

SEMINARIO DE TESIS I: CÁLCULO INTENSIVO DE LOS CEROS DE LAS SUMAS PARCIALES DE LA FUNCIÓN ZETA DE RIEMANN



Facultad de Ciencias - Escuela Profesional de Ciencia de la
Computación

Tesista: Víctor Racsó Galván Oyola

Asesor: Dr. Oswaldo José Velásquez Castañón

Código de estudiante: 20140097A

4 de agosto de 2021

Índice general

1. Introducción	4
1.1. Objeto de investigación	4
1.2. Objetivos	5
1.2.1. Objetivo general	5
1.2.2. Objetivo inmediato	5
1.2.3. Objetivos específicos	6
1.3. Antecedentes de la investigación	6
2. Análisis teórico	9
2.1. Números complejos	9
2.2. Espacio métrico	11
2.3. Conceptos básicos de topología	12
2.4. Topología y funciones	13
2.4.1. Límites y continuidad	13
2.4.2. Curvas	14
2.4.3. Conjuntos conexos	14
2.5. Funciones holomorfas	15
2.5.1. Integración sobre arcos	15
2.5.2. Analiticidad de una función holomorfa	17
2.5.3. Ceros de una función holomorfa	17
2.6. Principio del argumento	18

2.6.1. Singularidades	18
2.6.2. Teorema del residuo	19
2.6.3. Demostración del principio del argumento	21
2.7. Método confiable para aplicar el principio del argumento	25
2.7.1. Conteo de ceros	26
2.8. Método de bisección en el plano complejo	29
2.9. Algoritmo base	30
3. Análisis implementativo	32
3.1. Lenguaje y librerías	32
3.2. Funciones básicas para variación de argumento	33
3.2.1. <code>getMin(a, b)</code>	33
3.2.2. <code>M(s1, s2, n)</code>	33
3.2.3. <code>isGreater(s1, s2, localM, n)</code>	34
3.2.4. <code>varArg(s1, s2, n)</code>	34
3.3. Funciones básicas para procesamiento de respuestas	36
3.3.1. <code>newton(LD, RU, n)</code>	36
3.4. Pila con exclusión mutua para almacenar los rectángulos	37
3.4.1. Constructor	37
3.4.2. Funciones <code>pop</code> y <code>push</code>	38
3.4.3. Pila con exclusión mutua para almacenar los ceros	39
3.5. Función de inicialización de la pila	40
3.6. Funciones básicas para el procesamiento en los hilos	41
3.6.1. <code>compute(a, n)</code>	41
3.6.2. <code>solve(S, R, id, n)</code>	43
3.7. Script para generación de gráficos	43
4. Resultados	45
4.1. Tabla resumen de los resultados	45
4.2. Gráficas de los ceros encontrados	46

<i>ÍNDICE GENERAL</i>	3
4.3. Resultados modulares en los complejos	50
5. Conclusiones	53
5.1. Conclusión inmediata	53
Bibliografía	55

Capítulo 1

Introducción

1.1. Objeto de investigación

El objeto de estudio de esta investigación es la función zeta de Riemann, función de variable compleja definida por la serie

$$\zeta(s) = \sum_{i=1}^{\infty} i^{-s}, s \in \mathbb{C} \text{ y } \operatorname{Re}(s) > 1$$

que tiene una prolongación analítica a todo el plano complejo \mathbb{C} , como una función meromorfa con un único polo simple en $s = 1$, cuyo residuo es 1.

Se realizará el estudio de esta función, basándonos en la información que nos brindan las llamadas sumas parciales de la función zeta de Riemann, definidas de la siguiente forma:

$$\zeta_n(s) = \sum_{i=1}^n i^{-s}$$

Los llamados ceros no triviales de la función zeta de Riemann, los cuales pertenecen a la franja:

$$\{s \in \mathbb{C} : 0 \leq \operatorname{Re}(s) \leq 1\}$$

guardan información importante sobre la distribución de los números primos. Por ejemplo, la función:

$$\pi(x) = |\{p \leq x : p \text{ es un primo positivo}\}|$$

por el teorema de Hadamard y de la Vallée Poussin (1896) cumple que:

$$\lim_{x \rightarrow \infty} \frac{\pi(x) \log x}{x} = 1$$

de modo que:

$$\pi(x) \approx \frac{x}{\log x}$$

para x suficientemente grande.

Esto es equivalente a la no existencia de ceros de la función zeta de Riemann en la recta $\operatorname{Re}(s) = 1$. La hipótesis de Riemann establece que todos los ceros no triviales de la función ζ pertenecen a la recta $\operatorname{Re}(s) = \frac{1}{2}$.

En este primer seminario, estableceremos un método general para el cálculo de ceros de funciones holomorfas con ceros simples.

1.2. Objetivos

1.2.1. Objetivo general

Estudiar la distribución de los ceros de las sumas parciales de la función zeta de Riemann, considerando las primeras instancias $n \in [3, 9]$.

1.2.2. Objetivo inmediato

Implementar una versión paralelizada de algún algoritmo de cálculo de los ceros de las sumas parciales de la función zeta de Riemann.

1.2.3. Objetivos específicos

Analizar gráficamente la distribución de la nube de ceros de las sumas parciales de la función

1.3. Antecedentes de la investigación

La conexión entre la función zeta de Riemann y los números primos, se debe al producto que Euler demostró para s real mayor que 1:

$$\sum_{i=1}^{\infty} i^{-s} = \prod_p \left(1 - \frac{1}{p^s}\right)^{-1}$$

el mismo Euler usó esta identidad para dar una prueba analítica de la infinitud de los números primos. Riemann fue el primero en considerar esta identidad en términos de variable compleja, se demostró la ecuación funcional

$$\zeta(s) = 2^s \pi^{s-1} \sin\left(\frac{\pi s}{2}\right) \Gamma(1-s) \zeta(1-s)$$

donde Γ es la función gamma de Euler.

El problema de corroborar o descartar la hipótesis de Riemann, planteado por él en 1859 y propuesto formalmente por David Hilbert el año 1900, como el octavo problema de su lista de 23 problemas abiertos hasta entonces, indican la dificultad y trascendencia de resolver esta conjetura. Transcurridos ya más de cien años, con muchos intentos infructuosos llegamos a mayo del 2000, fecha en la que el instituto norteamericano Clay relanza el reto, dentro de su lista de los siete problemas del milenio, ofreciendo un millón de dólares al que resuelve. La sucesión de funciones que se analizará en este trabajo son las sumas parciales de la función zeta de Riemann, esto es

$$\zeta_n(s) = \sum_{i=1}^n i^{-s}, \text{ con } \operatorname{Re}(s) > 1$$

estas son funciones casi-periódicas (Hernández, 2009).

Paul Turán (Turán, 1948) demostró que la hipótesis de Riemann será cierta, si para todo n suficientemente grande, $\zeta_n(s)$ no tendría ceros en $\text{Re}(s) > 1$, Turán también demostró que $\zeta_n(s)$ en $\text{Re}(s) > 1$ no tiene ceros para $1 \leq n \leq 5$. Robert Spira demuestra que $\zeta_n(s)$ en $\text{Re}(s) > 1$ no tiene ceros para $6 \leq n \leq 9$ (Spira, 1966), y en un trabajo posterior Spira demuestra que si $\zeta_n(s)$ tiene un cero en $\text{Re}(s) > 1$, entonces tiene infinitos ceros.

Si

$$b_n = \sup \{ \text{Re}(s) : \zeta_n(s) = 0 \}$$

H.L.Montgomery (Montgomery, 1983) demostró que

$$b_n = 1 + \left(\frac{4}{\pi} - 1 + o(1) \right) \frac{\log(\log n)}{\log n}$$

Si

$$a_n = \inf \{ \text{Re}(s) : \zeta_n(s) = 0 \}$$

M. Balazard y O. Velásquez Castañón (Balazard and Castanón, 2009) demostraron que

$$a_n = -n \log 2 + o(n)$$

Un resultado clave en el estudio tanto de a_n como de b_n es el teorema de equivalencia de Bohr, que permite torcer los coeficientes de un polinomio de Dirichlet tal como $\zeta_n(s)$ (pero incluso de una serie) para estudiar de manera más sencilla sus propiedades (Dubon, 2015).

En un trabajo posterior Spira demuestra que $\zeta_n(s)$ en $\text{Re}(s) > 1$ (Spira, 1968) tiene ceros para:

$$n = 19, 22 \leq n \leq 27, 29 \leq n \leq 50$$

Por su parte, W.R Monach (Monach, 1980) demuestra que para todo $n > 50$ existen ceros de $\zeta_n(s)$ en $\operatorname{Re}(s) > 1$. Cerrando este enfoque D.J.Platt y T.S. Trudgian en el año 2016 demuestran que $\zeta_n(s)$ en $\operatorname{Re}(s) > 1$ no tiene ceros para

$$10 \leq n \leq 18, \quad n = 20, 21, 28.$$

Teorema 1.1. *Para $1 \leq n \leq 18$, $n = 20, 21, 28$ no hay ceros de $\zeta_n(s)$ en $\operatorname{Re}(s) > 1$. Para los demás enteros positivos n existen infinitos ceros en dicha región.*

Capítulo 2

Análisis teórico

2.1. Números complejos

Definición 2.1. *Definimos el conjunto de los números complejos como el conjunto:*

$$\mathbb{C} = \{(a, b) : a, b \in \mathbb{R}\}$$

provisto de las operaciones binarias de suma y producto:

$$+ : \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C}$$

$$((a, b), (c, d)) \mapsto (a, b) + (c, d) = (a + c, b + d)$$

$$\cdot : \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C}$$

$$((a, b), (c, d)) \mapsto (a, b) \cdot (c, d) = (ac - bd, ad + bc)$$

Se puede demostrar que el conjunto de los números complejos cumple con las propiedades de un cuerpo conmutativo:

- Asociatividad de la suma: $\forall z_1, z_2, z_3 \in \mathbb{C}, (z_1 + z_2) + z_3 = z_1 + (z_2 + z_3)$.
- Conmutatividad de la suma: $\forall z_1, z_2 \in \mathbb{C}, z_1 + z_2 = z_2 + z_1$.
- Existencia del cero: $0 = (0, 0) \rightarrow \forall z \in \mathbb{C}, z + 0 = 0 + z = z$.

- Neutro aditivo: $\forall z = (a, b), \exists w = (-a, -b) \in \mathbb{C} : z + w = w + z = 0$. w se denota como $-z$.
- Asociatividad del producto: $\forall z_1, z_2, z_3 \in \mathbb{C}, (z_1 \cdot z_2) \cdot z_3 = z_1 \cdot (z_2 \cdot z_3)$.
- Conmutatividad del producto: $\forall z_1, z_2 \in \mathbb{C}, z_1 \cdot z_2 = z_2 \cdot z_1$.
- Existencia de la unidad: $1 = (1, 0) \rightarrow \forall z \in \mathbb{C}, z \cdot 1 = 1 \cdot z = z$.
- Inverso multiplicativo: $\forall z \in \mathbb{C}^* = \mathbb{C} \setminus \{0\}, z = (a, b)$, el elemento $w = \left(\frac{a}{a^2+b^2}, \frac{-b}{a^2+b^2}\right)$ cumple con $z \cdot w = w \cdot z = 1$. w se denota como z^{-1} .
- Distributividad del producto en la suma: $\forall z_1, z_2, z_3 \in \mathbb{C}, z_1 \cdot (z_2 + z_3) = z_1 \cdot z_2 + z_1 \cdot z_3$.

El conjunto $\{(a, 0) : a \in \mathbb{R}\}$ es un subcuerpo de \mathbb{C} isomorfo a \mathbb{R} mediante el homomorfismo $\mathbb{R} \rightarrow \mathbb{C}, a \mapsto (a, 0)$. Escribimos $(a, 0) \equiv a$ y $i = (0, 1)$:

$$z = (a, b) = (a, 0) + (0, 1) \cdot (b, 0) = a + ib$$

Lo que nos permite usar la notación:

$$\mathbb{C} = \{a + ib : a, b \in \mathbb{R}\}$$

En donde cada componente sigue las reglas algebraicas de \mathbb{R} y además $i^2 = -1$.

