Tesis Doctoral

Métodos geométricos para aproximar raíces de polinomios, con aplicaciones a procesamiento de señal

Juan Luis García Zapata

Tecnología de los Computadores y las Comunicaciones

2015

Tesis Doctoral

Métodos geométricos para aproximar raíces de polinomios, con aplicaciones a procesamiento de señal

Juan Luis García Zapata

Tecnología de los Computadores y las Comunicaciones

2015

El director, Juan Carlos Díaz Martín, conforme:

Índice general

Capítulo 1

Introducción

Los polinomios son un tipo de funciones con una larga historia en matemáticas, ciencia e ingeniería, como puede verse en la revisión [Pan, 1997]. Una razón es que pueden ser evaluados numéricamente con un número finito de multiplicaciones y adiciones. Además, resultan ser densos en el conjunto de funciones continuas [Rudin, 1987], es decir, pueden ser usados para aproximar funciones continuas de evaluación más complicada, o ajustarse a los datos obtenidos en cualquier proceso de medición de magnitudes físicas.

Se introducen en enseñanza secundaria para ejemplificar los principios básicos del Álgebra y sus métodos. Esto hace que sea conocido que las raíces de polinomios de grado 2, 3 y 4 pueden obtenerse con ciertas fórmulas, y que no existen tales fórmulas (usando radicales) para grados superiores. En la práctica se recurre a métodos numéricos, más efectivos para aproximar raíces que las fórmulas por radicales. Los polinomios que aparecen en aplicaciones científicas y de ingeniería pueden ser de grado superior al centenar, por ejemplo en procesamiento digital de señal [Sitton et al., 2003].

Los métodos para calcular raíces de polinomios, se pueden clasificar a grandes rasgos en iterativos y geométricos. Los métodos iterativos están basados en una sucesión de estimaciones de error y corrección que, en la mayoría de los casos, conduce a un punto del plano complejo tan cerca de una raíz como se quiera. Como se detalla más adelante, estos métodos son rápidos (convergencia más que lineal) y su análisis, es decir, la prueba de su corrección como algoritmo y la determinación de los recursos necesarios, se basa en técnicas numéricas bien conocidas. Por el

contrario, los métodos geométricos se basan en la distribución de raíces en el plano complejo. Por ejemplo, acotan el módulo de las raíces, o las separan (es decir, definen regiones del plano que contienen precisamente una raíz).

Sin embargo, como se describirá en el siguiente apartado, los métodos iterativos no son fácilmente aplicables en la práctica a los polinomios que aparecen en las aplicaciones de procesamiento de señal. El objetivo de esta memoria es desarrollar, analizar y comparar un método geométrico de cálculo de raíces adecuado para polinomios de alto grado. Se basa en el índice de curvas planas cerradas (que es el número de vueltas alrededor del origen). En el siguiente capítulo se precisa un método para el cálculo del índice, que se usará en el capítulo 3 para la extracción de raíces. En el capítulo 4 llevaremos a cabo una comparativa de rendimiento con respecto a otros métodos.

En este capítulo de introducción, en un primer apartado describimos el uso de polinomios en procesamiento de señal, en particular para el estudio de formantes de la señal de voz. En un segundo apartado repasamos los métodos para hallar raíces, clasificados en iterativos y geométricos. En el tercer apartado repasamos las nociones de cálculo numérico y análisis de algoritmos necesarias para encuadrar las aportaciones de la presente memoria.

1.1. Modelos polinómicos en procesado de voz

En este apartado introducimos la técnica de predicción lineal en procesado de voz, que conduce a un modelo donde el concepto algebraico de raíz tiene un significado físico directo: el formante. Repasamos la literatura, tanto el fundamento teórico como algunas aplicaciones, aunque sea someramente. También mencionamos los proyectos que constituyen la motivación de este trabajo.

Espectrograma

Para analizar la señal de voz, cuyas características espectrales cambian rápidamente, es necesario dividirla en segmentos en los que estas sean estacionarias. La noción de señal estacionaria se corresponde con que el contenido espectral permanezca relativamente estable. En intervalos temporales pequeños (10-20 msg, 220-240 muestras de una señal tomadas con una frecuencia de 22050 kHz) el con-

tenido espectral de la señal de voz cambia poco, y puede considerarse estacionaria. A cada uno de estos intervalos casi-estacionarios (conocido como ventana de análisis o *frame*), se le pueden aplicar los procedimientos habituales de análisis espectral. Como ejemplo se han elaborado unos espectrogramas con la aplicación Praat [Boersma, 2002] a partir de la grabación SS40AANT (una pronunciación de la palabra *anterior*) de la base de datos del proyecto IVORY [Gómez, 1998] (figura 1.1).

En estos análisis tiempo-frecuencia, la precisión en una dimensión es inversamente proporcional a la precisión en la otra. En el caso de señales de voz, puede escogerse una ventana corta (banda ancha), para resaltar características temporales, o una ventana larga (banda estrecha), para resaltar las características espectrales. Puede compararse la figura 1.2 con la 1.3.

