker

---

**Entrada:**  $b$ : elementos de la primera fila;

**Salida:**  $z$ : solución del problema de Yule–Walker;

1:  $n \leftarrow$  Número de elementos de  $b - 1$ ;

2:  $z \leftarrow$  Arreglo de ceros de tamaño  $n$ ;

3:  $z[1] \leftarrow b[2]/b[1]$ ;

4: **para**  $i = 2, \dots, n$  **hacer**

5:      $y \leftarrow z$

6:      $z[i] \leftarrow (b[i + 1] - b[i; 2; -1] \cdot y[1; ; i - 1]) / (b[1] - b[2; ; i] \cdot y[1; ; i - 1])$

7:      $z[1; ; i - 1] \leftarrow z[1; ; i - 1] - y[i - 1; ; 1; -1] \cdot z[i]$

8: **fin para**

---

Ahora que conocemos el algoritmo Levinson–Durbin, lo utilizaremos para calcular los determinantes de matrices de Toeplitz simétricas y de Toeplitz simétricas de banda, para lo cual realizaremos algunas modificaciones al algoritmo.

### 3.4. Algoritmo de Levinson–Durbin modificado

El algoritmo de Levinson–Durbin, puede modificarse para calcular los determinantes de matrices de Toeplitz simétricas, la idea de esta modificación consiste en diagonalizar.

Sea una matriz de Toeplitz  $T_n$ , queremos calcular su determinante.

$$T_n = \begin{pmatrix} t_0 & t_1 & \dots & t_{n-1} \\ t_1 & t_0 & \dots & t_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & \dots & t_0 \end{pmatrix}$$

Utilizamos la idea del algoritmo Levinson–Durbin, para calcular el determinante,

$$\det(T_n) = \det(T_n) \cdot 1 = \det(T_n) \cdot \det(C) = \det(T_n C)$$

Se tiene entonces que,

$$\det(T_n) = \begin{vmatrix} t_0 & t_1 & \dots & t_{n-1} \\ t_1 & t_0 & \dots & t_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & \dots & t_0 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & \dots & -\alpha_{n-1} \\ 0 & 1 & \dots & -\alpha_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{vmatrix}$$

Como hemos visto, en el algoritmo de Levinson–Durbin, esto reduce el problema a otro similar, pero de orden  $n - 1$ , y puede usarse para calcular el determinante de la matriz  $T_n$  de la siguiente forma:

$$\begin{aligned} \det(T_n) = \det(T_n C) &= \begin{vmatrix} t_0 & t_1 & \dots & t_{n-2} & 0 \\ t_1 & t_0 & \dots & t_{n-3} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{n-2} & t_{n-3} & \dots & t_0 & 0 \\ t_{n-1} & t_{n-2} & \dots & t_1 & q_n \end{vmatrix} = \begin{vmatrix} t_0 & t_1 & \dots & t_{n-2} \\ t_1 & t_0 & \dots & t_{n-3} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-2} & t_{n-3} & \dots & t_0 \end{vmatrix} \cdot q_n \\ &= \begin{vmatrix} t_0 & t_1 & \dots & t_{n-3} \\ t_1 & t_0 & \dots & t_{n-4} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-3} & t_{n-4} & \dots & t_0 \end{vmatrix} \cdot q_{n-1} q_n = \dots = q_1 \cdots q_n \end{aligned}$$

Del algoritmo de Levison–Durbin, sabemos que

$$q_n = t_0 - \sum_{i=1}^{n-1} t_i \alpha_i.$$

Los valores  $\alpha_i$  se calculan como se explicó previamente. En conclusión se tiene que:

$$\det(T_n) = \det(T_{n-1})q_n = q_1 \cdots q_n \quad (5)$$

Para el caso de matrices de Toeplitz simétricas de banda, se hace la siguiente modificación

$$q_n = t_0 - \sum_{i=1}^l t_i \alpha_i \quad y \quad y_n = \frac{t_n - \sum_{i=1}^l t_{l-i} y_i}{q_n}$$

Donde

$$l = \min\{r, n\}$$

El producto  $Cy$ , se construye directamente la matriz resultante, no se hace la multiplicación, con lo que se evitan operaciones no necesarias.

$$\begin{pmatrix} 1 & 0 & \dots & 0 & -\alpha_{n-1} \\ 0 & 1 & \dots & 0 & -\alpha_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\alpha_1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} y_1 - \alpha_{n-1} y_n \\ y_2 - \alpha_{n-2} y_n \\ \vdots \\ y_{n-1} - \alpha_1 y_n \\ y_n \end{pmatrix}$$

Con el algoritmo Levinson–Durbin modificado se puede calcular determinantes de matrices de Toeplitz simétricas, cuando se trata de matrices de banda se realizan modificaciones para evitar operaciones no necesarias y lograr con ello una mejor funcionalidad del algoritmo.

---

**Algoritmo 14** Levinson–Durbin modificado para calcular determinantes.

---

**Entrada:**  $b$ : elementos de la banda en la primera fila;  $n$ : tamaño de la matriz;

**Salida:**  $det$ : determinante de la matriz de Toeplitz asociada a  $b$ ;

```

1:  $l \leftarrow$  Número de elementos de  $b$ ;
2:  $r \leftarrow l-1$ 
3:  $z \leftarrow$  Arreglo de ceros de tamaño  $n$ .
4:  $q_1 \leftarrow b[1]$ ;
5:  $q \leftarrow 1$ 
6: para  $i = 1 \dots n$  hacer
7:   si  $i \leq r$ , entonces
8:      $q_1 \leftarrow b[1] - (b[2;;i] \cdot z[1;;i-1])$ ;
9:      $z[i] \leftarrow (b[i+1] - b[i;;2;-1] \cdot z[1;;i-1])/q_1$ 
10:     $z[1;;i-1] \leftarrow z[1;;i-1] - \text{Reverso}[z[1;;i-1]] \cdot z[i]$ 
11:   si no
12:      $q_1 \leftarrow b[1] - (b[2;;r+1] \cdot z[1;;r])$ ;
13:      $z[i] \leftarrow (-b[r+1;;2;-1] \cdot z[i-r;;i-1])/q_1$ 
14:      $z[1;;i-1] \leftarrow z[1;;i-1] - \text{Reverso}[z[1;;i-1]] \cdot z[i]$ 
15:   fin si
16:    $q = q_1 \cdot q$ ;
17: fin para
18:  $det = q$ ;

```

---

## 4. Fórmula de Baxter–Schmidt

La fórmula de Baxter–Schmidt se aplica a matrices casi triangulares para calcular su determinante, este método requiere la serie recíproca de la serie de potencia que genera la matriz. Se revisará en esta sección la idea de la fórmula y la construcción de ésta con su demostración.

### 4.1. Serie recíproca

Sea  $a(z) = a_0 + a_1z + a_2z^2 + \dots$ ,  $a_0 \neq 0$ , su función recíproca, o serie recíproca, está representada por

$$c(z) := 1/a(z) = c_0 + c_1z + c_2z^2 + \dots$$

Notemos que  $c_0 = a_0^{-1}$ .

**Ejemplo 15.** Sea  $a(z) = 1 + 3z + 4z^2$ , los primeros términos de su serie recíproca son:

$$c(z) = 1 - 3z + 5z^2 - 3z^3 - 11z^4 + 45z^5 + \dots$$

A continuación se presenta el algoritmo para calcular los coeficientes de la serie recíproca de una serie de potencias.