Definición 2.2 (Argumento principal). *Definimos el argumento principal de $z = (a, b) = a + ib$ como el ángulo θ que se forma entre el vector que representa a z en el plano complejo y el eje real tal que $\theta \in [0, 2\pi)$.*

Definición 2.3 (Conjugado). *Definimos el conjugado de $z = (a, b) = a + ib$ como:*

$$\bar{z} = (a, -b) = a - ib$$

Definición 2.4 (Módulo). *Definimos el módulo de $z = (a, b) = a + ib$ $|\cdot| : \mathbb{C} \rightarrow \mathbb{R}$ como:*

$$|z| = |(a, b)| = |a + ib| = \sqrt{a^2 + b^2}$$

Dadas las definiciones podemos relacionar el inverso multiplicativo:

$$z^{-1} = \frac{\bar{z}}{|z|}, \forall z \in \mathbb{C} \setminus \{0\}$$

Definición 2.5 (Parte real e imaginaria de un complejo). *Dado un número complejo $z = a + ib$, definimos a sus partes real e imaginaria como:*

$$\operatorname{Re}(z) = a \quad \operatorname{Im}(z) = b$$

De manera que:

$$z = \operatorname{Re}(z) + i \cdot \operatorname{Im}(z)$$

Definición 2.6 (Representación polar). *La representación de un número $z \in \mathbb{C}$ como*

$$z = r e^{i\theta}$$

donde $r = |z| \geq 0$ y θ es el argumento principal de z es llamada representación polar de z .

2.2. Espacio métrico

Definición 2.7 (Métrica. Espacio métrico). *Sea X un conjunto. Definimos como una métrica sobre X a una aplicación $d : X \times X \rightarrow \mathbb{R}$ que satisface las siguientes propiedades:*

- $\forall x, y \in X, d(x, y) \geq 0$. Además, $d(x, y) = 0 \iff x = y$.
- $\forall x, y \in X, d(x, y) = d(y, x)$.
- $\forall x, y, z \in X, d(x, z) \leq d(x, y) + d(y, z)$.

El par (X, d) es llamado espacio métrico.

2.3. Conceptos básicos de topología

Definición 2.8 (Disco abierto). *Sea un espacio métrico (X, d) , definimos como disco abierto en $a \in X$ con radio $r > 0$ al conjunto:*

$$D(a, r) = \{x \in X : d(a, x) < r\}$$

Definición 2.9 (Punto interior. Interior). *Sea (X, d) un espacio métrico. Decimos que $a \in X$ es un punto interior de $A \subset X$ si existe $\varepsilon > 0$ tal que $D(a, \varepsilon) \subset A$. El conjunto de puntos interiores de A se denomina interior de A y se denota $\text{int}(A)$. Se puede probar que $\text{int}(A) \subset A$.*

Definición 2.10 (Conjunto abierto). *Un subconjunto $A \subset X$ de un espacio métrico (X, d) es abierto si todo punto de A es punto interior de A . Equivalentemente, $\text{int}(A) = A$.*

Definición 2.11 (Punto de adherencia. Clausura). *Sea (X, d) un espacio métrico y $A \subset X$. El punto $x \in X$ es un punto de adherencia de A si, $\forall \varepsilon > 0$:*

$$D(x, \varepsilon) \cap A \neq \emptyset$$

El conjunto de puntos de adherencia de A se denomina clausura de A y se denota \overline{A} (No confundir con la denotación de módulo).

Se puede probar que $A \subset \overline{A}$.

Definición 2.12 (Conjunto cerrado). *Un conjunto $A \subset X$ de un espacio métrico (X, d) es cerrado si todo punto de adherencia pertenece a A . Equivalentemente, $\overline{A} \subset A$ y $A \subset \overline{A}$, por lo que $\overline{A} = A$.*

En un espacio métrico (X, d) denotaremos por:

$$D'(a, r) = \{x \in X : 0 < d(a, x) < r\}$$

el disco abierto con centro en $a \in X$ y de radio $r > 0$ pero desprovisto de a . Es evidente que $D'(a, r) = D(a, r) \setminus \{a\}$.

Definición 2.13 (Punto de acumulación). *Sea (X, d) un espacio métrico. Se define al punto $a \in X$ como punto de acumulación del conjunto $A \subset X$ si para todo $\varepsilon > 0$:*

$$D'(a, \varepsilon) \cap A \neq \emptyset$$

Al conjunto de puntos de acumulación de A se le denota como A' .

2.4. Topología y funciones

2.4.1. Límites y continuidad

Definición 2.14 (Límite de una función en \mathbb{C}). *Sea $A \subset \mathbb{C}$ y z_0 un punto de acumulación de A . El límite de una función $f : A \rightarrow \mathbb{C}$ a medida que z tiende a $z_0 \in A$ es L , denotado por $\lim_{z \rightarrow z_0} f(z) = L$ si:*

$$\forall \varepsilon > 0, \exists \delta > 0 : \forall z \in A, 0 < |z - z_0| < \delta \longrightarrow |f(z) - L| < \varepsilon$$

Definición 2.15 (Continuidad de una función en \mathbb{C}). *Sea $A \subset \mathbb{C}$ y $z_0 \in A$. Decimos que la función $f : A \rightarrow \mathbb{C}$ es continua en $z_0 \in A$ si:*

$$\forall \varepsilon > 0, \exists \delta > 0 : \forall z \in A, |z - z_0| < \delta \longrightarrow |f(z) - f(z_0)| < \varepsilon$$

Decimos que f es continua en A si f es continua en todos los puntos $z \in A$.

Denotamos el conjunto de todas las funciones continuas en A como:

$$C(A) = \{f : A \rightarrow \mathbb{C} : f \text{ es continua en } A\}$$

Definición 2.16 (Continuidad uniforme). *Sea $A \subset \mathbb{C}$. Decimos que la función $f : A \rightarrow \mathbb{C}$ es uniformemente continua en A si:*

$$\forall \varepsilon > 0, \exists \delta > 0 : \forall x, y \in A, |x - y| < \delta \longrightarrow |f(x) - f(y)| < \varepsilon$$

2.4.2. Curvas

Definición 2.17 (Curva). *Una curva en el espacio métrico (X, d) es una aplicación $\alpha : [a, b] \rightarrow X$ continua. La imagen de la curva α es el conjunto $\alpha^* = \alpha([a, b])$.*

Si $\alpha(a) = \alpha(b)$ decimos que la curva es cerrada.

Dados $z_1, z_2 \in X$ en un espacio métrico (X, d) , definimos $\alpha : [0, 1] \rightarrow X$ mediante $\alpha(t) = t \cdot z_1 + (1 - t)z_2$. La función α es una curva y su imagen es el segmento que une a z_1 y z_2 en X . Denotaremos a la imagen de esta curva en particular como:

$$[z_1, z_2] = \alpha^* = \{t \cdot z_1 + (1 - t)z_2, t \in [0, 1]\}$$

Dadas las características de los valores a y b , no debe haber confusión al usar $[a, b]$ para intervalos en los números reales ($a, b \in \mathbb{R}$) o segmentos en el plano complejo ($a, b \in \mathbb{C}$).

Definición 2.18 (Reparametrización). *Dada una curva $\alpha : [a, b] \rightarrow X$ en un espacio métrico (X, d) , una reparametrización de α es una curva $\beta : [c, d] \rightarrow X$ de la forma $\beta = \alpha \circ s$, donde $s : [a, b] \rightarrow [c, d]$ es una función continua y biyectiva.*

Definición 2.19 (Curva Poligonal). *Una curva poligonal en A es una función continua $\gamma : [a, b] \rightarrow A$ para la cual existe una partición*

$$a = a_0 < a_1 < a_2 < \dots < a_{n-1} < a_n = b$$

tal que $\gamma|_{[a_{i-1}, a_i]}$ (la restricción de γ a $[a_{i-1}, a_i]$) es lineal para todo $i = 1, 2, \dots, n$.

2.4.3. Conjuntos conexos

Definición 2.20 (Conjunto conexo). *Sea (X, d) un espacio métrico. Decimos que $Y \subset X$ es conexo si, dados dos conjuntos abiertos relativos a Y , $A \subset Y$ y $B \subset Y$, tales que $Y = A \cup B$ y $A \cap B = \emptyset$, entonces se tiene forzosamente que $A = \emptyset$ o $B = \emptyset$.*

Proposición 2.21. *Sean (X, d_X) y (Y, d_Y) espacios métricos, $A \subset X$ y $f : A \rightarrow Y$. Si f es continua y $C \in A$ es conexo, entonces $f(C)$ es conexo.*

Definición 2.22 (Conexo por caminos). *Un subconjunto $C \subset X$ de un espacio métrico (X, d) es conexo por caminos si, dados $x, y \in C$ cualesquiera, existe una curva $\alpha : [a, b] \rightarrow C$ tal que $\alpha(a) = x$ y $\alpha(b) = y$.*

Definición 2.23 (Conjunto convexo). *Un subconjunto $C \subset X$ de un espacio métrico (X, d) es convexo si, dados $x, y \in X$, se tiene que $[x, y] \in C$.*

2.5. Funciones holomorfas

Definición 2.24 (Derivada de una función en \mathbb{C}). *Sea $A \subset \mathbb{C}$ y $z_0 \in A$ un punto de acumulación de A . Decimos que la función $f : A \rightarrow \mathbb{C}$ es diferenciable en a si el siguiente límite existe:*

$$f'(z_0) = \lim_{z \rightarrow z_0} \frac{f(z) - f(z_0)}{z - z_0}$$

donde $f'(z_0)$ es llamada la derivada de f en el punto z_0 .

Definición 2.25 (Función holomorfa o analítica). *Sea un conjunto abierto no vacío $A \subset \mathbb{C}$. Decimos que la función $f : A \rightarrow \mathbb{C}$ es holomorfa o analítica si es diferenciable en todos los puntos de A .*

Denotamos como $H(A)$ a la familia de funciones holomorfas en el dominio A , es decir:

$$H(A) = \{f : A \rightarrow \mathbb{C} : f \text{ es holomorfa en } A\}$$

2.5.1. Integración sobre arcos

Definición 2.26 (Arco). *Una curva $\gamma : [a, b] \rightarrow \mathbb{C}$ es un arco si γ es continuamente diferenciable excepto, tal vez, en un número finito de puntos $t_i \in [a, b]$, donde existen derivadas izquierda y derecha.*

Definición 2.27. *Para la función $g : [a, b] \rightarrow \mathbb{C}$, $g \in C([a, b])$, definimos la integral:*

$$\int_a^b g(t)dt = \int_a^b \operatorname{Re}(g(t))dt + i \int_a^b \operatorname{Im}(g(t))dt$$

donde las integrales del lado derecho son integrales de Riemann de funciones reales de variable real.

Si g es continua en $[a, b]$ excepto en un conjunto finito de puntos $a = t_0 < t_1 < \dots < t_n = b$, que pueden o no incluir a a o b , y g posee límites izquierdo y derecho, definimos

$$\int_a^b g(t)dt = \sum_{i=1}^n \int_{t_{i-1}}^{t_i} g(t)dt$$

Definición 2.28 (Integral a lo largo de un arco). Para un arco $\gamma : [a, b] \rightarrow \mathbb{C}$ y $f \in C(\gamma^*)$, definimos la integral de f a lo largo del arco γ mediante

$$\int_{\gamma} f(z)dz = \int_a^b f(\gamma(t))\gamma'(t)dt$$

Definición 2.29 (Índice). Dado un arco cerrado $\gamma : [a, b] \rightarrow \mathbb{C}$ y $z \notin \gamma^*$, definimos el índice del arco γ con respecto a z mediante

$$\operatorname{ind}_{\gamma}(z) = \frac{1}{2\pi i} \int_{\gamma} \frac{dw}{w - z} \quad (2.1)$$

Definimos el índice del arco γ como la función

$$\operatorname{ind}_{\gamma} : \mathbb{C} \setminus \gamma^* \rightarrow \mathbb{C}$$

Si $\gamma : [a, b] \rightarrow \mathbb{C}$, 2.1 se escribe explícitamente

$$\operatorname{ind}_{\gamma}(z) = \frac{1}{2\pi i} \int_a^b \frac{\gamma'(t)}{\gamma(t) - z} dt$$

2.5.2. Analiticidad de una función holomorfa

Teorema 2.30 (Representación local de una función holomorfa). *Sea A abierto en \mathbb{C} y $f \in H(A)$, entonces f se puede representar localmente por una serie de potencias de la siguiente manera. Si $z_0 \in A$ y para algún $r > 0$, $\overline{D(z_0, r)} \in A$ entonces*

$$\forall z \in D(z_0, r), f(z) = \sum_{n=0}^{\infty} c_n (z - z_0)^n, \quad (2.2)$$

donde los coeficientes c_n están dados por

$$\forall n \geq 0, c_n = \frac{1}{2\pi i} \int_{\partial D(z_0, r)} \frac{f(w)}{(w - z_0)^{n+1}} dw \quad (2.3)$$

Teorema 2.31 (Teorema local de Cauchy). *Supongamos que $F \in H(A)$ y $F' \in C(A)$. Entonces*

$$\int_{\gamma} F'(z) dz = 0$$

Para todo arco cerrado γ en A .