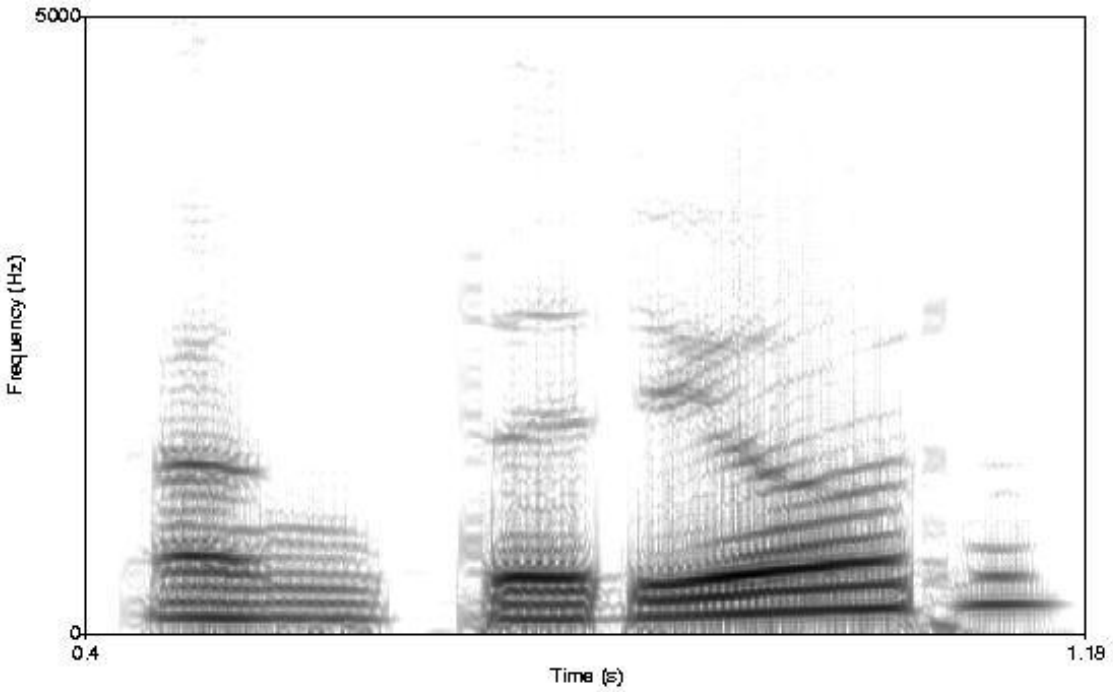


Figura 1.1: Espectrograma de *anterior*, con ventana rectangular de duración 20 ms.

Un segmento de análisis recortado de una señal de voz muestra un cambio de comportamiento muy brusco en el borde del segmento. Es conveniente suavizarlo multiplicando sus valores por una función de ventana, como por ejemplo la función

de ventana de Hamming. Por simplificar suele hablarse de ventana, obviando la palabra función. La ventana de Hamming es preferible a una ventana rectangular (esta consistiría en tomar como ventana de análisis los valores de la señal original) porque introduce menos distorsión, como se ve en la distribución de sus lóbulos laterales en frecuencia [Olive, 1971]. La elección de las características de esta ventana se tratan en [Smith, 2003]. Se discuten sus características y se proporcionan referencias en [Riley, 1989].

Esta segmentación de la señal sugiere un "procesado por bloques", en el que se somete a análisis las sucesivas ventanas obtenidas de la señal de voz. Estos bloques suelen superponerse parcialmente para dar una descripción más suave de toda la corriente de datos. Por otro lado, si el análisis al que se va a someter la voz es adaptativo (de modo que los parámetros de análisis se actualizan según van llegando nuevas muestras, en vez de realizar un análisis desde cero para cada ventana) tendríamos un "procesado muestra a muestra". En la discusión genérica de [Ouaaline and Radouane, 1998] se habla de estos dos paradigmas, los "métodos de bloque" frente a los métodos "muestra a muestra"