---

**Algoritmo 15** Coeficientes recíprocos de una serie de potencias.

---

**Entrada:**  $a$ : coeficientes de la serie de potencias;  $m$ : número de coeficientes recíprocos;

**Salida:**  $c$ : coeficientes de la serie recíproca;

- 1:  $l =$  Número de elementos de  $a$ ;
  - 2:  $c \leftarrow$  Arreglo de ceros de tamaño  $m$ .
  - 3: **para**  $i = 1 \dots n$  **hacer**
  - 4:    $k = \min\{i, l\}$ ;
  - 5:    $c[i] = -a[2 : k].\text{Inverso}(c[i - k + 1 : i - 1])/a[1]$ ;
  - 6: **fin para**
- 

### 4.2. Fórmula de Baxter–Schmidt

Supongamos que  $a(z)$  es una serie de potencias no negativas de  $z$ :

$$a(z) = \sum_{k=0}^{\infty} a_k z^k = a_0 + a_1z + a_2z^2 + \dots$$

Entonces  $z^{-r}a(z)$  es una serie de potencias  $-r, -r + 1, \dots$  de la variable  $z$ :

$$z^{-r}a(z) = z^{-r} \sum_{k=0}^{\infty} a_k z^k = \sum_{k=-r}^{\infty} a_{k+r} z^k = a_0 z^{-r} + a_1 z^{-r+1} + a_2 z^{-r+2} + \dots,$$

La matriz de Toeplitz correspondiente tiene sólo  $r$  diagonales no triviales por arriba de la diagonal principal. Para matrices de banda se tiene que  $r < n - 1$ . Las matrices generadas por series de potencia, tienen la siguiente forma,

$$T_n(z^{-2}a(z)) = \begin{pmatrix} a_r & a_{r-1} & \dots & a_0 & 0 & 0 & \dots \\ a_{r+1} & a_r & \dots & a_1 & a_0 & 0 & \dots \\ a_{r+2} & a_{r+1} & \dots & a_2 & a_1 & a_0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

La fórmula de Baxter–Schmidt (ecuación 6), sirve para calcular determinantes de matrices de esta forma. Para matrices casi triangulares superiores es también aplicable la fórmula, dado que:

$$\det(A) = \det(A^T)$$

**Teorema 2.** Si  $n, r \geq 1$ , entonces

$$a_0^{-n} D_n(z^{-r} a(z)) = (-1)^{rn} c_0^{-r} D_r(z^{-n} c(z)). \quad (6)$$

O equivalentemente,

$$\det(A) = D_n(z^{-r} a(z)) = \frac{(-1)^{rn} a_0^n D_r(z^{-n} c(z))}{c_0^r}.$$

Observemos que para  $r < 1$  se trata de una matriz triangular inferior, por lo que su determinante se calcula de manera inmediata.

**Dem.**

$$A = \begin{pmatrix} a_r & \dots & a_1 & a_0 & \dots & 0 \\ a_{r+1} & \dots & a_2 & a_1 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{n-1} & \dots & a_{n-r} & a_{n-r-1} & \dots & a_0 \\ a_n & \dots & a_{n-r+1} & a_{n-r} & \dots & a_1 \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{n+r-1} & \dots & a_n & a_{n-1} & \dots & a_r \end{pmatrix}, C = \begin{pmatrix} c_0 & \dots & 0 & 0 & \dots & 0 \\ c_1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ c_{r-1} & \dots & c_0 & 0 & \dots & 0 \\ c_{r-1} & \dots & c_1 & 1 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ c_{n-1} & \dots & c_{n-r} & 0 & \dots & 1 \end{pmatrix}$$

Tenemos que,

$$AC = \begin{pmatrix} 0 & D \\ -R & * \end{pmatrix},$$

donde,

$$R = \begin{pmatrix} a_0 c_0 & \dots & a_0 c_{n-r+1} \\ \vdots & \ddots & \vdots \\ a_{r-1} c_n + \dots + a_0 c_{n-r+1} & \dots & a_{r-1} c_{n-r+1} + \dots + a_0 c_n \end{pmatrix}$$

$$= \begin{pmatrix} a_0 & 0 & \dots & 0 \\ a_1 & a_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{r-1} & a_{r-2} & \dots & a_0 \end{pmatrix} \begin{pmatrix} c_n & c_{n-1} & \dots & c_{n-r+1} \\ c_{n+1} & c_n & \dots & c_{n-r+2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n+r-1} & c_{n+r-2} & \dots & c_n \end{pmatrix}$$

$$\det A = D_n(z^r a(z)), \quad \det C = c_0^r,$$

$$\det AC = (-1)^{r(n-r)} a_0^{n-r} (-1)^r \det R,$$

$$\det R = a_0^r D_r(z^{-n} c(z)),$$

Por lo tanto

$$c_0^r D_n(z^{-r} a(z)) = (-1)^{r(n-r)+r} a_0^n D_r(z^{-n} c(z)).$$

La fórmula de Baxter–Schmidt reduce el problema de calcular el determinante de una matriz de tamaño  $n$  a básicamente calcular la serie recíproca y el determinante de una matriz  $c$ , también de Toeplitz, de tamaño  $r$ , por ello mientras más pequeño sea el semiancho de banda superior, este método resulta más eficiente.

Para determinar la matriz  $c$ , se utiliza el algoritmo para calcular los  $n + r$  primeros coeficientes de la serie recíproca, anteriormente revisada.

**Ejemplo 16.** Para el caso donde  $r = 2$  y  $n = 7$ , se tiene,

$$A = \begin{pmatrix} a_2 & a_1 & a_0 & 0 & 0 & 0 & 0 \\ a_3 & a_2 & a_1 & a_0 & 0 & 0 & 0 \\ a_4 & a_3 & a_2 & a_1 & a_0 & 0 & 0 \\ a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 \\ a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 \\ a_8 & a_7 & a_6 & a_5 & a_4 & a_3 & a_2 \end{pmatrix},$$

Los primeros  $n + r$  términos de la serie recíproca, son :

$$c_0 + c_1z + c_2z^2 + c_3z^3 + c_4z^4 + c_5z^5 + c_6z^6 + c_7z^7 + c_8z^8$$

La matriz  $c$  asociada a la matriz  $A$ , está dada en términos de los coeficientes de la serie recíproca, de la siguiente forma

$$c = \begin{pmatrix} c_7 & c_6 \\ c_8 & c_7 \end{pmatrix}$$

Y finalmente el determinante de la matriz  $A$ , está dado por la fórmula de Baxter–Schmidt

$$\det(A) = \frac{(-1)^{rn} a_0^n D_r(z^{-n} c(z))}{c_0^r}.$$

En este caso el determinante de  $c$  es casi inmediato,  $\det(c) = c_7^2 - c_6c_8$ . De esta forma,

$$\det(A) = \frac{a_0^7(c_7^2 - c_6c_8)}{c_0^2}.$$

Para calcular determinantes de este tipo de matrices usando la fórmula de Baxter–Schmidt, primero se calculan los primeros  $n + r$  coeficientes de la serie recíproca, se construye la matriz  $c$  con los coeficientes de dicha serie y se calcula su determinante, posteriormente se aplica la fórmula.