Teorema 2.32 (Teorema de Cauchy). *Sea $f \in H(A)$, A abierto en \mathbb{C} y γ un ciclo en A que satisfice*

$$\text{ind}_{\gamma}(a) = 0, \forall a \notin A$$

entonces

$$f(z) \text{ind}_{\gamma}(z) = \frac{1}{2\pi i} \int_{\gamma} \frac{f(w)}{w - z} dw, \forall w \in A \setminus \gamma^*$$

$$\int_{\gamma} f(z) dz = 0.$$

Proposición 2.33. *Si $A \subset \mathbb{C}$ es abierto y conexo, $f \in H(A)$, $z_0 \in A$ y $f^{(k)}(z_0) = 0$ para todo $k \geq 0$, entonces $f(z) = 0$, $\forall z \in A$.*

2.5.3. Ceros de una función holomorfa

Definición 2.34. *Sea $A \subset \mathbb{C}$ y $f : A \rightarrow \mathbb{C}$, denotamos los ceros de f en A como:*

$$N(f) = \{a \in A : f(a) = 0\}$$

Teorema 2.35. *Sea A abierto y conexo y $f \in H(A)$, entonces $N(A) = A$ o $N(A)$ no tiene puntos de acumulación en A .*

En este último caso, para cada $z_0 \in N(f)$ existe un entero $m \geq m(z_0) \geq 1$ tal que:

$$f(z) = (z - z_0)^m g(z), \forall z \in A \quad (2.4)$$

Donde $g \in H(A)$ y $g(z_0) \neq 0$.

Definición 2.36 (Orden de un cero). *Dada $f \in H(A)$ no idénticamente nula y $z_0 \in N(f)$, el único entero $m \geq m(z_0) \geq 1$ tal que se cumple (2.4) con las condiciones dadas sobre g en el teorema, es llamado multiplicidad u orden del z_0 como cero de f .*

2.6. Principio del argumento

2.6.1. Singularidades

Definición 2.37 (Singularidad aislada). *Si $z_0 \in A$ y $f \in H(A \setminus \{z_0\})$, decimos que f tiene una singularidad aislada en z_0 .*

Decimos que la singularidad es removible si existe $g \in H(A)$ tal que $g|_{A \setminus \{z_0\}} = f$.

Teorema 2.38. *Si $f \in H(A \setminus \{z_0\})$ y f está acotada en*

$$D'(z_0, r) = \{z \in \mathbb{C} : 0 < |z - z_0| < r\}$$

para algún $r > 0$, entonces f tiene una singularidad removible en z_0 .

Teorema 2.39 (Clasificación de singularidades). *Si $z_0 \in A$ y $f \in H(A \setminus \{z_0\})$, entonces únicamente uno de los siguientes casos debe ocurrir:*

1. *f tiene una singularidad removible en z_0 .*
2. *Existe $\{c_1, c_2, \dots, c_m\} \subset \mathbb{C}$, $m \geq 1$, con $c_m \neq 0$, tal que:*

$$f(z) - \sum_{k=1}^m \frac{c_k}{(z - z_0)^k}$$

tiene una singularidad removible en z_0 .

3. Si $r > 0$ y $D(z_0, r) \subset A$, entonces

$$\overline{f(D'(z_0, r))} = \mathbb{C}$$

Definición 2.40 (Polo). Sean $z_0 \in A$ y $f \in H(A \setminus \{z_0\})$. Siguiendo las notaciones y casos del teorema 2.39:

I) En el caso 2, se dice que f tiene un polo de orden m en z_0 y c_1 se denomina residuo de f en z_0 y se le denota por $\text{Res}(f, z_0)$. Además

$$\sum_{k=1}^m \frac{c_k}{(z - z_0)^k}$$

es la parte principal de f en z_0 . En el caso particular de $m = 1$, decimos que z_0 es un polo simple de f .

Además, se cumple que:

$$f(z) = \sum_{k=1}^m \frac{c_k}{(z - z_0)^k} + h(z)$$

Donde h es holomorfa en una vecindad de z_0 .

Observación: La función f puede ser escrita como una serie de Laurent de la siguiente forma:

$$f(z) = \sum_{k=-m}^{\infty} c_k (z - z_0)^k = (z - z_0)^{-m} \sum_{k=0}^{\infty} c_{k-m} (z - z_0)^k$$

En cuyo caso, el valor c_{-1} es el residuo $\text{Res}(f, z_0)$.

II) En el caso 3 decimos que f tiene una singularidad esencial en z_0 .

2.6.2. Teorema del residuo

Definición 2.41 (Función meromorfa). Si A es abierto, decimos que f es una función meromorfa en A si existe $B \subset A$ tal que:

- B no tiene puntos de acumulación en A .
- $f \in H(A \setminus B)$.
- f tiene un polo en cada punto de B .

El conjunto B se denomina conjunto de polos de f y se denota por $P(f)$. El conjunto de funciones meromorfas en A se denota:

$$M(A) = \{f : f \text{ es meromorfa en } A\}$$

Teorema 2.42 (Teorema del residuo). Sea f una función meromorfa en A y B el conjunto de polos de f . Si γ es un ciclo en $A \setminus B$ tal que:

$$\text{ind}_\gamma(a) = 0, \forall a \notin A$$

entonces:

$$\frac{1}{2\pi i} \int_\gamma f(z) dz = \sum_{a \in A} \text{Res}(f, a) \text{ind}_\gamma(a)$$

Demostración. Primero consideremos que $Q(z)$ es la parte principal de f en un punto z_0 , como en la definición de polo, entonces:

$$\frac{1}{2\pi i} \int_\gamma Q(z) dz = \frac{1}{2\pi i} \sum_{k=-m}^{\infty} c_k \int_\gamma (z - z_0)^k dz$$

Por el teorema local de Cauchy, recordamos que si $f(z) = \frac{1}{n+1} z^{n+1}$ su derivada es $f'(z) = z^n$, por lo que su integral en toda curva cerrada γ es 0 si y solo si $n \neq -1$ y $0 \neq \gamma^*$.

$$\frac{1}{2\pi i} \int_\gamma Q(z) dz = c_{-1} \cdot \left(\frac{1}{2\pi i} \int_\gamma (z - z_0)^{-1} dz \right) = \text{Res}(f, z_0) \text{ind}_\gamma(z_0)$$

Sea $B = \{z_1, \dots, z_m\}$ y sea $Q_i(z)$ la parte principal de f en z_i para cada $i = 1, \dots, m$.

Entonces, $g = f - \sum_{i=1}^m Q_i$ posee singularidades removibles en a_i para cada $i = 1, \dots, m$.

Sea $A_0 = A \setminus (P(f) \setminus B)$, entonces A_0 es abierto y para cada $a \notin A_0$ se da que: o bien $a \notin A$, y por la hipótesis del teorema tenemos que $\text{ind}_\gamma(a) = 0$ o bien $a \in P(f) \setminus B$, en cuyo caso también tenemos que $\text{ind}_\gamma(a) = 0$ por definición de B . Por el teorema de Cauchy:

$$\int_{\gamma} g(z) dz = 0$$

Lo que equivale a que:

$$\frac{1}{2\pi i} \int_{\gamma} f(z) dz = \frac{1}{2\pi i} \sum_{i=1}^m \int_{\gamma} Q_i(z) dz = \sum_{i=1}^m \text{Res}(f, z_i) \text{ind}_\gamma(z_i)$$

□

2.6.3. Demostración del principio del argumento

Sean:

- γ una curva cerrada y simple en sentido antihorario.
- $f(z)$ holomorfa en γ excepto por sus posibles polos y ceros que están estrictamente dentro (la curva no los atraviesa) de γ .
- z_i y p_i los ceros y polos de la función f dentro de γ .
- $\text{ord}(z_i)$ el orden del cero z_i y $\text{ord}(p_i)$ el orden del polo p_i .

Teorema 2.43. *Con las anteriores consideraciones, se cumple que:*

$$\int_{\gamma} \frac{f'(z)}{f(z)} dz = 2\pi i \left(\sum \text{ord}(z_i) - \sum \text{ord}(p_i) \right)$$

Demostración. Consideremos un cero z_i tal que tenga orden m , entonces:

$$f(z) = (z - z_i)^m g(z)$$

Además g es holomorfa y no nula en una vecindad de z_i . Si derivamos la función:

$$f'(z) = m(z - z_i)^{m-1}g(z) + (z - z_i)^m g'(z)$$

$$\frac{f'(z)}{f(z)} = \frac{m(z - z_i)^{m-1}g(z) + (z - z_i)^m g'(z)}{(z - z_i)^m g(z)} = \frac{m}{z - z_i} + \frac{g'(z)}{g(z)}$$

Dado que $g(z)$ es no nula en una vecindad de z_i , entonces $\frac{g'(z)}{g(z)}$ es holomorfa alrededor de z_i . Esto implica directamente que z_i es un polo simple de $\frac{f'(z)}{f(z)}$, por lo que:

$$\text{Res}\left(\frac{f'(z)}{f(z)}, z_i\right) = m = \text{ord}(z_i)$$

De manera análoga, se puede concluir que si p_i es un polo de f con orden m , entonces:

$$\text{Res}\left(\frac{f'(z)}{f(z)}, p_i\right) = -m = -\text{ord}(p_i)$$

Aplicando 2.42 sobre todo el dominio, obtenemos que:

$$\int_{\gamma} \frac{f'(z)}{f(z)} dz = 2\pi i \left(\sum \text{Res}\left(\frac{f'(z)}{f(z)}, z_i\right) + \sum \text{Res}\left(\frac{f'(z)}{f(z)}, p_i\right) \right) = 2\pi i (\sum \text{ord}(z_i) - \sum \text{ord}(p_i))$$

□

Teorema 2.44 (Principio del argumento). *Bajo las condiciones iniciales:*

$$\int_{\gamma} \frac{f'(z)}{f(z)} dz = 2\pi i \text{ind}(f \circ \gamma, 0)$$

Demostración. Aplicando 2.43 obtenemos que:

$$\int_{\gamma} \frac{f'(z)}{f(z)} dz = 2\pi i (\sum \text{ord}(z_i) - \sum \text{ord}(p_i))$$

Si definimos $w = f(z)$, considerando que $dw = f'(z)dz$, notamos que en primer lugar se cumple que:

$$z = \gamma(t), \quad f(\gamma(t)) = f(z) = w \neq 0$$

Por lo tanto, no hay ninguna indeterminación con el término $\frac{1}{w}$. Esto nos lleva a que:

$$\int_{\gamma} \frac{f'(z)}{f(z)} dz = \int_{f \circ \gamma} \frac{dw}{w} = 2\pi i \operatorname{ind}(f \circ \gamma, 0)$$

Donde la última igualdad es por la definición de índice. □

Logaritmo complejo

Definición 2.45 (Rama principal del logaritmo). *Definimos la rama principal del logaritmo como la función $\log : \mathbb{C} \setminus (-\infty, 0] \rightarrow \mathbb{C}$ que asigna a todo número complejo $z \in \mathbb{C} \setminus (-\infty, 0]$ el número:*

$$\log(z) = \ln |z| + i \arg(z)$$

Teorema 2.46 (Relación entre los ceros de una función holomorfa y la variación de su argumento en una curva cerrada). *Bajo las condiciones iniciales, se cumple que:*

$$\frac{\Delta \arg_{f \circ \gamma}}{2\pi} = \frac{1}{2\pi i} \int_{f \circ \gamma} \frac{f'(z)}{f(z)} dz$$

Donde $\Delta \arg_g$ es la variación del argumento a lo largo de la función g .