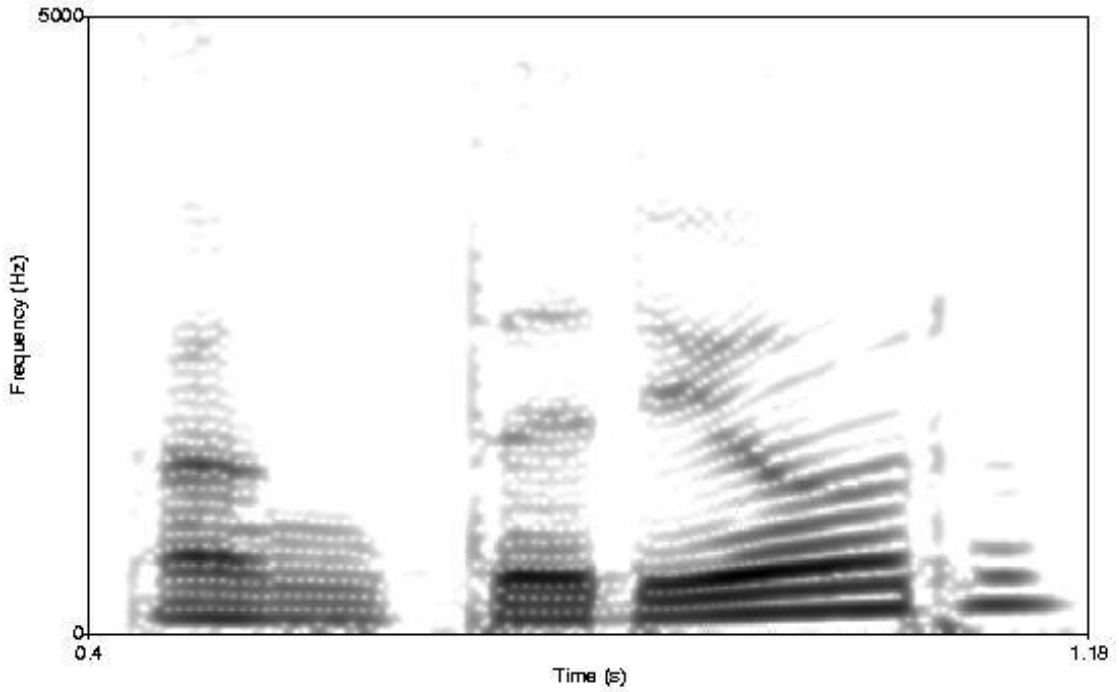


Figura 1.2: Espectrograma de *anterior*, con ventana Hamming de duración 20 ms.

Comparando con un espectrograma de la misma señal pero con una ventana de menor tamaño, se aprecia que algunos aspectos (como el rayado horizontal frente al vertical) son artefactos introducidos por la ventana de análisis. Las rayas verticales en el análisis de tiempo corto se corresponden con los golpes individuales que produce la glotis. El rayado horizontal se debe a la interacción del tren de impulsos glotales con la ventana de análisis. Sin embargo otros aspectos (los barras horizontales en 0'5 s, 1000 Hz y 2000 Hz) son independientes de la longitud de la ventana de análisis.

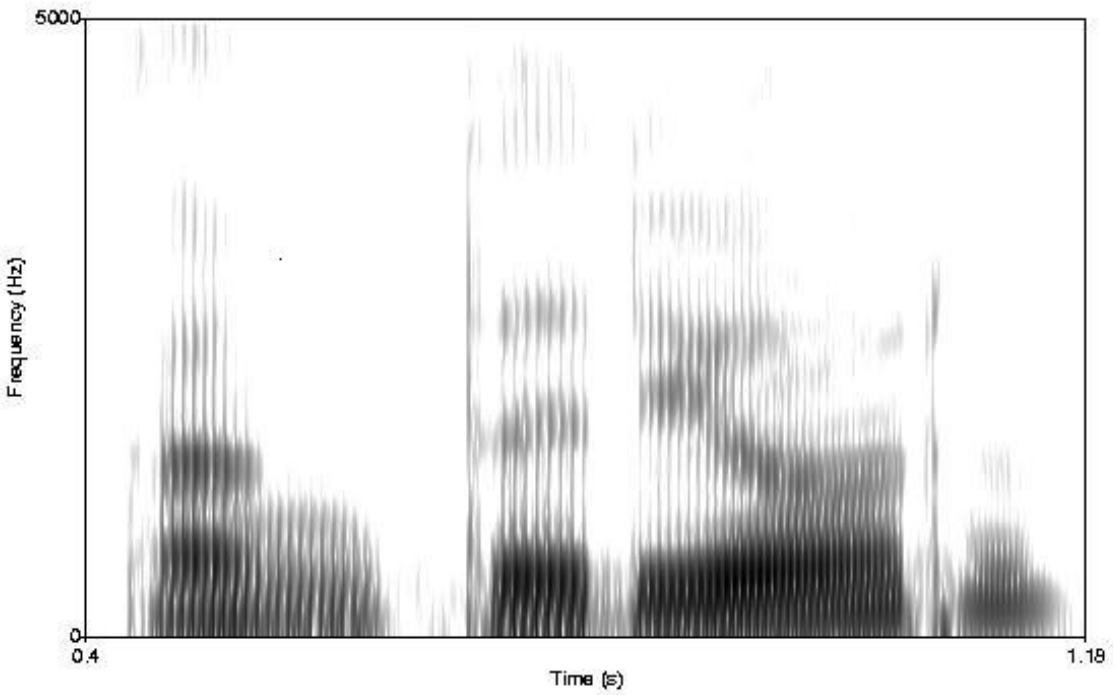


Figura 1.3: Espectrograma de *anterior*, con ventana Hamming de duración 5 ms.

Independientemente del método seguido para la obtención de características espectrales fugaces (*short-time spectrum*), la que más sobresale a primera vista es la *frecuencia fundamental* F_0, y sus armónicos. Se debe a la vibración de las cuerdas vocales, y perceptivamente da el tono agudo o grave a las voces. Hay tramos en que esta vibración no se da, que son los tramos no sonoros, por ejemplo en fricativas o habla susurrada. También puede verse que los armónicos en voces graves están más separados, y en voces agudas más próximos. Si se requiere separar diversos armónicos en una voz aguda es necesario más precisión en frecuencia, a costa de

la precisión en el tiempo.