---

#### Algoritmo 16 Baxter–Schmidt

---

**Entrada:**  $a$ : coeficientes de la serie de potencias;  $r$ : semiancho superior;  $n$ : tamaño de la matriz;

**Salida:**  $\det$ : determinante de la matriz;

- 1:  $c \leftarrow$  coeficientes de la serie recíproca;
  - 2:  $c1 \leftarrow$  matriz de Toeplitz generada por  $c$ ;
  - 3:  $\det \leftarrow ((-1)^{(rn})a[1]^n \det[c1]/c[1]^r$ ;
-

## 5. Método de Widom

En esta sección estudiaremos la fórmula de Widom para calcular determinantes de matrices de Toeplitz. Este método resulta ser muy eficiente para matrices de banda, cuando ésta es pequeña. La eficiencia de este método no depende del tamaño de la matriz, sólo del ancho de banda.

### 5.1. Fórmula de Widom

Sea  $b$  un polinomio de Laurent

$$b(t) = \sum_{j=-r}^s b_j t^j$$

La fórmula de Widom nos permite calcular el determinante de la matriz de Toeplitz  $T_n(b)$ , cuya complejidad es independiente de  $n$ , por lo cual es un método muy eficiente cuando se trata de matrices de órdenes grandes con anchos de banda pequeños. Si  $r \leq 0$  o  $s \leq 0$ , entonces  $T_n(b)$  es triangular, por tanto su determinante está dado por la  $n$ -ésima potencia del elemento en la diagonal principal. Así que, asumimos que  $r \geq 1$  y  $s \geq 1$ , con  $b_{-r} \neq 0$  y  $b_s \neq 0$ .

En este caso denotamos por  $D_n(b)$  el determinante de la matriz de Toeplitz generada por el polinomio de Laurent  $b$ . Sean  $z_1, \dots, z_{r+s}$  las raíces, todas distintas, del polinomio

$$z^r b(z) = b_{-r} + b_{-r+1}z + \dots + b_s z^{r+s}.$$

**Teorema 3.** Fórmula de Widom. Si las raíces  $z_1, \dots, z_{r+s}$ , son distintas a pares entonces, para cada  $n \geq 1$

$$D_n(b) = \sum_{\substack{M \subset \{1, \dots, r+s\} \\ |M|=s}} C_M W_M^n,$$

donde la suma es sobre todos los  $\binom{r+s}{s}$  subconjuntos  $M \subset \{1, \dots, r+s\}$  de cardinalidad  $|M| = s$ , con  $\bar{M} := \{1, \dots, r+s\} - M$ ,

$$W_M := (-1)^s b_s \prod_{j \in M} z_j, \quad C_M := \prod_{j \in M} z_j^r \prod_{j \in M, k \in \bar{M}} (z_j - z_k)^{-1}$$

**Ejemplo 17.** Sea  $b(t) = b_{-1}t^{-1} + b_0 + b_1t + b_2t^2$ , tenemos que  $r = 1$  y  $s = 2$ , con  $n = 4$ , entonces la matriz asociada a este polinomio es:

$$T_4 = \begin{pmatrix} b_0 & b_{-1} & 0 & 0 \\ b_1 & b_0 & b_{-1} & 0 \\ b_2 & b_1 & b_0 & b_{-1} \\ 0 & b_2 & b_1 & b_0 \end{pmatrix}$$

Sean  $z_1, z_2, z_3$  las raíces del polinomio  $b(t) = b_{-1} + b_0z + b_1z^2 + b_2z^3$ . Tenemos que, para  $M = \{1, 2\}$ ,  $\bar{M} = \{3\}$

$$W_M := b_2z_1z_2, \quad C_M := z_1z_2(z_1 - z_3)^{-1}(z_2 - z_3)^{-1}$$

Análogamente para  $M = \{1, 3\}$  y  $M = \{2, 3\}$ . Por lo anterior el determinante de esta matriz está dado por la suma de tres productos

$$D_4(b) = \sum_{\substack{M \subset \{1,2,3\} \\ |M|=2}} C_M W_M^4,$$

$$D_4(b) = \underbrace{C_M W_M^4}_{M=\{1,2\}} + \underbrace{C_M W_M^4}_{M=\{1,3\}} + \underbrace{C_M W_M^4}_{M=\{2,3\}}$$

**Ejemplo 18.** Observemos que, si tenemos el mismo polinomio del ejemplo anterior generando una matriz de tamaño  $n \neq 4$ , los cálculos son los mismos que en el ejemplo anterior, con la excepción de que  $W_M$  estaría elevado a la potencia  $n$ , es decir, si en el ejemplo anterior tuviéramos que  $n = 100$ , el determinante quedaría de la siguiente forma:

$$D_{100}(b) = \underbrace{C_M W_M^{100}}_{M=\{1,2\}} + \underbrace{C_M W_M^{100}}_{M=\{1,3\}} + \underbrace{C_M W_M^{100}}_{M=\{2,3\}}$$

con  $W_M$  y  $C_M$  los mismos a los calculados anteriormente.

Aquí se puede observar la eficiencia de la fórmula de Widom, ya que su complejidad no depende del tamaño de la matriz.

El algoritmo para calcular determinantes de matrices de Toeplitz usando la fórmula de Widom, requiere, calcular las raíces del polinomio de Laurent ( $z^r b(z)$ ), calcular los valores  $W_M$  y  $C_M$  para cada subconjunto  $M$  y finalmente realizar la suma de los productos  $C_M W_M$ . En el algoritmo almacenamos la información en forma de vectores, por lo que utilizamos el producto punto para el cálculo del determinante.

---

**Algoritmo 17** Fórmula de Widom
 

---

**Entrada:**  $b$ : coeficientes del polinomio de Laurent;  $r$ : semiancho superior;  $n$ : tamaño de la matriz;

**Salida:**  $D_n$ : determinante de la matriz generada por el polinomio  $b$ ;

- 1:  $l \leftarrow$  Número de elementos de  $b$ ;
  - 2:  $s \leftarrow l - r - 1$  ;
  - 3:  $num = \binom{r+s}{s}$ ;
  - 4:  $z \leftarrow$  raíces del polinomio  $z^r b(z)$ ;
  - 5:  $M0 \leftarrow$  conjunto  $\{1, \dots, r + s\}$ ;
  - 6:  $M \leftarrow$  Subconjuntos de  $M0$  de tamaño  $s$ ;
  - 7:  $\overline{M} \leftarrow$  Complementos de los subconjuntos de  $M$ ;
  - 8:  $W_M \leftarrow$  lista de ceros de tamaño  $num$ ;
  - 9:  $C_M \leftarrow$  lista de ceros de tamaño  $num$ ;
  - 10: **para**  $i = 1 \dots num$  **hacer**
  - 11:    $w_M[i] \leftarrow (-1)^s b[l] \prod_{j \in M} z_j$
  - 12: **fin para**
  - 13: **para**  $k = 1 \dots num$  **hacer**
  - 14:    $C_M[k] \leftarrow \prod_{j \in M} z_j^r \prod_{j \in M, k \in \overline{M}} (z_j - z_k)^{-1}$
  - 15: **fin para**
  - 16:  $D_n \leftarrow C_M \cdot W_M^n$
-

## 6. Comparación de métodos

En esta sección se revisarán métodos para generar matrices aleatorias en los distintos formatos, anteriormente analizados. Los generadores de matrices aleatorias, son útiles para realizar la comparación de eficiencia de los algoritmos estudiados para calcular los determinantes. Posteriormente, revisaremos la comparación de los métodos.

### 6.1. Generador de matrices aleatorias

Para realizar la comparación de los métodos estudiados, requerimos usar distintas matrices de tamaños grandes, para ello necesitaremos programas que generen los formatos estudiados y luego los transformaremos en el formato necesario para cada método.

La programación de los algoritmos se realizó en *Wolfram Mathematica*. Y para generar los formatos se utiliza el generador de números aleatorios, en esta subsección mostramos las ideas para generar cada formato, o en su caso el algoritmo para generalo.