Demostración. Debido a que γ no tiene ningún cero de f , el logaritmo complejo está bien definido a lo largo de toda la curva evaluada en la función, por lo tanto:

$$\log(f(z)) \text{ existe, } \forall z = \gamma(t)$$

Debido a que la función logaritmo complejo cumple con las ecuaciones de Cauchy-Riemann, es diferenciable, por lo que podemos aplicar:

$$\log(f(z))' = \frac{f'(z)}{f(z)} \tag{2.5}$$

Si integramos a lo largo de la curva γ :

$$\int_{\gamma} \log(f(z))' dz = \int_{\gamma} \frac{f'(z)}{f(z)} dz \quad (2.6)$$

$$\log(f(z)) \Big|_{\gamma} = \int_{\gamma} \frac{f'(z)}{f(z)} dz \quad (2.7)$$

Notemos que no es sencillo evaluar el logaritmo a lo largo de la curva γ , por lo que plantearemos una partición p de $(n+1)$ puntos a lo largo de la misma con la finalidad de que se cumpla lo siguiente:

$$-\pi < \Delta \arg(p_i, p_{i+1}) \leq \pi \wedge p_{n+1} = p_1$$

Por lo tanto, al aplicar la sumatoria sobre la partición, obtendremos:

$$\log(|f(p_i)|) + i \cdot \arg(f(p_i)) - \log(|f(p_{i-1})|) - i \cdot \arg(f(p_{i-1})), \forall i = 2, \dots, (n+1)$$

Reacomodando el aporte de cada segmento:

$$\log \left(\left| \frac{f(p_i)}{f(p_{i-1})} \right| \right) + i (\arg(f(p_i)) - \arg(f(p_{i-1})))$$

Sin embargo, debemos recordar que $\arg(a) - \arg(b) = \arg\left(\frac{a}{b}\right)$ si $b \neq 0$:

$$\log \left(\left| \frac{f(p_i)}{f(p_{i-1})} \right| \right) + i \cdot \arg \left(\frac{f(p_i)}{f(p_{i-1})} \right)$$

Consideremos la suma de todos los aportes:

$$\log(f(z)) \Big|_{\gamma} = \sum_{i=2}^{n+1} \log \left(\left| \frac{f(p_i)}{f(p_{i-1})} \right| \right) + i \cdot \arg \left(\frac{f(p_i)}{f(p_{i-1})} \right)$$

Notemos que para el logaritmo se irán cancelando los denominadores y numeradores de posiciones consecutivas, para finalmente dejarnos con:

$$\log(f(z)) \Big|_{\gamma} = \log \left(\left| \frac{f(p_{n+1})}{f(p_1)} \right| \right) + \sum_{i=2}^{n+1} i \cdot \arg \left(\frac{f(p_i)}{f(p_{i-1})} \right)$$

Sin embargo, por la definición de la partición $p_{n+1} = p_1$:

$$\log(f(z))\Big|_{\gamma} = \log(1) + \sum_{i=2}^{n+1} i \cdot \arg\left(\frac{f(p_i)}{f(p_{i-1})}\right) = \sum_{i=2}^{n+1} i \cdot \arg\left(\frac{f(p_i)}{f(p_{i-1})}\right) = i \cdot \Delta \arg_{f \circ \gamma} \quad (2.8)$$

Reemplazando 2.8 en 2.7:

$$i \Delta \arg_{f \circ \gamma} = \int_{\gamma} \frac{f'(z)}{f(z)} dz \quad (2.9)$$

Si dividimos entre $2\pi i$ obtenemos finalmente:

$$\frac{\Delta \arg_{f \circ \gamma}}{2\pi} = \frac{1}{2\pi i} \int_{\gamma} \frac{f'(z)}{f(z)} dz \quad (2.10)$$

□

Si aplicamos el resultado de 2.42 en 2.46, llegamos a la conclusión final de esta sección:

$$\frac{\Delta \arg_{f \circ \gamma}}{2\pi} = \frac{1}{2\pi i} \int_{\gamma} \frac{f'(z)}{f(z)} dz = (\sum \text{ord}(z_i) - \sum \text{ord}(p_i)) \quad (2.11)$$

2.7. Método confiable para aplicar el principio del argumento

Un resultado de Ying [13] establece que un número finito de valores de una función y sus derivadas no es suficiente para determinar el número de ceros de una función holomorfa, o incluso un polinomio (a menos que se conozca el grado), en una región acotada. La clave de un algoritmo confiable para el cálculo de los ceros es el uso de datos globales sobre la función dada. El algoritmo de Ying y Katz utilizado necesita una cota superior de la segunda derivada de la función a tratar sobre todo intervalo fijo del plano, una condición no muy exigente (sobretudo para las funciones de clases particulares como los polinomios exponenciales). Presentaremos y completaremos

aquí el método de Ying y Katz [13], este método nos proporcionará la información precisa sobre el número de ceros de una función holomorfa sobre una región acotada para aislar y calcular los ceros de nuestras funciones.

2.7.1. Conteo de ceros

Por el principio del argumento, si ∂R es una curva poligonal simple y recorrida en sentido antihorario, el número de ceros de $f(s)$ en R , denotado por $N(R)$, es:

$$N(R) = \frac{1}{2\pi} \Delta_{\partial R} \arg f(s)$$

Disponemos de una función argumento \arg con valores en el intervalo $(-\frac{\pi}{2}, \frac{\pi}{2})$ de forma que, para dos complejos s_1 y s_2 , se cumpla que $|\Delta_{[s_1, s_2]} \arg f(s)| < \pi$ y podamos calcular fácilmente:

$$\Delta_{[s_1, s_2]} \arg f(s) = \arg \left(\frac{f(s_2)}{f(s_1)} \right)$$

Si $\partial R = \bigcup_{i=0}^{n-1} [z_i, z_{i+1}]$, donde $z_0 = s_1$ y $z_n = s_2$, y para todo $i = 0, \dots, n-1$ la variación de argumento es $|\Delta_{[z_i, z_{i+1}]} \arg f(s)| < \pi$, entonces:

$$N(R) = \frac{1}{2\pi} \sum_{i=0}^{n-1} \Delta_{[z_i, z_{i+1}]} \arg f(s) = \frac{1}{2\pi} \sum_{i=0}^{n-1} \arg \left(\frac{f(z_{i+1})}{f(z_i)} \right)$$

El criterio en [13] (Teorema 2.1) nos permite identificar los puntos z_i que dividen la curva ∂R y así determinar la variación de argumento de $f(s)$ en ∂R .

Teorema 2.47 (Ying et Katz). *Sea $P(s)$ una función compleja lineal, $f(s)$ una función holomorfa sobre el intervalo $[s_1, s_2]$ y $R(s)$ una función definida como $R(s) = f(s) - P(s)$. Si*

$$\min_{s \in [s_1, s_2]} |P(s)| > \max_{s \in [s_1, s_2]} R(s) \quad (2.12)$$

entonces

$$|\Delta_{[s_1, s_2]} \arg f(s)| < \pi \wedge \Delta_{[s_1, s_2]} \arg f(s) = \arg \left(\frac{f(s_2)}{f(s_1)} \right) \quad (2.13)$$

Más precisamente, si $s_0 \in [s_1, s_2]$ es tal que $P(s_0) = \min_{s \in [s_1, s_2]} |P(s)| > 0$, entonces para todo $s \in [s_1, s_2]$:

$$\langle f(s), P(s_0) \rangle > 0$$

Donde $\langle x, y \rangle = \operatorname{Re}(x\bar{y})$ es el producto escalar real en \mathbb{C} .

Lema 2.48. Sea $f(s)$ una función holomorfa sobre el intervalo $[s_1, s_2]$, $P(s)$ y $R(s)$ definidas por:

$$P(s) = f(s_1) + \frac{f(s_2) - f(s_1)}{s_2 - s_1}(s - s_1)$$

y $R(s) = f(s) - P(s)$. Si $M(s_1, s_2)$ es tal que para todo $s \in [s_1, s_2]$,

$$|f''(s)| \leq M(s_1, s_2)$$

entonces

$$\max_{s \in [s_1, s_2]} R(s) \leq M(s_1, s_2) \frac{|s_1 - s_2|^2}{8}$$

El lema anterior nos sirve para reemplazar el cálculo del máximo de la función $R(s)$ por un cálculo más simple a partir de una cota de $f''(s)$.

Por otro lado, el cálculo del mínimo de la norma de la función lineal $P(s)$ es simple y directo. Este valor se puede obtener fácilmente usando el siguiente lema:

Lema 2.49. Sean $a, b \in \mathbb{C}$ con $a \neq b$,

$$m = \min_{t \in [0, 1]} |a + t(b - a)|$$

Si t_0 denota el único elemento del intervalo donde se alcanza el mínimo, $m = |a + t_0(b - a)|$, entonces:

- Si $\operatorname{Re}(a\bar{b}) \geq |a|^2$, entonces $m = |a|$ y $t_0 = 0$.
- Si $\operatorname{Re}(a\bar{b}) \geq |b|^2$, entonces $m = |b|$ y $t_0 = 1$.

- En caso contrario, $m = \frac{Im(\bar{a}b)}{|b-a|}$ y $t_0 = \frac{|a|^2 - Re(\bar{a}b)}{|b-a|^2}$

Es evidente que podemos verificar 2.12, gracias a los lemas anteriores, usando:

$$\min_{s \in [s_1, s_2]} |P(s)| > M(s_1, s_2) \frac{|s_1 - s_2|^2}{8} \quad (2.14)$$

Con estas condiciones, nuestro criterio de discretización del contorno consiste en encontrar segmentos $[s'_1, s'_2]$ tales que cumplan con (2.14).

La siguiente proposición, de interés puramente teórico, nos asegura la existencia de una discretización conveniente de un intervalo fijo del plano complejo que no contiene ceros de $f(s)$, la cual nos permite calcular la variación del argumento de la función en el intervalo.

Proposición 2.50 (Convergencia). *Dada una función $f(s)$ holomorfa y sin cero en el intervalo $[s_1, s_2]$ y $M(s_1, s_2) > 0$ tal que para todo $s \in [s_1, s_2]$, $M(s_1, s_2) \geq |f''(s)|$, existe $\delta > 0$ tal que, para $[s'_1, s'_2] \subset [s_1, s_2]$ se cumple que $|s'_1 - s'_2| < \delta$,*

$$P(s) = f(s'_1) + \frac{f(s'_2) - f(s'_1)}{s'_2 - s'_1}(s - s'_1)$$

entonces

$$\min_{s \in [s'_1, s'_2]} |P(s)| > M(s_1, s_2) \frac{|s'_1 - s'_2|^2}{8} \geq \max_{s \in [s'_1, s'_2]} R(s)$$

donde $R(s) = f(s) - P(s)$, y por lo tanto se cumplen las condiciones de 2.47.

Demostración. En primer lugar,

$$m = \min_{s \in [s_1, s_2]} |f(s)| > 0$$

Por continuidad uniforme, existe $\delta' > 0$ tal que, para $x, y \in [s_1, s_2]$, $|x - y| < \delta'$, se tiene que $|f(x) - f(y)| < m/2$. Si $[s'_1, s'_2] \subset [s_1, s_2]$ con $|s'_1 - s'_2| < \delta = \min\{\delta', 2\sqrt{m/M(s_1, s_2)}\}$, entonces $|f(s'_1) - f(s'_2)| < m/2$, de donde

$$|P(s)| \geq |f(s'_1)| - |f(s'_1) - f(s'_2)| > \frac{m}{2}$$

por la definición de δ tenemos también que

$$|s'_1 - s'_2| < 2\sqrt{\frac{m}{M(s_1, s_2)}}, \text{ de donde } M(s_1, s_2) \frac{|s'_1 - s'_2|^2}{4} < m$$

y

$$\min_{s \in [s'_1, s'_2]} |P(s)| > \frac{m}{2} > M(s_1, s_2) \frac{|s'_1 - s'_2|^2}{8} \geq \max_{s \in [s'_1, s'_2]} R(s).$$

Nótese que la última desigualdad de la cadena se cumple debido a que $M(s_1, s_2) \geq M(s'_1, s'_2)$ por la misma razón de que $[s'_1, s'_2] \subset [s_1, s_2]$. \square

2.8. Método de bisección en el plano complejo

El método de bisección en una dimensión se usa para hallar una raíz de una función continua en un intervalo $[a, b]$ tal que $f(a) \cdot f(b) < 0$. En términos generales, cada iteración consiste en dividir el intervalo actual a la mitad y elegir el subintervalo que contenga la raíz (por ello se debe garantizar que exista solamente una). Se realizan las iteraciones hasta que se obtenga una aproximación de la raíz correcta dentro de una tolerancia de error deseada.