Otra característica espectral sobresaliente son los *formantes*. El concepto de formante proviene de la lingüística, y se refiere a las bandas de energía en el plano tiempo-frecuencia. Habría que diferenciar entre formante fonético y formante analítico. El primero es resultado de la inspección visual de un espectrograma por un experto, marcando las zonas de contenido energético. El segundo se corresponde con las zonas de concentración de energía que surgen del análisis de un modelo de la señal de voz. Por ejemplo, si se modela la señal usando la transformada de Fourier como es habitual en telecomunicación, un formante analítico se corresponde con un máximo en el espectro. En el modelo de predicción lineal que se describe más adelante, un formante analítico se corresponde con un polo de cierta función de transferencia. En cualquier caso, si el modelo es fiel, habrá una adecuación entre los formantes fonéticos observados realmente y los formantes analíticos predichos. Para ver que un formante fonético es un objeto "natural" y un formante analítico un aspecto de un modelo, puede notarse que los formantes fonéticos se interrumpen en los intervalos en que la señal de voz cambia de régimen (de sonoro a no sonoro). En cambio los formantes analíticos, deducidos de un modelo, van a ser continuos en tanto se siga modelando la señal con el mismo procedimiento [Rabiner, 1999]. Otra nota de distinción, cuando dos bandas de energía se unen, se dice, en los estudios fonéticos, que uno de los formantes ha desaparecido. Sin embargo, analíticamente se describe de otro modo esta situación: dos formantes pueden mezclarse o incluso cruzarse [Riley, 1989].

El modelo de producción de voz que incorpora los formantes como parámetros descriptores de alto nivel tiene muchas aplicaciones. Históricamente fue desarrollado en el contexto de codificación (compresión) de voz, para aprovechar el ancho de banda disponible en las lineas telefónicas [Flanagan, 1960]. Asociada con esta técnica está la síntesis de voz, necesaria para recuperar la señal original [Atal and Hanauer, 1971]. La síntesis de voz es también la base de los sistemas de conversión texto-habla. Un sintetizador basado en formantes (frente a uno basado, por ejemplo, en "recortes" de sílaba) produce un habla fluida [Markel, 1976].

La descripción que hace el modelo de formantes es físicamente significativa, con lo que pueden deducirse aspectos sobre la posición de los órganos fonadores, y su movimiento (dinámica articulatoria) a partir de los parámetros. Relacionado con el desacoplamiento que el modelo hace de las diversas influencias en la señal (del

medio, de las características de los órganos, de la articulación, etc.) está que se pueden modificar estos parámetros, ajustándolos para conseguir algún efecto. Por ejemplo [Morris and Clements, 2002] proponen alterar los formantes medidos en hablas alteradas (en condiciones de baja presión atmosférica, o en alta gravedad, o habla susurrada) para resintetizar esa misma voz con los parámetros normalizados, es decir, de modo que suenen como voz producida en condiciones normales.

En cuanto al reconocimiento de voz, como comentan [Welling and Ney, 1998], casi todos los reconocedores están basados en coeficientes cepstrales o de banco de filtros. Sin embargo, hay características muy atractivas de los formantes para usarlos como componentes de plantillas de reconocimiento: son robustos frente a distorsión de canal o ruido; se pueden ajustar las condiciones de reconocimiento a las de entrenamiento; tienen "significado físico", es decir tienen un significado directo en los modelos de producción y percepción de habla. En resumen, sería muy útil disponer en tiempo real de la localización (es decir, frecuencia central y ancho de banda) de los formantes. Los parámetros estimados de formantes, como componentes del vector de características acústicas (o *template*), pueden ser usados directamente (*datos crudos* en la terminología de [Markel, 1976]) como se hace en [Welling and Ney, 1998] o [Garner and Holmes, 1998], o ser sometidos a un proceso de seguimiento de formantes [Holmes and Russell, 1999]. En aplicaciones de reconocimiento de voz la posición relativa de los formantes es el parámetro principal en la clasificación de vocales, y las trayectorias de los formantes están ligadas con los puntos de articulación, como se revisa en [Schmid and Barnard, 1995]. El reconocimiento del hablante también puede beneficiarse de esta técnica. La motivación para hallar formantes de [Snell and Milinazzo, 1993] es usar el número de ellos en ciertas bandas de frecuencia para verificar la identidad el hablante. En esta última referencia se introducen técnicas de análisis complejo del polinomio de predicción lineal que están en el origen del presente trabajo.

Predicción lineal

Para estimar los parámetros de los formantes es frecuente localizar los picos de resonancia del filtro de predicción (LPC, *Linear Prediction Code* [Markel, 1976]) obtenido a partir de segmentos de la señal de voz. La predicción lineal es una herramienta muy usada en varios aspectos de proceso de señal. Quizá por esto hay

gran variedad de exposiciones de sus fundamentos y de los algoritmos para llevarla a cabo, que resumimos a continuación.