Para los formatos de Toeplitz, Teplitz de banda, Toeplitz simétrica y Teplitz de banda simétrica, se genera una lista o vector aleatorio, para generar esta lista se requieren datos como  $n$ ,  $r$ , y  $s$ , según sea el caso, como se muestra a continuación.

Formato	Requisitos	Resultado
Toeplitz	$n$	Lista aleatoria de tamaño $2n - 1$
Toeplitz de banda	$r$ y $s$	Lista aleatoria de tamaño $r + s + 1$
Toeplitz simétrica	$n$	Lista aleatoria de tamaño $n$
Toeplitz de banda simétrica	$r$ o $s$	Lista aleatoria de tamaño $r$

Las listas o vectores aleatorios se pueden generar en *Mathematica* con la siguiente función:

$$Table[Random[], \{2n - 1\}]$$

El resultado de la función anterior es una lista de números aleatorios de tamaño  $2n - 1$ .

\*La función que genera un número aleatorio del intervalo  $(0, 1)$ , en código de programación *Wolfram* es *Random[]*, si se quiere otro intervalo de números, se puede usar la función *RandomReal[{a,b}]* y para enteros *RandomInteger[{a,b}]*, en este caso  $a$  y  $b$  enteros.

*Wolfram* tiene una función para generar matrices aleatorias en formato general, el cual puede programarse fácilmente usando la función para generar números aleatorios (función *Random*). A continuación se muestra el algoritmo.

---

**Algoritmo 18** Matrices aleatorias en formato general
 

---

**Entrada:**  $n$  : tamaño de la matriz;

**Salida:**  $M$  : Matriz aleatoria;

- 1:  $M \leftarrow$  Arreglo de ceros de tamaño  $n \times n$ ;
  - 2: **para**  $i = 1 \dots n$  **hacer**
  - 3:   **para**  $j = 1 \dots n$  **hacer**
  - 4:      $M[i, j] \leftarrow$  número aleatorio entre  $(0, 1)$ ;
  - 5:   **fin para**
  - 6: **fin para**
- 

El siguiente algoritmo sirve para generar matrices en formato de banda. Para generar este formato se requiere indicar el tamaño  $n$  y los semianchos de banda superior e inferior  $r$  y  $s$ , de la matriz que se desea obtener.

---

**Algoritmo 19** Matrices aleatorias en formato de banda
 

---

**Entrada:**  $n$  : tamaño de la matriz;  $r$  : semiancho superior;  $s$  : semiancho inferior;

**Salida:**  $B$  : Matriz aleatoria en formato de banda;

- 1:  $B \leftarrow$  arreglo vacío;
  - 2: **para**  $i = 1 \dots s + 1$  **hacer**
  - 3:    $a \leftarrow \min(r + i, r + s + 1, n)$ ;
  - 4:    $f \leftarrow$  lista de números aleatorios de tamaño  $a$ ;
  - 5:   Agregar  $f$  al arreglo  $B$ ;
  - 6: **fin para**
  - 7: **para**  $j = s + 2 \dots n$  **hacer**
  - 8:    $b \leftarrow \min(s + n - j + 1, r + s + 1, n)$ ;
  - 9:    $f \leftarrow$  lista de números aleatorios de tamaño  $b$ ;
  - 10:   Agregar  $f$  al arreglo  $B$ ;
  - 11: **fin para**
- 

**Ejemplo 19.** Un resultado del algoritmo 19, para  $n = 4$ ,  $r = 2$  y  $s = 1$ , con valores aleatorios en  $(0, 1)$ , es el siguiente,

$$\begin{pmatrix} 0,734568 & 0,662361 & 0,220904 & \\ 0,359979 & 0,166463 & 0,389195 & 0,479602 \\ 0,639884 & 0,918773 & 0,891152 & \\ 0,826778 & 0,272988 & & \end{pmatrix}$$

Este formato es utilizado en el algoritmo de eliminación de Gauss modificado. Para hacer la comparación con el método de Gauss, se realiza el cambio de formato  $B$  a  $M$ , de esta forma se realiza la comparación con la misma matriz, en diferentes formatos.

## 6.2. Comparación de métodos

En la siguiente tabla se muestra la comparación del método de eliminación de Gauss, con el método de Gauss modificado, que se utiliza para matrices en formato de banda.

Método	Eliminación de Gauss	Gauss para banda
Complejidad Teórica	$C_1 n^3$	$C_2 n w^2$
$n = 1000 \quad w = 20$	68.5	0.382
$n = 1000 \quad w = 11$	68.4	0.166
$n = 1000 \quad w = 10$	69.4	0.167
$n = 1000 \quad w = 5$	68.4	0.0739
$n = 2000 \quad w = 11$	472	0.348

Después de programar los métodos estudiados para el cálculo de los determinantes de matrices de Toeplitz de banda, se realizó una comparación de los métodos y la función interna de *Wolfram Mathematica* para calcular el determinante. Esta comparación de los métodos consistió en generar matrices aleatorias de un tamaño específico  $n$  y un ancho de banda  $w$ , posteriormente se calculó el determinante con cada uno de los métodos y se midió el tiempo de ejecución.

En la siguiente tabla se muestran los resultados obtenidos en la comparación.

Método	Complejidad (Est. teórica)	$n = 1000$ $w = 5$	$n = 5000$ $w = 5$	$n = 1000$ $w = 11$	$n = 5000$ $w = 11$	$n = 10000$ $w = 9$
Función de Wolfram	$C_1 n^3$	0.131	7.65	0.122	7.87	54.4
Modificación de Levinson-Durbin	$C_2 n w + C_3 n^2$	0.153	2.61	0.156	2.67	10.7
Gauss para matrices de banda	$C_4 n w^2$	0.0436	0.228	0.128	0.668	1.04
Fórmula de Baxter-Schmidt	$C_5 n w + C_6 w^3$	0.0156	0.0718	0.0156	0.137	0.149
Fórmula de Widom	$C_6 2^w$	0	0	0.025	0.025	0.0124

Se puede observar que cuando el ancho de banda es pequeño, la fórmula de Widom resulta ser el método más eficiente, conforme el ancho de banda aumenta, el tiempo de ejecución de este método empieza a aumentar. Resulta ser la fórmula de Baxter-Schmidt más eficiente, cuando tanto el ancho de banda como el tamaño de la matriz son grandes.

Comparación de los métodos de Widom y Baxter-Schmidt.

Método	Baxter-Schmidt	Widom
Complejidad Teórica	$C_5 n w + C_6 w^3$	$C_7 2^w$
$n = 1000 \quad w = 5$	0.00156	0.0144
$n = 1000 \quad w = 8$	0.003	0.014
$n = 5000 \quad w = 8$	0.0592	0.0721
$n = 10000 \quad w = 5$	0.00186	0.264
$n = 10000 \quad w = 11$	0.0577	0.148

## 7. Programas en el lenguaje *Wolfram Mathematica*

En esta sección se presentan en lenguaje *Wolfram* los programas de los algoritmos estudiados en las secciones anteriores, así como ejemplos de su utilización.