Existen muchas formas de variación de este método, también llamado *búsqueda binaria*, debido a que se puede usar la propiedad siguiente:

Sean $[a_i, b_i]$ los extremos del intervalo luego de la i -ésima iteración, entonces (si $f(x) = 0$):

$$x \in \bigcap_{i=1}^{\infty} [a_i, b_i]$$

Basándose en el método de bisección unidimensional, se propone un método de bisección bidimensional en el plano complejo (rectángulo) para obtener las ubicaciones de los ceros a determinar.

Cada iteración de este método se define de la siguiente forma, dependiente de la cantidad de ceros dentro del rectángulo a analizar:

- Hay 0 ceros: El rectángulo debe ser descartado.
- Hay 1 cero: El rectángulo es considerado para hallar la respuesta. Si no se halla la respuesta, proceder como si hubiera más de 1 cero en el rectángulo.
- Hay $q > 1$ ceros: El rectángulo debe ser reducido a 2 rectángulos de menor tamaño, esto se logrará realizando un corte sobre la dimensión más larga del mismo.

Nótese que la bisección se realiza siempre sobre la dimensión más larga para poder obtener rectángulos con dimensiones no tan lejanas entre sí; esto beneficia a la ejecución del algoritmo pues llega a uniformizar las iteraciones a partir de cierto punto. Cabe recordar que las dimensiones de los rectángulos originales son un intervalo muy pequeño para la parte real y un intervalo muy grande para la parte imaginaria. Por último, es evidente que los rectángulos generados mantendrán la cantidad de ceros original y llegarán a reducirse lo suficiente como para ser descartados u obtener uno de los ceros.

2.9. Algoritmo base

El algoritmo base consiste en mantener curvas poligonales en forma de rectángulo (por simplicidad), contabilizar los ceros usando el principio del argumento y luego

tomar la decisión correspondiente de acuerdo al método de bisección planteado.

Algoritmo 1: Algoritmo secuencial

Datos: n y T

Resultado: Ceros de $\zeta_n(s)$ en el rectángulo $[1 - n, 1,74] \times [0, T]$

$S \leftarrow \emptyset;$

$\text{Ans} \leftarrow \emptyset;$

$S.\text{push}(\text{rectangulo}(1 - n, 0, 1,74, T));$

mientras $S \neq \emptyset$ **hacer**

$u \leftarrow S.\text{pop}();$

si $\text{ceros}(u) = 0$ **entonces**

continue;

$\text{found} \leftarrow \text{false};$

si $\text{ceros}(u) = 1$ **entonces**

si $\text{newtonSucceded}(u)$ **entonces**

$\text{Ans}.\text{push}(u);$

$\text{found} \leftarrow \text{true};$

si *not found* **entonces**

$L, R \leftarrow \text{bisect}(u);$

$S.\text{push}(L);$

$S.\text{push}(R);$

Devolver $\text{Ans};$

La variación para volver el algoritmo paralelo es mantener las funciones de procesamiento de rectángulo y extracción de la pila por separado, con la idea de usar hilos que accedan paralelamente a la pila para extraer los datos a ser procesados, así como que los mismos puedan acceder a una pila extra para almacenar las respuestas que vayan encontrando.

Capítulo 3

Análisis implementativo

En este capítulo describimos el método de cálculo de raíces complejas. Este método se basa en una combinación entre un método de bisección bidimensional para aislar los ceros, combinado con un método de Newton para acelerar el cálculo de ceros una vez aislados. A su vez, la implementación presentada se da en paralelo, aprovechando la estructura particular del problema.

3.1. Lenguaje y librerías

Los resultados presentados en este informe han sido generados usando un código escrito en Python 3, mientras que los gráficos de los ceros obtenidos han sido generados usando gnuplot en formato eps.

Para poder mantener una buena precisión en los valores procesados, se decidió usar la librería *Mpmath* [5], la cual permite realizar cálculos con números reales y complejos manteniendo una precisión de hasta 1000 dígitos significativos. Como referencia, la librería de matemática simbólica *SymPy* usa como base a *Mpmath* para realizar cálculos numéricos.

Para el cálculo de los ceros, se usó una precisión de 100 dígitos significativos en el entorno de *Mpmath*.

3.2. Funciones básicas para variación de argumento

En esta sección se explicarán los fragmentos de código que corresponden a los cálculos relacionados con la variación de argumento, basándose en los conceptos explicados anteriormente en el análisis teórico.

3.2.1. `getMin(a, b)`

La naturaleza de esta función es autoexplicativa en base al fundamento teórico. Se definió la función `norm2(a)`, que devuelve el cuadrado del módulo del complejo a .

```
def norm2(a):
    return a.real**2 + a.imag**2

def getMin(a, b):
    prod = a * b.conjugate()
    if prod.real >= norm2(a): return abs(a)
    if prod.real >= norm2(b): return abs(b)
    return abs(prod.imag / abs(b-a))
```

3.2.2. `M(s1, s2, n)`

La cota superior propuesta en el análisis teórico es un bucle simple, de manera similar al evaluar la función de suma parcial f y su derivada df .

```
def M(s1, s2, n):
    ans = 0
    for i in range(2,n+1):
        ans += exp((-min(s1.real,s2.real)) * log(i)) * log(i) * log(i)
    return ans
```

3.2.3. isGreater(s1, s2, localM, n)

Esta función realiza una comparación en los valores $\min_{s \in L(s1, s2)} f(s)$ y $localM \cdot \frac{|s1-s2|^2}{8}$ para verificar que la variación de argumento actual sea estrictamente menor a π . Se usó un $\epsilon = 10^{-20}$ para poder permitir un ligero error en el cálculo.

```
def isGreater(s1, s2, localM, n):
    L = getMin(f(s1, n), f(s2, n))
    R = localM * norm2(s1 - s2) / 8
    EPS = mpf('10') ** (-20)
    return L > R + EPS
```

3.2.4. varArg(s1, s2, n)

Para poder calcular la variación de argumento de forma confiable, se planteó una partición de $L(s1, s2)$ de forma que cada elemento tenga una variación local menor que π .

Se decidió usar un método similar al de bisección, el cual obtiene un desplazamiento máximo posible t a ser tomado en cuenta para la suma de variaciones total. En la siguiente iteración se usará $2t$ como uno de los límites superiores de desplazamiento, debido a que en el peor de los casos solo aumentará en 1 la cantidad de iteraciones necesarias para el nuevo elemento de la partición (el nuevo $t' < t$), mientras que en el mejor, disminuye en 1 la cantidad de elementos en la partición (el nuevo $t' = 2t$). Es importante recordar que t no puede excederse nunca de la distancia entre el punto actual y el extremo opuesto de la recta.

Optimización: El valor de la cota superior $M(s1, s2, n)$ cumple para toda la recta $L(s1, s2)$, por lo que cumplirá para todo subsegmento de la misma: Basta con procesar una sola vez la cota (en la variable *localM*) y usarla para todos los elementos de la partición.

Observación: Si es que en algún momento el valor de t llega a ser muy pequeño, se realiza un ligero desplazamiento de la recta, pues esto implicaría que existe un cero

en la vecindad de alguno de los puntos de la recta. Esto es contemplado en el primer condicional que verifica si $t \leq EPS2 = 10^{-80}$.

```
def varArg(s1, s2, n):
    localM = M(s1,s2,n)
    t0 = mpf(0)
    t = mpf(1)
    L = s1
    d = s2 - s1
    ans = mpf(0)
    EPS1 = mpf('10') ** (-25)
    EPS2 = mpf('10') ** (-80)
    EPS3 = mpf('10') ** (-40)
    while t0 + EPS1 < 1:
        R = L + t * d
        while (not isGreater(L,R,localM,n)) and t > EPS2:
            t /= 2
            R = L + t * d
        if t <= EPS2:
            if s1.real == s2.real:
                d = EPS3 if s1.imag < s2.imag else -EPS3
                return varArg(s1 + mpc(d,0), s2 + mpc(d,0), n)
            else:
                d = EPS3 if s1.real > s2.real else -EPS3
                return varArg(s1 + mpc(0,d), s2 + mpc(0,d), n)
        ans += arg(f(R, n) / f(L, n))
        t0 += t
        L += d * t
        t = min(2 * t, 1 + EPS1 - t0)
```

```
return ans
```

3.3. Funciones básicas para procesamiento de respuestas

3.3.1. newton(LD, RU, n)

Esta función intenta calcular el único cero que está presente en el rectángulo del plano complejo con extremo inferior izquierdo LD y extremo superior derecho RU usando el Método de Newton. Realiza 10000 iteraciones como máximo, luego de las cuales verifica que la respuesta actual esté en una vecindad con radio $tol = 10^{-30}$ de la respuesta correcta (también debe estar dentro del rectángulo) y la devuelve; en caso contrario, devuelve un valor centinela (3, pues no hay ceros con parte real igual a 3) que significa que el rectángulo procesado debe ser biseccionado.

```
def newton(LD, RU, n):
    it = 10000
    x0 = (LD + RU) * mpf('0.5')
    xi = x0 + mpc(1,0)
    tol = mpf('10') ** (-30)
    while abs(x0 - xi) > tol and enrect(LD, RU, x0) and it > 0:
        xi = x0
        x0 = xi - f(xi, n) / df(xi, n)
        it -= 1
    if abs(x0 - xi) <= tol and enrect(LD, RU, x0):
        return x0
    return mpc(3,0)
```

3.4. Pila con exclusión mutua para almacenar los rectángulos

Dado que el paradigma de paralelismo usado es el multihilo, se planteó definir una clase con las siguientes características:

- Su método de almacenamiento se comporte como una pila (estructura LIFO).
- Su método de acceso sea excluyente, es decir, no se pueda dar el evento en el que dos hilos diferentes accedan a los datos del objeto simultáneamente.

3.4.1. Constructor

El constructor de la clase recibe 3 parámetros:

- *LD*: Límite inferior izquierdo en el plano complejo del rectángulo inicial a procesar.
- *RU*: Límite superior derecho en el plano complejo del rectángulo inicial a procesar.
- *n*: Límite de iteración de la suma parcial a procesar.

Esto con la finalidad de inicializar la información almacenada en la pila:

- *n*: Límite de iteración de la suma parcial a procesar.
- *stack*: Lista que almacenará los valores de cada rectángulo (6 valores: 2 extremos + 4 variaciones de argumento por lado).
- *lock*: Variable del tipo *threading.Lock()* que permite la exclusión mutua de acceso.

3.4.2. Funciones pop y push

La función *pop* es equivalente a quitar el elemento en el tope de la pila y devolverlo. En este caso, se devuelven los últimos 6 elementos agregados a la pila usando exclusión mutua. En caso que no hayan elementos, se devuelve una lista vacía.

La función *push* es equivalente a agregar elementos al tope de la pila. En este caso, se agregan todos los elementos que estén en la lista *v* que es brindada como parámetro usando exclusión mutua.