Consideremos una señal de tiempo discreto $(x(t))_{t\in\mathbb{N}} = (x(0), x(1), \dots)$, es decir, tal que los valores $x(t) \in \mathbb{R}$ están distribuidos continuamente, y el índice de tiempo $t \in \mathbb{N}$ es discreto. Para un número natural $n \geq 1$, una *aproximación por predicción lineal* de $(x(t))_{t\in\mathbb{N}}$ de orden n consiste en unos números reales $a_1, a_2, ..., a_n$. Estos se denominan *coeficientes* de la predicción lineal. La señal $(\hat{x}(t))_{t\in\mathbb{N}}$ definida como

$$\hat{x}(t) = a_1x(t-1) + a_2x(t-2) + \cdots + a_nx(t-n) \text{ si } t \geq n \ (\hat{x}(t) = 0 \text{ si } t < n),$$

es la *predicción lineal* de $(x(t))$ con coeficientes $a_i, 1 \leq i \leq n$. La diferencia entre la señal y su predicción lineal es el *error* de la predicción, la señal $(e(t))_{t\in\mathbb{N}}$ definida como $e(t) = x(t) - \hat{x}(t)$. Como se ve, el error depende de los coeficientes a_i. También es evidente que conociendo solamente la señal de error (y los coeficientes), puede reconstruirse iterativamente el valor de la señal $x(t)$ en cada instante t, como $x(t) = e(t)$ para $t < n$, y

$$x(t) = \hat{x}(t) + e(t) = a_1x(t-1) + a_2x(t-2) + \cdots + a_nx(t-n) + e(t) \text{ para } t \geq n.$$

El *problema de la predicción lineal* es, dado $(x(t))$, hallar los coeficientes a_i de la predicción lineal que hacen mínima la señal de error $(e(t))$. Este mínimo suele tomarse en el sentido de la norma cuadrática

$$\|(e(t))_{t\in\mathbb{N}}\|_2 = \sqrt{\sum_{t=0}^{\infty} e(t)^2}.$$

Así se considera que cada predicción lineal $(\hat{x}(t))$ es un modelo de $(x(t))$ y que la de mínimo error es la mejor predicción. En la literatura, un primer grupo de cuestiones que aparece es la *exposición y desarrollo* de este problema de predicción lineal, a veces en un contexto de aproximación funcional (con el aparato matemático de proyecciones en espacios de Hilbert [Wiener, 1975]). Más frecuente es exponerlo mediante filtros, usando técnicas de señales y sistemas [Haykin, 1995]. También es frecuente introducirlo en un contexto de señales estocásticas y modelado AR [Box et al., 2008].

Independiente del contexto en que se introduce, con su terminología asociada, un segundo grupo de cuestiones atañen al *uso del modelo* de predicción lineal. Por ejemplo el espectro de potencia del modelo es más sencillo de describir que el espectro de potencia original [Priestley, 1981]. Como otro ejemplo de aplicación, si la señal ha sido producida aplicando diversos filtros (físicos o computacionales) desconocidos, puede resultar más fácil realizar la ingeniería inversa (deducir los filtros usados) a partir del modelo que a partir de la señal original. Esto pasa en el modelo vibración glotal - tracto vocal de la producción de voz [Rabiner, 1999].

Un tercer grupo de cuestiones son las relacionadas con la *solución efectiva* del problema de la producción lineal. Normalmente se hace restringiendo el análisis a señales definidas en un segmento finito $(x(t))_{t\in\{1,\dots,m\}}$. Señales de duración mayor que este segmento se dividen en ventanas en las que se realiza el análisis [Markel, 1976]. Este análisis se hace mediante el método de la autocorrelación (que describimos más adelante) o por el método de la covarianza. También es frecuente realizar adaptativamente la predicción lineal [Widrow and Stearns, 1985].

Comentamos ahora estas cuestiones (planteamiento del problema, usos del modelo y solución efectiva), y en el siguiente apartado se expondrá el problema concreto que motiva este trabajo. Para resolver el problema de la predicción lineal, es decir, para hallar los coeficientes a_i, se considera el error cuadrático

$$E = \sum_{t=0}^{\infty} e(t)^2 = \sum_{t=0}^{\infty} \left(x(t) - a_1 x(t-1) + a_2 x(t-2) + \cdots + a_n x(t-n)\right)^2$$

como si fuese una función de los coeficientes a_i. Suponiendo [Haykin, 1995] que los extremos de la función de error se alcanzan donde se anula la derivada, y que ese extremo es mínimo por convexidad, el error es mínimo donde se verifique:

$$\frac{\partial E}{\partial a_i} = 0 \text{ para } 1 \leq i \leq n$$

Desarrollando las derivadas, estas condiciones son equivalentes a

$$\frac{\partial E}{\partial a_i} = \frac{\partial}{\partial a_i} \sum_{t=0}^{\infty} \left(x(t) - \sum_{j=1}^{n} a_j x(t-j) \right)^2 =$$

$$= \sum_{t=0}^{\infty} \frac{\partial}{\partial a_i} \left(\left(x(t) - \sum_{j=1}^{n} a_j x(t-j) \right)^2 \right) =$$