**Algoritmo 1.** Toeplitz a General. Realiza el cambio de formato de Toeplitz a matriz general.

```
TtoM[a_] :=
Module[{long, n, M, j, k},
  long = Length[a];
  n = IntegerPart[long/2] + 1;
  M = Table[0, {n}, {n}];
  For[j = 1, j <= n, j++,
    For[k = 1, k <= n, k++,
      M[[j, k]] = a[[n + j - k]]];
  M]
```

**Ejemplo 20.** Generar la matriz en formato general a partir de la lista de coeficientes  $\{-4, -3, -2, -1, 5, 1, 2, 3, 4\}$ , utilizando el algoritmo 1. Con el programa anterior se utiliza el siguiente comando,

$$TtoM[-4, -3, -2, -1, 5, 1, 2, 3, 4]$$

El resultado es un arreglo de listas,

$$\{\{5, -1, -2, -3, -4\}, \{1, 5, -1, -2, -3\}, \{2, 1, 5, -1, -2\}, \{3, 2, 1, 5, -1\}, \{4, 3, 2, 1, 5\}\}$$

Su forma matricial es la siguiente,

$$\begin{pmatrix} 5 & -1 & -2 & -3 & -4 \\ 1 & 5 & -1 & -2 & -3 \\ 2 & 1 & 5 & -1 & -2 \\ 3 & 2 & 1 & 5 & -1 \\ 4 & 3 & 2 & 1 & 5 \end{pmatrix}$$

Esta es la matriz generada por el polinomio de Laurent,

$$-4t^{-4} - 3t^{-3} - 2t^{-2} - 1t^{-1} + 5 + t + 2t^2 + 3t^3 + 4t^4$$

**Nota:** Se puede usar el comando *MatrixForm* para visualizar la matriz en forma matricial, pero para realizar operaciones con ella, se requiere en la forma de arreglo.

**Algoritmo 2.** Toeplitz banda a matriz general. Realiza en cambio de formato de Toeplitz banda a formato de matriz general.

```
TBtoM[b_, r_, n_] :=
Module[{long, s, b1, i, j, k, M},
  long = Length[b];
  s = long - r - 1;
  b1 = Table[0, {i, 2*n - 1}];
  b1[[n - r ;; n + s]] = b;
  M = Table[0, {n}, {n}];
  For[j = 1, j <= n, j++,
    For[k = 1, k <= n, k++,
      M[[j, k]] = b1[[n + j - k]]];
  M]
```

**Ejemplo 21.** Generar la matriz en formato general a partir del formato Toeplitz de banda  $\{1, 2, 3, 4, 5, 6, 7\}$ , con  $r = 2$  y  $n = 8$ .

$$TBtoM[\{1, 2, 3, 4, 5, 6, 7\}, 2, 8]$$

El resultado, como en el caso anterior, es un arreglo de listas, cada una es una fila de la matriz.

$$\{\{3, 2, 1, 0, 0, 0, 0, 0\}, \{4, 3, 2, 1, 0, 0, 0, 0\}, \{5, 4, 3, 2, 1, 0, 0, 0\}, \{6, 5, 4, 3, 2, 1, 0, 0\}, \{7, 6, 5, 4, 3, 2, 1, 0\}, \{0, 7, 6, 5, 4, 3, 2, 1\}, \{0, 0, 7, 6, 5, 4, 3, 2\}, \{0, 0, 0, 7, 6, 5, 4, 3\}\}.$$

Para obtener el resultado en forma matricial se utiliza, doble diagonal y el comando *MatrixForm*, después de la función **TBtoM**.

$$TBtoM[\{1, 2, 3, 4, 5, 6, 7\}, 2, 8]//MatrixForm$$

$$\begin{pmatrix} 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 \\ 5 & 4 & 3 & 2 & 1 & 0 & 0 & 0 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 0 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 7 & 6 & 5 & 4 & 3 & 2 \\ 0 & 0 & 0 & 7 & 6 & 5 & 4 & 3 \end{pmatrix}$$

**Algoritmo 3.** Banda a formato general. Este algoritmo realiza el cambio de formato de banda a matriz general.

```
BtoM[B_] :=
Module[{n, a1, s, i},
  n = Length[B];
  s = Length[B[[n]]] - 1;
  a1 = Table[0, {n}, {n}];
  For[i = 1, i <= n, i++,
    a1[[i, Max[1, i - s] ;; Max[1, i - s] - 1 + Length[B[[i]]]]] =
      B[[i]];
  a1]
```

**Ejemplo 22.** Generar la matriz completa a partir del siguiente formato de banda, utilizando el programa anterior.

$$\begin{pmatrix} \{4, 1, 4\} \\ \{0, -6, 4, 6\} \\ \{1, 3, -5, -5, -1\} \\ \{8, -1, -5, 4, 7\} \\ \{10, 0, 8, 10\} \end{pmatrix}$$

`BtoM[{{4, 1, 4}, {0, -6, 4, 6}, {1, 3, -5, -5, -1}, {8, -1, -5, 4, 7}, {10, 0, 8, 10}}]`

Cuyo resultado es el siguiente arreglo de listas,

`{{4, 1, 4, 0, 0}, {0, -6, 4, 6, 0}, {1, 3, -5, -5, -1}, {8, -1, -5, 4, 7}, {0, 10, 0, 8, 10}}`

Y en forma matricial,

$$\begin{pmatrix} 4 & 1 & 4 & 0 & 0 \\ 0 & -6 & 4 & 6 & 0 \\ 1 & 3 & -5 & -5 & -1 \\ 8 & -1 & -5 & 4 & 7 \\ 0 & 10 & 0 & 8 & 10 \end{pmatrix}$$

**Algoritmo 4.** Toeplitz banda a banda. Hace en cambio de formato de Toeplitz banda a formato de banda.

```
TBtoB[b_, r_, n_] :=
Module[{long, b1, c, i, j, l, s, Mb},
  c = Reverse[b];
  long = Length[c];
  s = long - r - 1;
  b1 = Table[
    c[[Max[s + 2 - i, 1] ;; Min[long, n - i + s + 1]]], {i, 1, n}
  ]
```

**Ejemplo 23.** Utilizando el algoritmo 4, convierta el siguiente formato de Toeplitz de banda a formato de banda. Lista de coeficiente en TB:  $\{-3, -2, -1, 4, 1, 2\}$ ,  $r = 3$  y  $n = 7$ .

TBtoB[ $\{-3, -2, -1, 4, 1, 2\}, 3, 7]$

$\{\{4, -1, -2, -3\}, \{1, 4, -1, -2, -3\}, \{2, 1, 4, -1, -2, -3\}, \{2, 1, 4, -1, -2, -3\}, \{2, 1, 4, -1, -2\}, \{2, 1, 4, -1\}, \{2, 1, 4\}\}$

A continuación se muestra el formato de banda, corresponde a la información no nula de cada fila.

TBtoB[ $\{-3, -2, -1, 4, 1, 2\}, 3, 7]$ //*MatrixForm*

$$\begin{pmatrix} \{4, -1, -2, -3\} \\ \{1, 4, -1, -2, -3\} \\ \{2, 1, 4, -1, -2, -3\} \\ \{2, 1, 4, -1, -2, -3\} \\ \{2, 1, 4, -1, -2\} \\ \{2, 1, 4, -1\} \\ \{2, 1, 4\} \end{pmatrix}$$

**Algoritmo 5.** Toeplitz simétrica a matriz general. Realiza el cambio de formato.

```
TStoM[asup_] :=
Module[{long, n, a, M, j, k},
  n = Length[asup];
  long = 2 n - 1;
  a = Table[0, {i, long}];
  a[[n + 1 ;; long]] = asup[[2 ;; n]];
  a[[1 ;; n]] = Reverse[asup];
  M = Table[0, {n}, {n}];
  For[j = 1, j <= n, j++,
    For[k = 1, k <= n, k++,
      M[[j, k]] = a[[n + j - k]]];
  M]
```

**Ejemplo 24.** Para crear una matriz de Toeplitz simétrica sólo se requiere la información de la primera fila. A continuación se muestra el ejemplo de este programa.