La función *size* es una herramienta que ayuda a verificar la cantidad de rectángulos almacenados en la pila, con el fin de procesar rectángulos hasta que hayan al menos *threads* de ellos en la pila simultáneamente, donde *threads* es la cantidad de hilos usados.

```
class MutexStack:
    def __init__(self, LD, RU, _n):
        self.stack = []
        self.n = _n
        self._lock = threading.Lock()
        LU = mpc(LD.real, RU.imag)
        RD = mpc(RU.real, LD.imag)
        self.stack = [LD, RU, varArg(LD, RD, _n), varArg(RD, RU, _n)
            , varArg(RU, LU, _n), varArg(LU, LD, _n)]

    def pop(self):
        n = self.n
        with self._lock:
            if self.stack:
                res = self.stack[-6:]
                for _ in range(6): self.stack.pop()
                return res
```

```
        else:
            return []

    def push(self, v):
        with self._lock:
            for x in v:
                self.stack.append(x)

    def size(self):
        return len(self.stack)
```

3.4.3. Pila con exclusión mutua para almacenar los ceros

Esta pila es una versión ligeramente diferente de aquella para almacenar los rectángulos, las funciones son autoexplicativas e iguales a la anteriormente referida.

```
class MutexAnswer:
    def __init__(self, threads):
        self.v = []
        self.len = 0
        self._lock = threading.Lock()

    def push(self, x):
        with self._lock:
            self.v.append(x)
            self.len += 1

    def pop(self):
        with self._lock:
            if self.v:
```



```
        res = self.v.pop();
        self.len -= 1
        return res
    else:
        return mpc('3','0')

def size(self):
    return self.len
```

3.5. Función de inicialización de la pila

Para evitar que alguno de los hilos asuma que todo el procesamiento ya terminó (la pila está vacía) como falso positivo, esta función se asegura de que la pila tenga la cantidad de rectángulos suficiente como para que cada hilo procese al menos 1 en su primer acceso a la pila.

```
def initialize(S, R, n, nodes):
    while S.size() < 6 * nodes:
        cur = S.pop()
        val = compute(cur, n)
        if val[0] == 1:
            R.push(val[1])
        elif val[0] == 2:
            S.push(val[1:])
```

3.6. Funciones básicas para el procesamiento en los hilos

3.6.1. `compute(a, n)`

Esta función es la que se encarga de procesar un rectángulo según corresponda, usando el primer elemento de la lista devuelta como código (0 es descarte, 1 es respuesta, 2 es bisección):

- La cantidad de ceros es 0: Descartar el rectángulo, devolviendo una lista con el primer elemento igual a 0.
- La cantidad de ceros es 1: Intentar hallar el único cero del rectángulo mediante método de Newton. Si se llega a obtener el cero con éxito, devolverlo en una lista con el primer elemento igual a 1 y el segundo elemento igual al cero. Si no se obtiene la respuesta deseada, biseccionar.
- La cantidad de ceros es mayor que 1: Biseccionar el rectángulo mediante la dimensión más larga y reciclar las variaciones de argumento lo más posible según corresponda. Luego obtener los dos rectángulos generados y devolverlos en una lista con 13 elementos: el primer elemento igual a 2, los siguientes 6 correspondientes al primer rectángulo y los últimos 6 correspondientes al segundo rectángulo.

```
def compute(a, n):
```

```
    LD = a[0]
```

```
    RU = a[1]
```

```
    V0 = a[2]
```

```
    V1 = a[3]
```

```
    V2 = a[4]
```

```
    V3 = a[5]
```

```

zeros = round(sum(a[2:]) / (2 * acos(mpf('-1'))))
if zeros == 0: return [0, mpc(3, 0)]
if zeros == 1:
    O = (LD + RU) / 2
    res = newton(LD, RU, n)
    if enrect(LD, RU, res):
        return [1, res]
if abs(LD.real - RU.real) > abs(LD.imag - RU.imag):
    M1 = mpc((LD.real + RU.real) / 2, LD.imag)
    M2 = mpc((LD.real + RU.real) / 2, RU.imag)
    FD = varArg(LD,M1,n)
    FU = varArg(RU,M2,n)
    FM = varArg(M1,M2,n)
    v1 = FD + FM + V2 - FU + V3
    v2 = V0 - FD + V1 + FU - FM
    if round((v1 + v2) / (2 * acos(-1))) != zeros:
        print("Fatal Error. Sum of partitions isn't equal to total")
    return [2, LD, M2, FD, FM, V2 - FU, V3, M1, RU, V0 - FD, V1, FU, -FM]
else:
    M1 = mpc(RU.real, (LD.imag + RU.imag) / 2)
    M2 = mpc(LD.real, (LD.imag + RU.imag) / 2)
    FD = varArg(mpc(RU.real,LD.imag),M1,n)
    FU = varArg(mpc(LD.real,RU.imag),M2,n)
    FM = varArg(M1,M2,n)
    v1 = V0 + FD + FM + (V3 - FU)
    v2 = -FM + (V1 - FD) + V2 + FU
    if round((v1 + v2) / (2 * acos(-1))) != zeros:
        print("Fatal Error. Sum of partitions isn't equal to total")
    return [2, LD, M1, V0, FD, FM, V3 - FU, M2, RU, -FM, V1 - FD, V2, FU]

```

3.6.2. solve(S, R, id, n)

Esta función mantiene al hilo *id* activo para procesar datos hasta que se pueda afirmar, con alta probabilidad, que la pila está vacía porque ya se hallaron todos los cerros.

La variable *failed* almacena la cantidad de accesos “fallidos” (es decir, que la pila no devolvió un rectángulo); de forma que, si hay 12 intentos sin nada que procesar, el hilo deje de trabajar.

Luego de procesar el valor actual (en el caso que el acceso haya sido un éxito), agrega los datos a alguna de las dos pilas (si es necesario) en base al código explicado en la función anterior.

```
def solve(S, R, id, n):
    failed = 0
    while failed < 12:
        cur = S.pop();
        if len(cur) == 6:
            val = compute(cur, n)
            if val[0] == 1:
                R.push(val[1])
            elif val[0] == 2:
                S.push(val[1:])
            failed = 0
        else:
            failed += 1
```

3.7. Script para generación de gráficos

Se usó *gnuplot* para generar las imágenes en base a la información obtenida por el algoritmo en cada suma parcial. El script se puede generalizar (funcionalmente, debido

a que la sintaxis del software no permite la estructura mostrada):

```
#!/usr/bin/env gnuplot
set terminal postscript eps color
set output "zeta(n).eps"
set xrange [L:R]
set yrange [0:T]
set xlabel 'Re(s)'
set ylabel 'Im(s)'
plot "toplotzeta(n)" with dots lw 4 title ""
```

Los valores L y R fueron extraídos usando un código en C++ para comparar los resultados de los ceros manipulando los números como cadenas de caracteres y usando como base a la comparación lexicográfica nativa.

Capítulo 4

Resultados

4.1. Tabla resumen de los resultados

Definimos que los ceros z_i tales que $0 < \text{Im}(z_i) < T$ con $T > 0$ de la función $\zeta_n(s)$ están en la franja:

$$\varphi_n(T) \leq \text{Re}(z_i) \leq \theta_n(T)$$

Un resultado obtenido por (Borwein et al., 2007) indica que si $\text{Re}(s) < 1 - n$ entonces $\zeta_n(s) \neq 0$, así como el obtenido por (Spira, 1966) que restringe superiormente la posibilidad de obtener un cero considerando que si $\text{Re}(s) > 1$ entonces $\zeta_n(s) \neq 0$.

Se usó como referencia el intervalo real:

$$L = 1 - n \leq \varphi_n(T) \leq \text{Re}(z_i) \leq \theta_n(T) \leq 1,74 = R$$

Para obtener los ceros en cada suma parcial procesada, de manera que en la tabla presentada se cumple

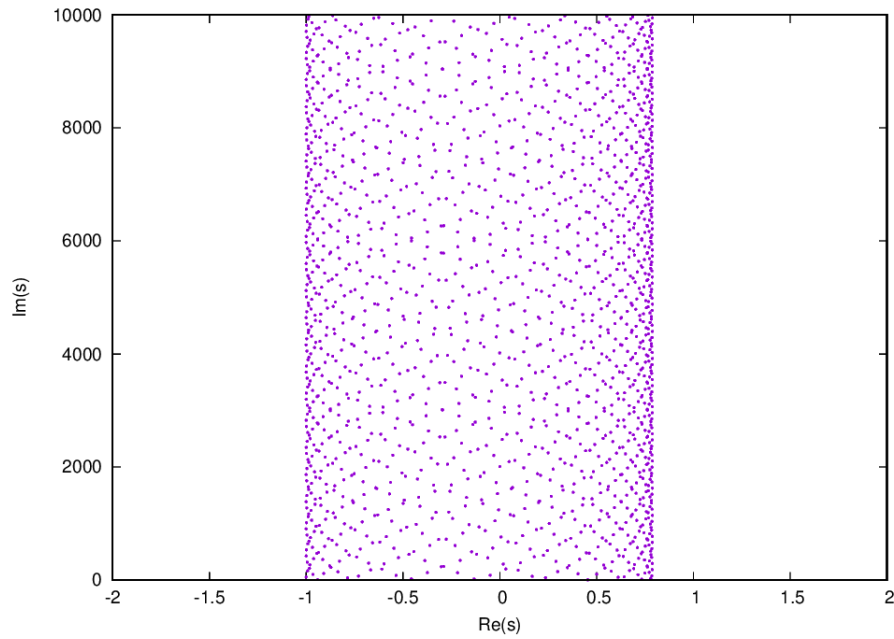
$$\varphi_n(T) = \inf\{\text{Re}(s) : s \in [1 - n, 1,74] \times [0, T] \text{ y } \zeta_n(s) = 0\}$$

$$\theta_n(T) = \sup\{\text{Re}(s) : s \in [1 - n, 1,74] \times [0, T] \text{ y } \zeta_n(s) = 0\}$$

n	T	$N(n, T)$	$\frac{T}{2\pi} \log n$	$\varphi_n(T)$	$\theta_n(T)$
3	10000	1748	1748.50	-0.999996283002667	0.787883123800742
4	10000	2206	2206.36	-1.214276156461922	0.626286448302958
5	10000	2561	2561.50	-2.423596440213824	0.890682184115720
6	10000	2851	2851.67	-2.885468694620030	0.840975970572894
7	10000	3096	3097.01	-3.765472207336119	0.972526738368160
8	10000	3309	3309.53	-4.347145915134515	0.917947429667945
9	10000	3496	3496.99	-5.005833617526464	0.887011426922500

Cuadro 4.1: Recopilación de los ceros obtenidos

4.2. Gráficas de los ceros encontrados

Figura 4.1: 1748 ceros de $\zeta_3(s)$ en $-2 < Re < 1,74$ y $0 < Im < 10^4$

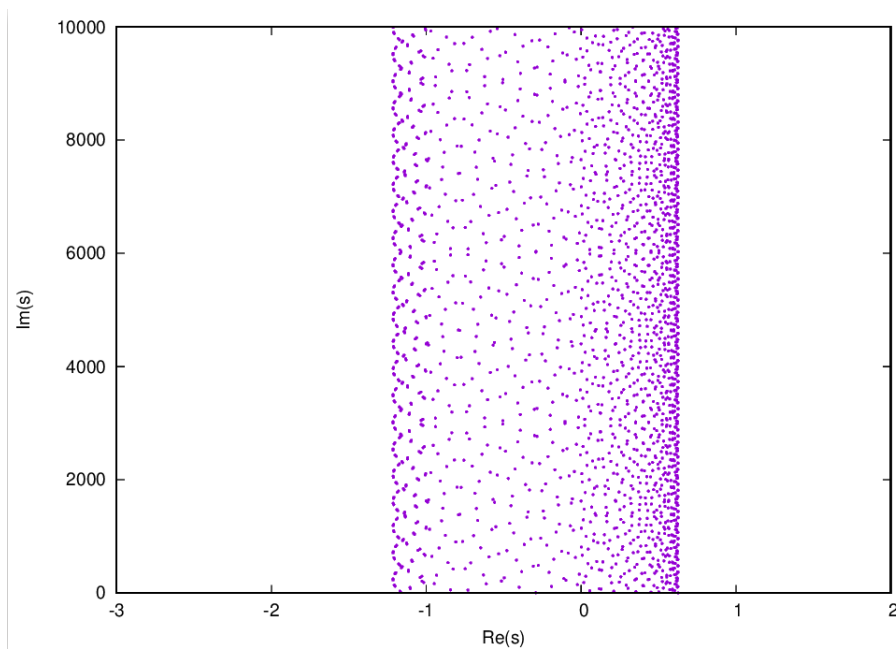


Figura 4.2: 2206 ceros de $\zeta_4(s)$ en $-3 < \text{Re} < 1,74$ y $0 < \text{Im} < 10^4$