$$= \sum_{t=0}^{\infty} 2 \frac{\partial}{\partial a_i} \left(x(t) - \sum_{j=1}^{n} a_j x(t-j) \right) \left(x(t) - \sum_{j=1}^{n} a_j x(t-j) \right) =$$

$$= -2 \sum_{t=0}^{\infty} x(t-i) \left(x(t) - \sum_{j=1}^{n} a_j x(t-j) \right) =$$

$$= -2 \sum_{t=0}^{\infty} x(t-i) e(t) = 0$$

Salvo el factor -2, las condiciones son $\sum_{t=0}^{\infty} x(t-i) \left(x(t) - \sum_{j=1}^{n} a_j x(t-j) \right) = 0$. Si se definen los *coeficientes de correlación* como $\phi(i,j) = \sum_{t=0}^{\infty} x(t-i)x(t-j)$, la anterior expresión puede ponerse como $\phi(i,0) = \sum_{j=1}^{n} a_j \phi(i,j)$. En la teoría de filtros óptimos estas condiciones se conocen como ecuaciones de Wiener-Hopf [Widrow and Stearns, 1985].

Con respecto al segundo grupo de cuestiones, las aplicaciones, el campo más natural es la teoría de filtros. Usando la técnica de la trasformada Z, denotamos $X(z), \hat{X}(z)$ y $E(z)$ a las transformaciones de $x(t), \hat{x}(t)$ y $e(t)$. La reconstrucción de $x(t)$ por predicción lineal puede verse como la aplicación de un filtro $F(z)$ a esta señal, $\hat{X}(z) = F(z)X(z)$, siendo $F(z) = -\sum_{j=1}^{n} a_j z^{-j}$. Usando este filtro de predicción lineal, la señal de error se produce con $E(z) = (1 - F(z))X(z)$. Llamando *polinomio LPC* a $A(z) = 1 - F(z)$, tenemos:

$$X(t) = \frac{E(z)}{A(z)}$$

Esta expresión tiene una importante interpretación práctica: si disponemos de una suposición razonable sobre la forma de la señal de error $E(z)$, la señal que se pretende modelar, $X(z)$, es el resultado de aplicar al error un filtro con función de transferencia $H(z) = \frac{1}{A(z)}$. En nuestro caso, $x(t)$ es una señal de voz y dos suposiciones razonables sobre la forma que puede tomar $E(z)$ son la de tren de impulsos (en periodos de habla vocalizados, es decir, con actividad de las cuerdas

vocales) o la de ruido blanco (en habla no vocalizada) [Rabiner, 1999].

El tercer grupo de cuestiones atañe a los cálculos prácticos con los que llevar a cabo este planteamiento. Las operaciones anteriores, con sumatorios y correlaciones que se extienden sobre todo el eje de tiempos, deben truncarse para poder realizar cálculos con efectividad. Se suele restringir el análisis a una ventana de la señal $x(t)$, a lo largo de la cual es razonable suponer que se mantienen las características que recoge el modelado LPC. Con este truncamiento, con una ventana de longitud N, la expresión del error ventaneado queda $E_N = \sum_{t=n}^{N-1} e(t)^2$, y la correlación $\phi_N(i,j) = \sum_{t=\text{máx}(i,j)}^{N-1-\text{mín}(i,j)} x(t-i)x(t-j)$. Esto último puede ponerse como $\phi_N(i,j) = \sum_{t=0}^{N-1+|i-j|} x(t)x(t-|i-j|)$, que es una expresión que depende solo de $|i-j|$, es decir $\phi_N(i,j) = r_N(|i-j|)$. Esto permite expresar las ecuaciones de Wiener-Hopf en forma matricial: de $r_N(i) = \sum_{j=1}^{n} a_j r_N(|i-j|)$ para $0 \leq i \leq n$ pasamos a

$$\begin{pmatrix} r_N(0) & r_N(1) & r_N(2) & \dots & r_N(n-1) \\ r_N(1) & r_N(0) & r_N(1) & \dots & r_N(n-2) \\ r_N(2) & r_N(1) & r_N(0) & \dots & r_N(n-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_N(n-1) & r_N(n-2) & r_N(n-3) & \dots & r_N(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} r_N(1) \\ r_N(2) \\ r_N(3) \\ \vdots \\ r_N(n) \end{pmatrix}$$

Para hallar las incógnitas a_i, suelen usarse métodos de álgebra lineal numérica adaptados al hecho de que esta matriz de coeficientes de correlación es de tipo Toeplitz (es decir, simétrica y con valores constantes a lo largo de diagonales paralelas a la principal), como el algoritmo de Levinson-Durbin [Haykin, 1995].

Formantes y modelo de producción

La aproximación inicial a la señal de voz, mediante un análisis tiempo-frecuencia como los espectrogramas anteriores, suele complementarse con el modelado por predicción lineal. Las características del filtro LPC pueden ponerse en correspondencia con los órganos de fonación. El funcionamiento de la glotis y el tracto vocal

sugiere un modelo de producción de la señal de voz como un generador de impulsos seguido de un resonador. Con más detalle, las tres componentes serían glotis, tracto vocal, radiación desde los labios (modelo de Liljencrants-Fant [Markel, 1976]).