Entrada:

TStoM[{1, 2, 3, 4, 5}]

Resultado:

{{1, 2, 3, 4, 5}, {2, 1, 2, 3, 4}, {3, 2, 1, 2, 3}, {4, 3, 2, 1, 2}, {5, 4, 3, 2, 1}}

Forma matricial:

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

**Algoritmo 6.** Formato Toeplitz de banda simétrica a formato general.

```
TBStoM[bsup_, n_] :=
Module[{long, b, b1, r, i, j, k, M},
  r = Length[bsup] - 1;
  b1 = Table[0, {i, 2*n - 1}];
  b1[[n - r ;; n]] = Reverse[bsup];
  b1[[n ;; n + r]] = bsup;
  M = Table[0, {n}, {n}];
  For[j = 1, j <= n, j++,
    For[k = 1, k <= n, k++,
      M[[j, k]] = b1[[n + j - k]]];
  M]
```

**Ejemplo 25.** Construir una matriz de banda simétrica, a partir de la información de banda y el tamaño de la matriz.

TBStoM[{1, 2, 3}, 6]

{{1, 2, 3, 0, 0, 0}, {2, 1, 2, 3, 0, 0}, {3, 2, 1, 2, 3, 0}, {0, 3, 2, 1, 2, 3}, {0, 0, 3, 2, 1, 2}, {0, 0, 0, 3, 2, 1}}

Resultado en forma matricial:

$$\begin{pmatrix} 1 & 2 & 3 & 0 & 0 & 0 \\ 2 & 1 & 2 & 3 & 0 & 0 \\ 3 & 2 & 1 & 2 & 3 & 0 \\ 0 & 3 & 2 & 1 & 2 & 3 \\ 0 & 0 & 3 & 2 & 1 & 2 \\ 0 & 0 & 0 & 3 & 2 & 1 \end{pmatrix}$$

**Algoritmo 7.** Formato Toeplitz de banda simétrica a formato de banda. Cambio de formato.

```
TBStoB[bs_, n_] :=
Module[{long, b1, c, r, i, j, l, s, Mb},
  r = Length[bs] - 1;
  c = Table[0, {2 r + 1}];
  c[[1 ;; r]] = Reverse[bs[[2 ;; r + 1]]];
  c[[r + 1 ;; 2 r + 1]] = bs;
  long = Length[c];
  s = long - r - 1;
  b1 = Table[
    c[[Max[s + 2 - i, 1] ;; Min[long, n - i + s + 1]]], {i, 1, n}
  ]
```

**Ejemplo 26.** Cambiar de formato de Toeplitz banda a formato de banda, usando el algoritmo 7.

Entrada:

TBStoB[{1, 2, 3, 4}, 8]

Resultado: {{1, 2, 3, 4}, {2, 1, 2, 3, 4}, {3, 2, 1, 2, 3, 4}, {4, 3, 2, 1, 2, 3, 4}, {4, 3, 2, 1, 2, 3, 4}, {4, 3, 2, 1, 2, 3}, {4, 3, 2, 1, 2}, {4, 3, 2, 1}}

Resultado en forma matricial.

$$\begin{pmatrix} \{1, 2, 3, 4\} \\ \{2, 1, 2, 3, 4\} \\ \{3, 2, 1, 2, 3, 4\} \\ \{4, 3, 2, 1, 2, 3, 4\} \\ \{4, 3, 2, 1, 2, 3, 4\} \\ \{4, 3, 2, 1, 2, 3\} \\ \{4, 3, 2, 1, 2\} \\ \{4, 3, 2, 1\} \end{pmatrix}$$

**Algoritmo 8.** Gauss con pivotes diagonales. Realiza la eliminación de Gauss con pivotes diagonales, para matrices en formato de matriz general.

```
ReduceGaussN[A_] :=
Module[{G, i, k, n, mu},
  G = A;
  n = Length[G];
  For[k = 1, k <= n - 1, k++,
    For[i = k + 1, i <= n, i++,
      mu = -G[[i, k]]/G[[k, k]];
      G[[i, k]] = 0;
      G[[i, k + 1 ;; n]] += mu*G[[k, k + 1 ;; n]]];
  G]
```

**Ejemplo 27.** Para la siguiente matriz realizar la eliminación con el método de Gauss con pivotes diagonales.

$$\begin{pmatrix} 3 & -6 & 2 & 8 & -5 \\ 6 & 3 & -6 & 8 & -2 \\ -8 & -1 & -2 & -8 & 8 \\ 7 & -1 & -6 & -6 & 5 \\ 7 & 1 & -3 & -8 & -1 \end{pmatrix}$$

Con el programa del algoritmo 8, se usa el siguiente comando:

```
ReduceGaussN[{{3, -6, 2, 8, -5}, {6, 3, -6, 8, -2}, {-8, -1, -2, -8, 8}, {7, -1, -6, -6, 5}, {7, 1, -3, -8, -1}}]
```

El resultado de la eliminación de Gauss, en forma matricial,

$$\begin{pmatrix} 3 & -6 & 2 & 8 & -5 \\ 0 & 15 & -10 & -8 & 8 \\ 0 & 0 & -8 & \frac{64}{15} & \frac{56}{15} \\ 0 & 0 & 0 & -\frac{94}{5} & \frac{44}{5} \\ 0 & 0 & 0 & 0 & -\frac{1861}{423} \end{pmatrix}$$

**Algoritmo 9.** Calcula el determinante de una matriz en formato general, utilizando la función de eliminación de Gauss con pivotes diagonales.

```

DetBandN[G_] :=
Module[{T, n},
  n = Length[G];
  T = ReduceGaussN[G];
  Product[T[[i, i]], {i, n}]]

```

**Ejemplo 28.** Con la eliminación de Gauss, resulta sencillo calcular el determinante utilizando en algoritmo 9, en el caso del ejemplo anterior, tenemos lo siguiente:

```

DetBandN[{{3, -6, 2, 8, -5}, {6, 3, -6, 8, -2}, {-8, -1, -2, -8, 8}, {7, -1, -6, -6, 5}, {7, 1, -3, -8, -1}}]

```

$$\begin{vmatrix} 3 & -6 & 2 & 8 & -5 \\ 6 & 3 & -6 & 8 & -2 \\ -8 & -1 & -2 & -8 & 8 \\ 7 & -1 & -6 & -6 & 5 \\ 7 & 1 & -3 & -8 & -1 \end{vmatrix} = \begin{vmatrix} 3 & -6 & 2 & 8 & -5 \\ 0 & 15 & -10 & -8 & 8 \\ 0 & 0 & -8 & \frac{64}{15} & \frac{56}{15} \\ 0 & 0 & 0 & -\frac{94}{5} & \frac{44}{5} \\ 0 & 0 & 0 & 0 & -\frac{1861}{423} \end{vmatrix} = -29776$$

**Algoritmo 10.** Realiza la identificación de los elementos de la matriz que serán modificados en un paso de la eliminación de Gauss, así como los elementos que se emplean del renglón del pivote.

```
ElementGauss[n_, s_, r_, ki_] :=
Module[{C, i, j1, j2, j3, j4},
  j1 = Min[ki, s + 1];
  j2 = Min[j1 + r, j1 + n - ki];
  C = Table[0, {n}, {n}];
  C[[ki, j1 ;; j2]] = f;
  For[i = ki + 1, i <= Min[ki + s, n], i++,
    j3 = Min[s + ki - i + 1, ki] + Max[0, i - (s + 1)];
    j4 = Min[j3 + r, j3 + n - ki];
    C[[i, j3 ;; j4]] = m]; {C, j1, j2, j3, j4}
]
```

**Ejemplo 29.** El algoritmo 10 sirve como auxiliar para identificar los elementos de una matriz que participan en un paso de la eliminación de Gauss. En este ejemplo se trata de una matriz de tamaño  $n = 9$ , con  $r = 5$ ,  $s = 4$  en la tercera iteración del algoritmo. Se representan con  $m$  las estradas que serán modificadas y con  $f$  los elementos de la fila del pivote que participan. Los ceros indican que dicha entrada no se modifica.