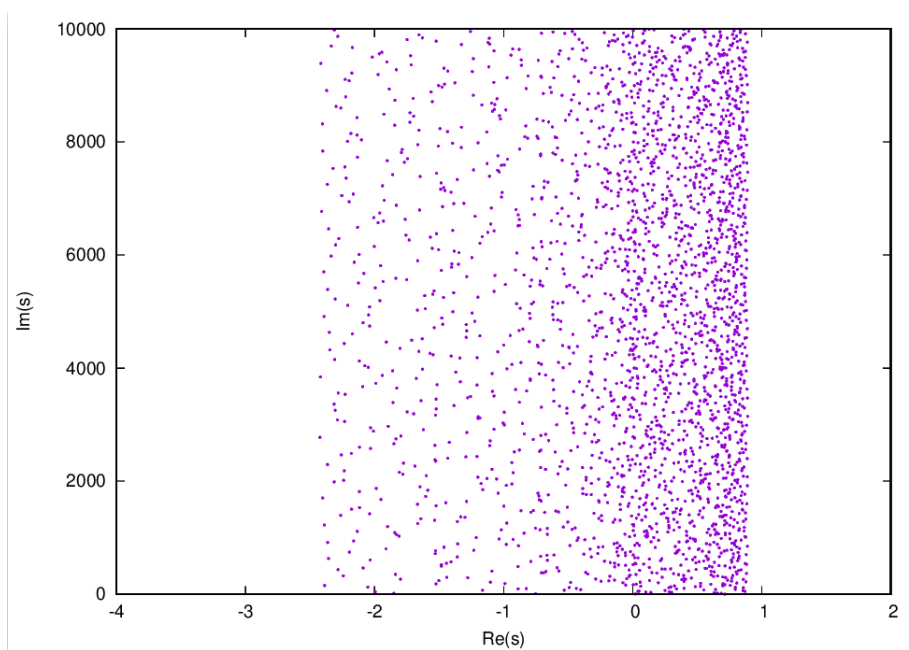


Figura 4.3: 2561 ceros de $\zeta_5(s)$ en $-4 < \text{Re} < 1,74$ y $0 < \text{Im} < 10^4$

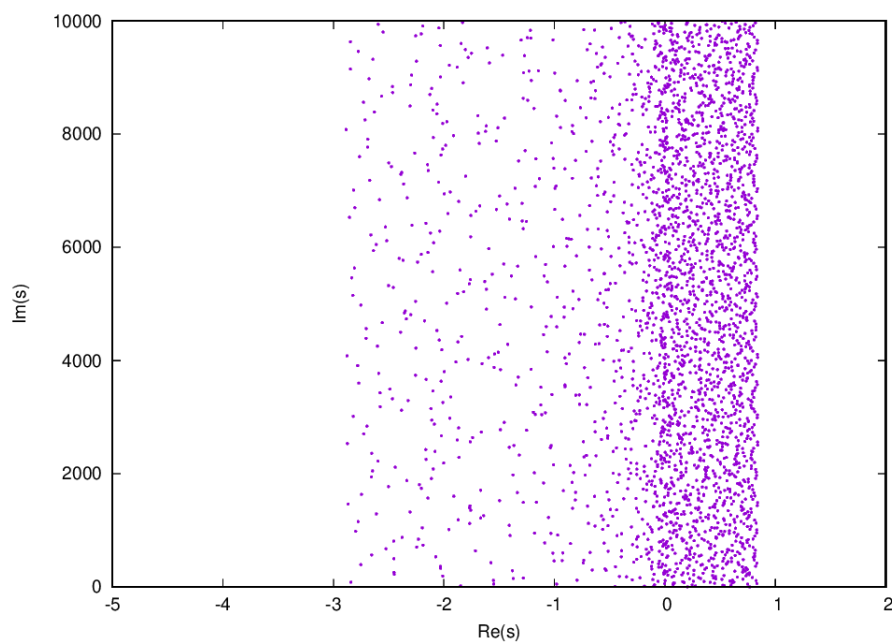


Figura 4.4: 2851 ceros de $\zeta_6(s)$ en $-5 < \text{Re} < 1,74$ y $0 < \text{Im} < 10^4$

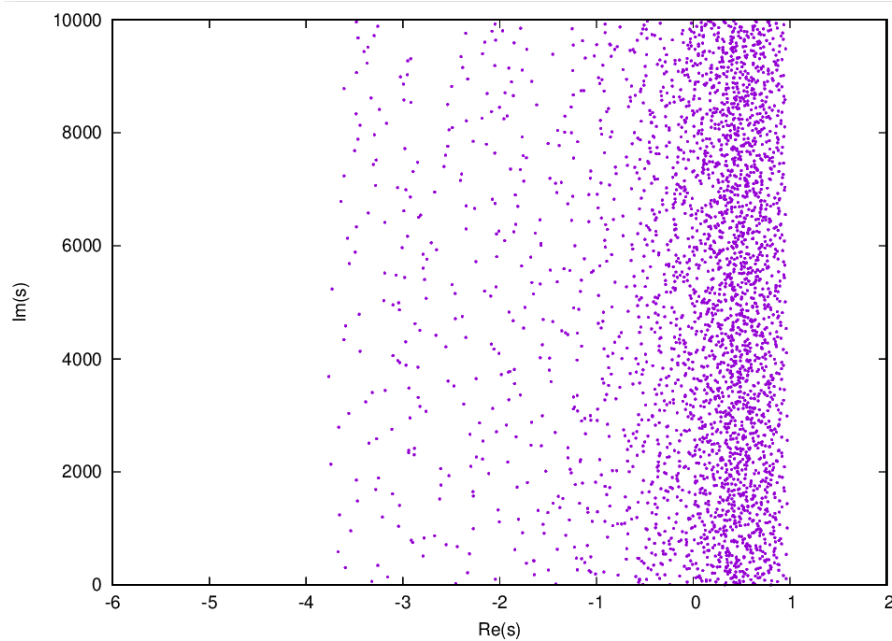


Figura 4.5: 3096 ceros de $\zeta_7(s)$ en $-6 < \text{Re} < 1,74$ y $0 < \text{Im} < 10^4$

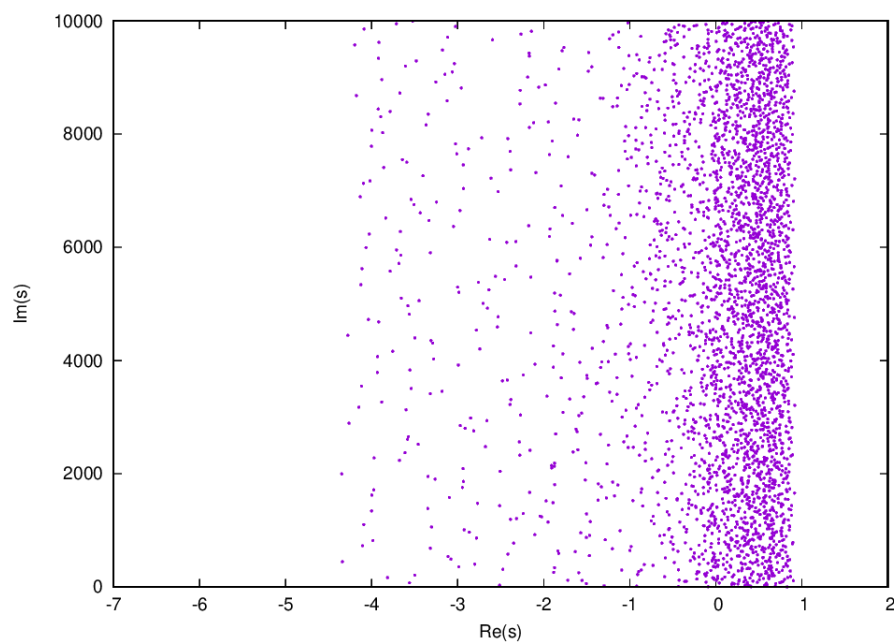


Figura 4.6: 3309 ceros de $\zeta_8(s)$ en $-7 < \text{Re} < 1,74$ y $0 < \text{Im} < 10^4$

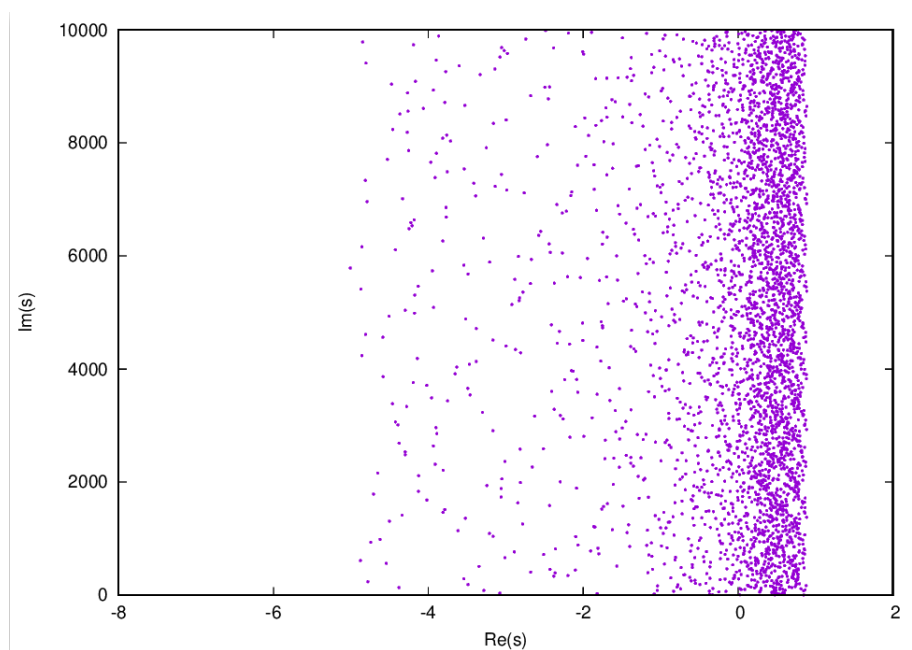


Figura 4.7: 3496 ceros de $\zeta_9(s)$ en $-8 < \text{Re} < 1,74$ y $0 < \text{Im} < 10^4$

4.3. Resultados modulares en los complejos

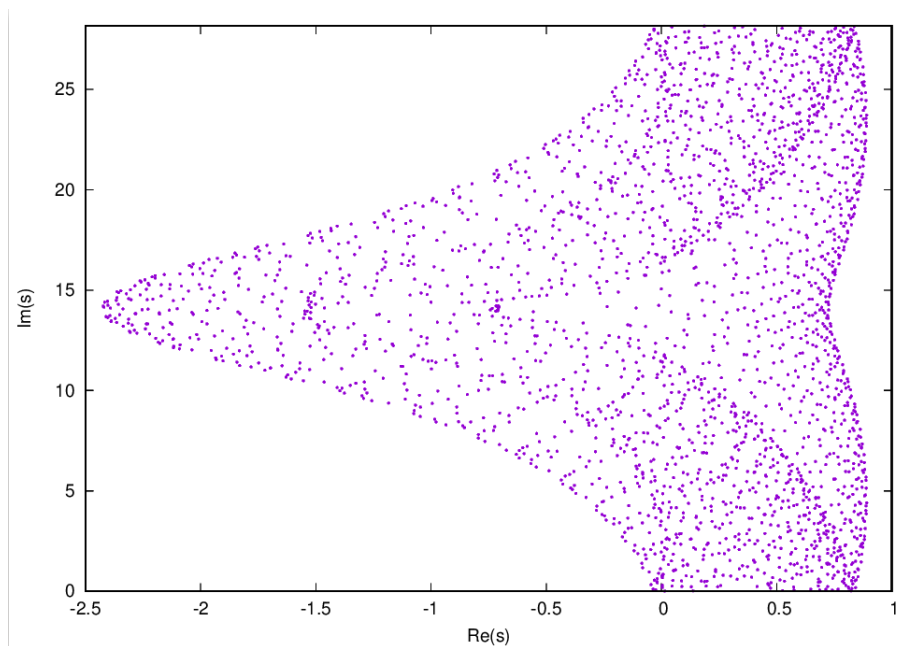


Figura 4.8: 2561 ceros de $\zeta_5(s)$ en $-4 < Re < 1,74$ y $0 < Im < 10^4$ módulo $\frac{2\pi}{\log 5/4}i$