En términos de funciones de transferencia (ver [Oppenheim et al., 1996] como referencia para señales y sistemas y [Ogata, 2010] para técnicas de control y análisis del lugar de las raíces), la función de la glotis y la de la radiación son sencillas de describir, y aportan ceros a la señal de voz. El pre-énfasis consiste en eliminar estos ceros mediante un filtro de +6 decibelios por octava, como explica [Duncan and Jack, 1988] (se considera que la glotis aporta +12db/o, y la radiación de los labios -6db/o, por tanto se requiere +6 de filtrado para aislar la función de transferencia del tracto vocal, que aporta los formantes). El resultado puede verse en la figura 1.4.

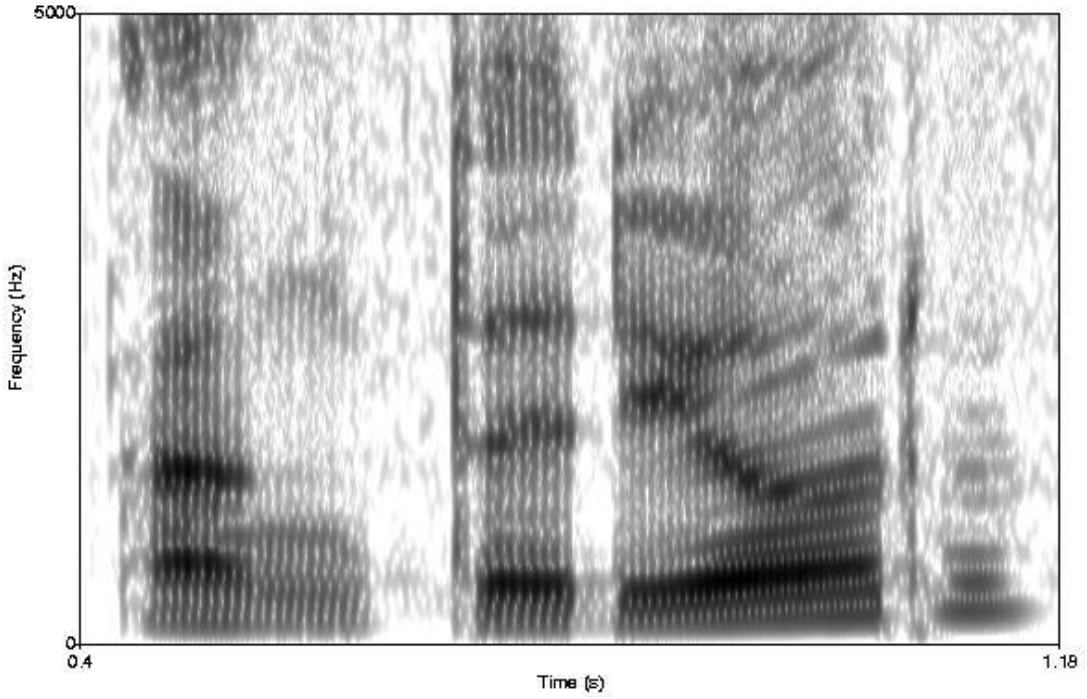


Figura 1.4: Espectrograma de *anterior*, con ventana Hamming de duración 10 ms, y pre-énfasis de 6 dB. Notemos como al aumentar la frecuencia fundamental al final de la enunciación, el intervalo entre picos glotales baja de los 10 ms, y el rayado vertical da paso a un rayado horizontal.

Tras este filtrado, el espectro resultante muestra unas bandas donde se concentra el aporte del tracto vocal al espectro total. Este espectro puede considerarse de

modulación, siendo la señal de la glotis la portadora. Esto es lo que se conoce como formante: bandas de energía en el plano tiempo-frecuencia. Como comenta [Markel, 1976], el filtrado conforme al modelo de producción realza los formantes superiores (denotados como F_1, F_2, ..., frente a F_0, la frecuencia fundamental).

Para tener una expresión sencilla de la función de transferencia del tracto vocal, se aproxima con un filtro todo-polos. Esta hipótesis simplificadora es razonable, pues el tracto vocal puede asimilarse físicamente a un tubo sin pérdidas. Por ejemplo, aunque el análisis de [Kapilow et al., 1999] es continuo (es decir, no basado en predicción lineal), respalda la hipótesis de que el espectro del tracto no tiene ceros. La señal de habla resultante de aplicar el filtro de preénfasis sí puede tenerlos debido a la fuente de señal de la glotis, o a una excitación acústica en el tracto (como *cliks* o fricaciones), o a un acoplamiento nasal. Teniendo en cuenta la distinción antes comentada entre formantes observados - formantes del modelo, varios trabajos examinan la validez de la hipótesis todo-polos. En el artículo fundacional de [McCandless, 1974] se describen las frecuencias de atenuación, indicando la presencia de ceros en el tracto vocal. Con el nombre de antiformantes, o nasalformantes, algunos autores los incorporan a la descripción del tracto, en un modelo más preciso. En [Duncan and Jack, 1988] y [Ouaaline and Radouane, 1998] se describen en detalle los efectos que tienen los ceros de la señal en nasales, fricativas y *stops* (plosivas). Estas realizaciones están producidas por causas externas al tracto vocal, y por tanto escapan de las hipótesis del modelo LPC. Es necesario desarrollar otros métodos para detectarlos (como el modelado ARMA de series temporales en general [Box et al., 2008]). Para el caso particular de modelos de voz bajo hipótesis más relajadas se han propuesto algoritmos de "corrección de ceros", como los de factorización de [Starer, 1990], originales de [Orfanidis and Vail, 1986], o el referido en [Ouaaline and Radouane, 1998], que describe uno basado en filtrado adaptativo.