*ElementGauss[9, 5, 4, 3]//MatrixForm*

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & f & f & f & f & f & 0 & 0 \\ 0 & 0 & m & m & m & m & m & 0 & 0 \\ 0 & 0 & m & m & m & m & m & 0 & 0 \\ 0 & 0 & m & m & m & m & m & 0 & 0 \\ 0 & 0 & m & m & m & m & m & 0 & 0 \\ 0 & 0 & m & m & m & m & m & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

**Algoritmo 11.** Gauss modificado para matrices de banda. Realiza la eliminación de Gauss para matrices en formato de banda.

```
ReduceGauss[B_] :=
Module[{C, i, j1, j2, j3, j4, k, n, p, q, mu},
  C = B;
  n = Length[C];
  q = Length[C[[1]]] - 1;
  p = Length[C[[n]]] - 1;
  For[k = 1, k <= n - 1, k++,
    j1 = Min[k, p + 1];
    j2 = Min[j1 + q, j1 + n - k];
    For[i = k + 1, i <= Min[k + p, n], i++,
      j3 = Min[p + k - i + 1, k];
      j4 = Min[j3 + q, j3 + n - k];
      mu = -C[[i, j3]]/C[[k, j1]];
      C[[i, j3]] = 0;
      C[[i, j3 + 1 ;; j4]] += mu*C[[k, j1 + 1 ;; j2]]];
  C]
```

**Ejemplo 30.** Dada una matriz en formato de banda, se puede realizar la eliminación de Gauss con pivotes diagonales sin tener que construir la matriz completa.

Comando:

```
ReduceGauss[{{1, 2, 3}, {2, 1, 2, 3}, {3, 2, 1, 2, 3}, {3, 2, 1, 2}, {3, 2, 1}}//MatrixForm
```

Resultado:

$$\begin{pmatrix} \{1, 2, 3\} \\ \{0, -3, -4, 3\} \\ \{0, 0, -\frac{8}{3}, -2, 3\} \\ \{0, 0, \frac{11}{2}, -\frac{1}{4}\} \\ \{0, 0, \frac{48}{11}\} \end{pmatrix}$$

**Algoritmo 12.** Calcula el determinante de matrices en formato de banda usando la eliminación de Gauss modificada para este formato.

```
DetBand1[B_, s_] :=
Module[{T, n},
  n = Length[B];
  T = ReduceGauss1[B];
  Product[T[[i, Min[i, s + 1]]], {s, n}]]
```

**Ejemplo 31.** Dada la siguiente matriz en formato de banda, calcula su determinante con el programa anterior.

$$\begin{pmatrix} \{1, 2, 3\} \\ \{2, 1, 2, 3\} \\ \{3, 2, 1, 2, 3\} \\ \{3, 2, 1, 2\} \\ \{3, 2, 1\} \end{pmatrix}$$

Se utiliza el siguiente comando para calcular el determinante de una matriz, a partir de su información de banda y el semiancho inferior.

```
DetBand[{{1, 2, 3}, {2, 1, 2, 3}, {3, 2, 1, 2, 3}, {3, 2, 1, 2}, {3, 2, 1}}, 2]
```

Después de aplicar el algoritmo 11, la matriz reducida en formato de banda queda de la siguiente forma,

$$\begin{pmatrix} \{1, 2, 3\} \\ \{0, -3, -4, 3\} \\ \{0, 0, -\frac{8}{3}, -2, 3\} \\ \{0, 0, \frac{11}{2}, -\frac{1}{4}\} \\ \{0, 0, \frac{48}{11}\} \end{pmatrix}$$

Una vez reducida, se calcula el producto de los elementos que corresponden a la diagonal. El determinante de la matriz expresada en formato de banda es 192.

**Algoritmo 13.** Levinson–Durbin para resolver Yule–Walker. Resuelve sistemas ecuaciones del tipo que establece el problema de Yule–Walker.

```
LevisonDurbinI[b_] :=
Module[{C, n, i, z},
  n = Length[b] - 1;
  z = {};
  For[i = 1, i <= n, i++,
    AppendTo[z,
      (b[[i + 1]] - Sum[b[[i + 1 - j]]*z[[j]], {j, 1, i - 1}]) / (b[[1]] -
        Sum[b[[j + 1]]*z[[j]], {j, 1, i - 1})];
  C = IdentityMatrix[i];
  C[[1 ;; i - 1, i]] = Reverse[-z[[1 ;; i - 1]]];
  z = C.z;
  z]
```

**Ejemplo 32.** Resolver el siguiente problema de Yule–Walker utilizando el algoritmo de Levinson–Durbin.

$$\begin{pmatrix} 1 & 2 & -2 & 3 \\ 2 & 1 & 2 & -2 \\ -2 & 2 & 1 & 2 \\ 3 & -2 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \\ 3 \\ 9 \end{pmatrix}$$

Con el programa anterior se usa el comando:

```
LevisonDurbinI[{1, 2, -2, 3, 9}]
```

Se obtiene el siguiente resultado,

$$\left\{ \frac{23}{56}, -\frac{25}{21}, \frac{32}{21}, \frac{131}{56} \right\}$$

Es decir,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} \frac{23}{56} \\ -\frac{25}{21} \\ \frac{32}{21} \\ \frac{131}{56} \end{pmatrix}$$

**Algoritmo 14.** Levinson–Durbin modificado. La modificación de este algoritmo, realiza el cálculo de determinantes de matrices de Toeplitz de banda simétricas.

```
LevinsonDurbinDet[b_, n_] :=
Module[{r, i, z, q = 1, bi, q1 = b[[1]]},
  r = Length[b] - 1;
  z = Table[0, {n}];
  For[i = 1, i <= n, i++,
    If[i <= r,
      q1 = b[[1]] - (b[[2 ;; i]].z[[1 ;; i - 1]]);
      z[[i]] = (b[[i + 1]] - b[[i ;; 2 ;; -1]].z[[1 ;; i - 1]])/q1;
      z[[1 ;; i - 1]] -= Reverse[z[[1 ;; i - 1]]]*z [[i]];
      q = q1*q,
      q1 = b[[1]] - (b[[2 ;; r + 1]].z[[1 ;; r]]);
      z[[i]] = -b[[r + 1 ;; 2 ;; -1]].z[[i - r ;; i - 1]]/q1;
      z[[1 ;; i - 1]] -= Reverse[z[[1 ;; i - 1]]]*z [[i]];
      q = q1*q]];
  q]
```

**Ejemplo 33.** Calcular el determinante de la siguiente matriz de Teopltiz simétrica de banda, utilizando el algoritmo 14 (Levinson–Durbin modificado).

$$A = \begin{pmatrix} 1 & 3 & 2 & 4 & 0 & 0 & 0 \\ 3 & 1 & 3 & 2 & 4 & 0 & 0 \\ 2 & 3 & 1 & 3 & 2 & 4 & 0 \\ 4 & 2 & 3 & 1 & 3 & 2 & 4 \\ 0 & 4 & 2 & 3 & 1 & 3 & 2 \\ 0 & 0 & 4 & 2 & 3 & 1 & 3 \\ 0 & 0 & 0 & 4 & 2 & 3 & 1 \end{pmatrix}$$

Sólo se requiere indicar los elementos de la banda en la primera fila y el tamaño de la matriz, en este caso  $n = 7$ .