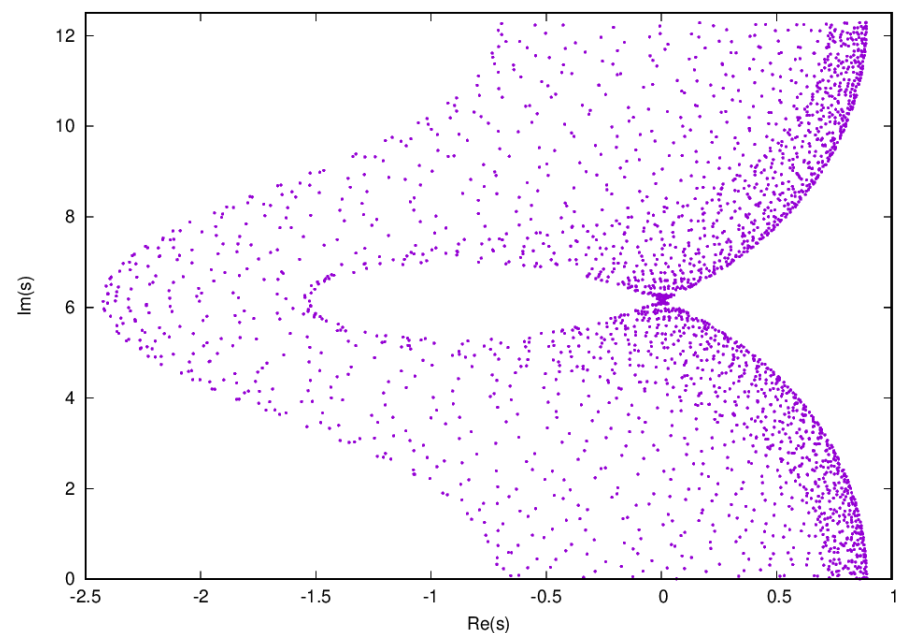


Figura 4.9: 2561 ceros de $\zeta_5(s)$ en $-4 < Re < 1,74$ y $0 < Im < 10^4$ módulo $\frac{2\pi}{\log 5/3}i$

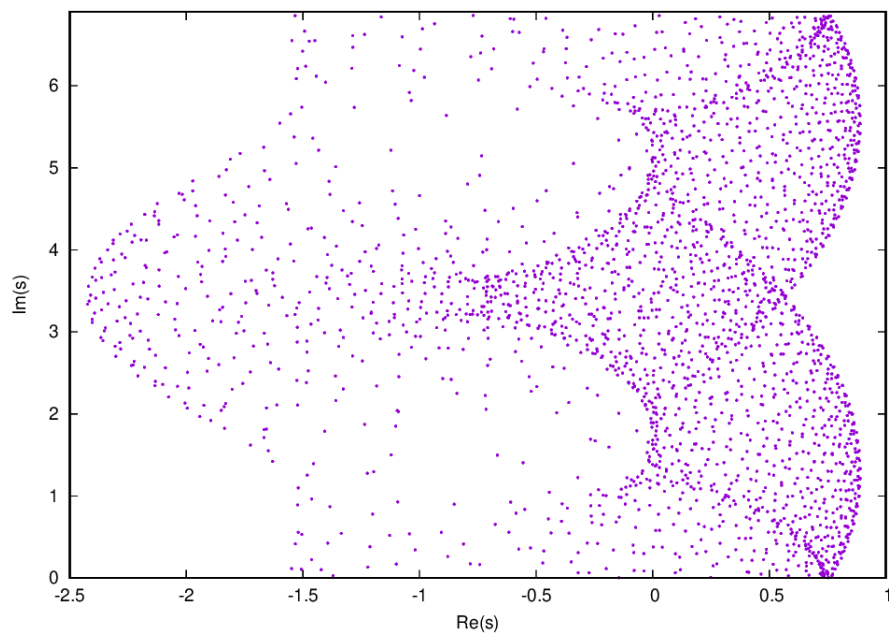


Figura 4.10: 2561 ceros de $\zeta_5(s)$ en $-4 < \text{Re} < 1,74$ y $0 < \text{Im} < 10^4$ módulo $\frac{2\pi}{\log 5/2}i$

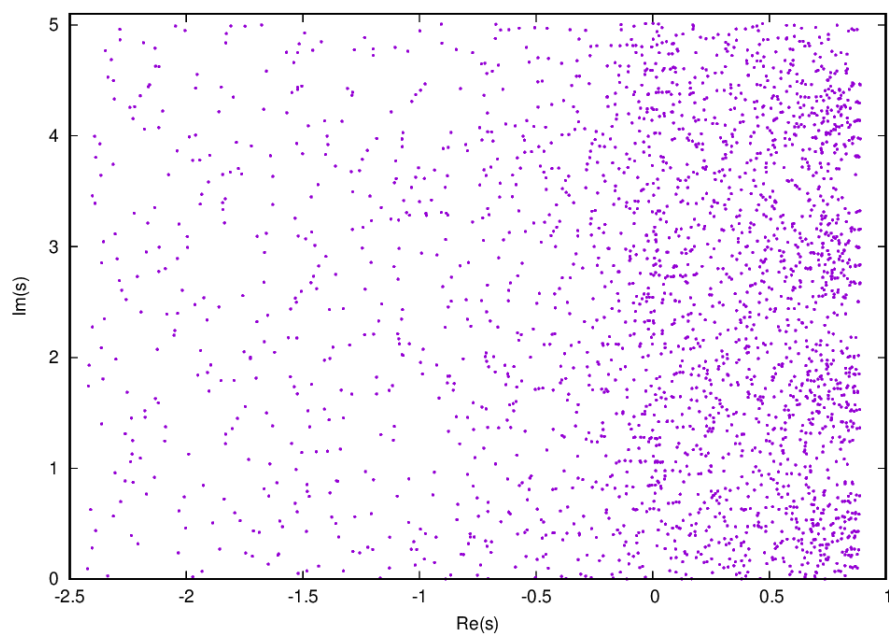


Figura 4.11: 2561 ceros de $\zeta_5(s)$ en $-4 < \text{Re} < 1,74$ y $0 < \text{Im} < 10^4$ módulo $\frac{2\pi}{\log 7/2}i$

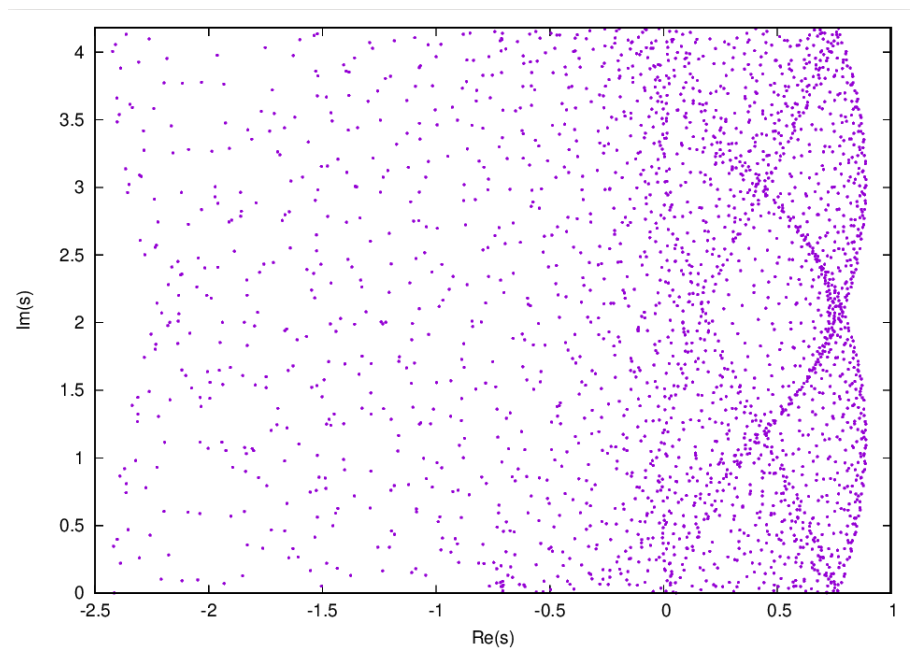


Figura 4.12: 2561 ceros de $\zeta_5(s)$ en $-4 < Re < 1,74$ y $0 < Im < 10^4$ módulo $\frac{2\pi}{\log 9/2}i$

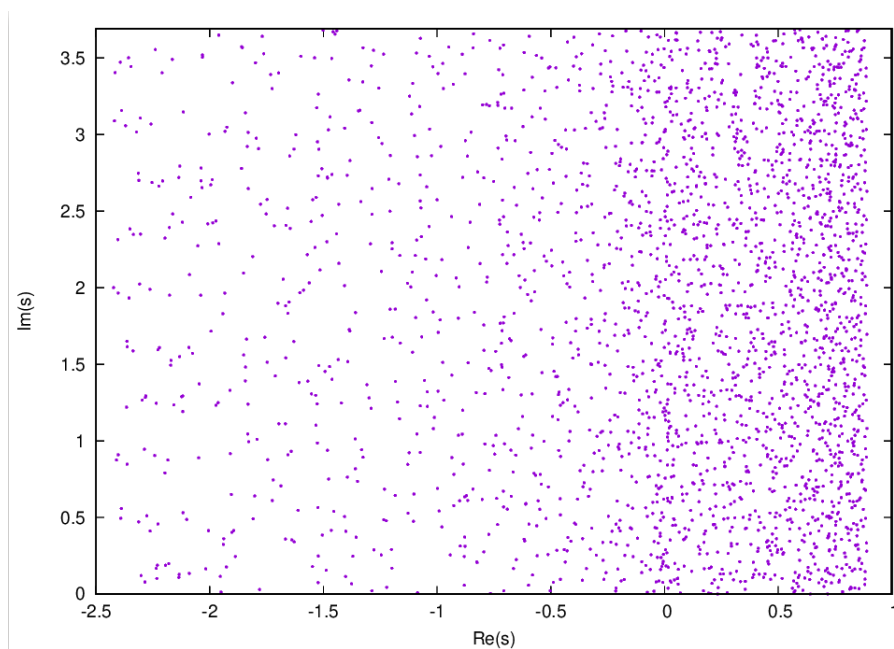


Figura 4.13: 2561 ceros de $\zeta_5(s)$ en $-4 < Re < 1,74$ y $0 < Im < 10^4$ módulo $\frac{2\pi}{\log 11/2}i$

Capítulo 5

Conclusiones

5.1. Conclusión inmediata

Se pudo paralelizar de manera sencilla el algoritmo propuesto para cálculo de ceros de las sumas parciales de la función zeta de Riemann usando el paradigma de multihilos, así como una librería de alta precisión aritmética, en el lenguaje Python 3.

Bibliografía

- [1] M. Balazard and O. V. Castanón. Sur l'infimum des parties réelles des zéros des sommes partielles de la fonction zêta de riemann. *Comptes Rendus Mathématique*, 347(7-8):343–346, 2009.
- [2] E. Bombieri. Problems of the millennium: The riemann hypothesis, 2000.
- [3] P. Borwein, G. Fee, R. Ferguson, and A. van der Waall. Zeros of partial sums of the riemann zeta function. *Experimental Mathematics*, 16(1):21–39, 2007.
- [4] E. Dubon. Sobre los ceros de polinomios de dirichlet, en general, y los de las sumas parciales de la función zeta de riemann, en particular. *Proyecto de investigación.*, 2015.
- [5] F. Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.18)*, December 2013. <http://mpmath.org/>.
- [6] H. L. Montgomery. Zeros of approximations to the zeta function. In *Studies in pure mathematics*, pages 497–506. Springer, 1983.
- [7] G. Mora and J. M. Sepulcre. Computing the zeros of the partial sums of the riemann zeta function. *Annali di Matematica Pura ed Applicata (1923-)*, 194(5):1499–1504, 2015.
- [8] D. J. Platt and T. S. Trudgian. Zeroes of partial sums of the zeta-function. *LMS Journal of Computation and Mathematics*, 19(1):37–41, 2016.

- [9] B. Riemann. Consecuencias de la hipótesis de riemann. *Revista de educación matemática*, 30(3), 2015.
- [10] R. Spira. Zeros of sections of the zeta function. i. *Mathematics of Computation*, 20(96):542–550, 1966.
- [11] R. Spira. Zeros of sections of the zeta function. ii. *Mathematics of Computation*, 22(101):163–173, 1968.
- [12] P. Turán. *On some approximative Dirichlet-polynomials in the theory of the zeta-function of Riemann*, volume 24. I kommission hos Munksgaard, 1948.
- [13] X. Ying and I. N. Katz. A reliable argument principle algorithm to find the number of zeros of an analytic function in a bounded domain. *Numerische Mathematik*, 53(1-2):143–163, 1988.