LevinsonDurbinDet[{1, 3, 2}, 7]

Obtenemos que,

$$\det(A) = 2449$$

**Algoritmo 15.** Coeficientes recíprocos de una serie de potencias. Calcula un número dado de estos coeficientes.

```
CoefsReciprocalBand[a_, m_] :=
Module[{c, i, k, la},
  la = Length[a];
  c = Table[0, {m}];
  c[[1]] = 1/a[[1]];
  For[i = 2, i <= m, i++,
    k = Min[i, la];
    c[[i]] = -a[[2 ;; k]].Reverse[c[[i - k + 1 ;; i - 1]]]/a[[1]];
  ]
c]
```

**Ejemplo 34.** Obtener los primeros siete coeficientes recíprocos de la serie  $1 + 3z + 4z^2 + 0z^3 + \dots$ . Utilizamos el programa del algoritmo 15, con los términos significativos de la serie,

$$\text{CoefsReciprocalBand}\{1, 3, 4\}, 7]$$

Y obtenemos los primeros siete coeficientes de la serie recíproca

$$\{1, -3, 5, -3, -11, 45, -91\}$$

**Algoritmo 16.** Baxter–Schmidt. Calcula el determinante de matrices casi triangulares superiores, empleando la fórmula de Baxter–Schmidt.

```
BaxterSchmidtBand[a_, r_, n_] :=
Module[{cs, a1, c1, da, dac},
  cs = CoefsReciprocalBand[a, n + r];
  c1 = TBtoM[cs[[n - r + 2 ;; n + r]], r - 1, r];
  dac = ((-1)^(r*n)) * a[[1]]^n * Det[c1];
  dac/cs[[1]]^r
]
```

**Ejemplo 35.** Utilizando la fórmula de Baxter–Schmidt calcular el determinante de la matriz  $B$ .

$$B = \begin{pmatrix} 1 & 3 & 0 & 0 & 0 & 0 \\ 2 & 1 & 3 & 0 & 0 & 0 \\ 3 & 2 & 1 & 3 & 0 & 0 \\ 4 & 3 & 2 & 1 & 3 & 0 \\ 5 & 4 & 3 & 2 & 1 & 3 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

Cuya serie de potencias es:  $3 + 1z + 2z^2 + 3z^3 + 4z^4 + 5z^5 + 6z^6 + \dots$ , con  $n = 6$  y  $r = 1$ .

$$\text{BaxterSchmidtBand}\{3, 1, 2, 3, 4, 5, 6\}, 1, 6]$$

$$\det(B) = 160.$$

**Algoritmo 17.** Fórmula de Widom. Calcula el determinante de matrices de Toeplitz utilizando la fórmula de Widom.

```

DetBandToeplitzWidom[b_, r_, n_, prec_] :=
Module[{Det, pol, long, s, i, k, l, num, a, M0, M, M1, zs, CM, wM},
  long = Length[b];
  s = long - r - 1;
  num = Binomial[r + s, s];
  pol = FromDigits[Reverse[b], z];
  zs = z /. {ToRules[NRoots[pol == 0, z, PrecisionGoal -> prec]++]};
  M0 = Table[i, {i, r + s}];
  M = Subsets[M0, {s}];
  M1 = Table[Complement[M0, M[[i]]], {i, num}];
  wM = Table[0, {i, num}];
  CM = Table[0, {i, num}];
  For[ i = 1, i <= num, i++,
    wM[[i]] = ((-1)^s)*b[[long]]*(Product[zs[[M[[i, j]]]], {j, 1, s}]);
  For[ k = 1, k <= num, k++,
    CM[[k]] = (Product[(zs[[M[[k, j]]]])^r, {j, 1,
      s}])*(Product[(zs[[M[[k, j]]]] - zs[[M1[[k, l]]]])^-1, {j, 1,
      s}, {l, 1, r}]);
  Det = Re[CM.(wM^n)]

```

**Ejemplo 36.** Calcular el determinante de la matriz  $W$  utilizando la fórmula de Widom.

$$W = \begin{pmatrix} 1 & -1 & -2 & -3 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & -1 & -2 & -3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & -1 & -2 & -3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & -1 & -2 & -3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & -1 & -2 & -3 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & -1 & -2 & -3 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \end{pmatrix}$$

Con el programa del algoritmo 17, utilizamos el siguiente comando, indicando los coeficientes del polinomio de Laurent asociado a la matriz,  $r = 3$ ,  $n = 9$  y una precisión de diez dígitos.

```
DetBandToeplitzWidom[{-3, -2, -1, 1, 2}, 3, 9, 10]
```

$$\det(W) = -371,00000$$

**Algoritmo 18.** Matrices aleatorias en formato general.

```

GeneraM[n_] :=
Module[{M, i, j},
  M = Table[0, {n}, {n}];
  For[i = 1, i <= n, i++,
    For[j = 1, j <= n, j++,
      M[[i, j]] = Random[]];
  M]

```

**Ejemplo 37.** Generar una matriz aleatoria de tamaño 4.

*GeneraM[4]//MatrixForm*

$$\begin{pmatrix} 0,0149066 & 0,686984 & 0,211355 & 0,808762 \\ 0,69108 & 0,990112 & 0,475459 & 0,214754 \\ 0,322048 & 0,729927 & 0,396892 & 0,316142 \\ 0,774333 & 0,668959 & 0,368387 & 0,69852 \end{pmatrix}$$

**Algoritmo 19.** Matrices aleatorias en formato de banda.

```

GeneraB[n_, r_, s_] :=
Module[{B, i, j, f},
  B = {};
  For[i = 1, i <= s + 1, i++,
    f = Table[Random[], {Min[r + i, r + s + 1, n]}]; AppendTo[B, f];
  For[j = s + 2, j <= n, j++,
    f = Table[Random[], {Min[s + n - j + 1, r + s + 1, n]}];
    AppendTo[B, f];
  B]

```

**Ejemplo 38.** Generar una matriz aleatoria en formato de banda, con  $n = 4$ ,  $r = 2$  y  $s = 1$ .

*GeneraB[4, 2, 1]//MatrixForm*

$$\begin{pmatrix} \{0,394225, 0,96494, 0,257808\} \\ \{0,248416, 0,542168, 0,511212, 0,0913398\} \\ \{0,702342, 0,527262, 0,824228\} \\ \{0,879985, 0,89358\} \end{pmatrix}$$

## Referencias

- [1] BÖTTCHER, A. AND GRUDSKY, S. M. (2005): Spectral Properties of Banded Toeplitz Matrices. SIAM. ISBN: 978-0-89871-599-6.
- [2] GOLUB, G. H. AND VAN LOAN, C. F. (2013): Matrix Computations, 4th ed. The Johns Hopkins University Press, Baltimore. ISBN 13: 978-4214-0794-4.
- [3] DURBIN, J. (1960): The fitting of time-series models. *Review of the International Statistical Institute*, 28:3, 233–244.