Resumiendo, se dispone de un modelo sencillo de producción de la señal de voz, que incorpora sus características más importantes, conocido como modelo de Liljencrants-Fant [Rabiner, 1999]. Con la hipótesis adicional de que el tracto vocal tiene un espectro todo-polos, puede aplicarse las técnicas de análisis LPC para calcular sus parámetros. Entre los aspectos de los que no da cuenta el modelo está la naturaleza del generador de impulsos (vibración-no vibración de las cuerdas), y la simplificación que introduce la hipótesis todo-polos.

Lo más habitual es ceñirse al modelado LPC o al modelo espectral, pero con indicadores sobre la validez del modelo. Por ejemplo [Riley, 1989] comenta los problemas de LPC, con imágenes de polos que desaparecen por influencia nasal. Este autor defiende el "compromiso mínimo": no ajustar la señal de habla a un modelo tan restrictivo como LPC al principio del análisis, sino tras haber extraído conclusiones de bajo nivel (como sonoro-no sonoro), y etiquetar explícitamente la señal con estas conclusiones. Otro trabajo que hace uso de indicadores de validez es [Bermúdez et al., 2000], donde se desarrollan métodos de aproximación de formantes sobre LPC, tras detectar el régimen sonoro-sordo.

Con respecto al seguimiento (*tracking*) a lo largo del tiempo, ya [McCandless, 1974] estudia cómo concatenar las zonas de energía del espectro para obtener los formantes, en un compromiso de la continuidad con el cambio. Pero el trazado de formantes sobre el espectrograma no es automatizable, pues responde a criterios visuales y fonéticos no formalizados. En el trabajo de [Kopec, 1986] sobre el seguimiento automático mediante modelos ocultos de Markov, se comenta esta falta de precisión conceptual en los trazados manuales de formantes que usa como entrenamiento. También se han intentado modelar las transiciones suaves (por ejemplo, sin consonantes fricativas) en el seguimiento de formantes mediante la respuesta a impulso de un sistema de segundo orden [Kawahara et al., 1999].

Estimación y uso de los formantes

Entre los parámetros más importantes de la señal de voz están la sonoridad, la frecuencia fundamental y los formantes. Para medir los valores de estos parámetros, primero se estima si hay sonoridad o sordez, es decir, vibración de las cuerdas o no. Esto puede hacerse con la energía, o comprobando los cruces por cero [Bermúdez et al., 2000]. El siguiente parámetro, en orden de dificultad de medición, en una señal periódica es su frecuencia fundamental F_0. En el caso de la señal de voz, al no ser exactamente periódica, F_0 es difícil de estimar. Puede estar incluso ausente, si se da una fonación sorda, y en los casos sonoros, puede variar bastante incluso en una misma palabra. Además, en caso de F_0 muy grave, puede llegar a confundirse con el primer formante F_1 [Parsons, 1987]. El método canónico para hallar la frecuencia fundamental de una señal es la autocorrelación de la misma, cuyo segundo máximo nos da la amplitud de onda, y el segundo método usado es el análisis homomórfico

(*cepstrum* [Deller et al., 2000]) para separar la contribución del tracto de la de la glotis. Sin embargo en presencia de ruido estos métodos pierden rendimiento [Parsons, 1987, Olive, 1971].

Como se ha comentado, las técnicas que aparecen con más frecuencia en la literatura para la extracción de formantes se basan en el modelado por predicción lineal. Hay tres clases de métodos para la detección de formantes a partir de este modelo todo polos de la señal de habla, como se revisa en [García Zapata et al., 2004a]. Primero, el espectro del modelo LPC es una aproximación suavizada del espectro de potencia de la señal, y por tanto los máximos del espectro del modelo se corresponden con la frecuencia central de los formantes [Rabiner, 1999]. Una segunda clase de métodos consiste en escoger (por medios estadísticos o adaptativos) un conjunto de valores para las posiciones de formantes y anchos de banda, que encajen lo mejor posible en la distribución de energía del espectro de señal [Olive, 1971, Welling and Ney, 1998]. Finalmente, otra aproximación es el análisis de los polos de la función de transferencia $\frac{1}{A(z)}$ del modelo LPC fuera de la circunferencia unidad (*off-axis* según [McCandless, 1974]). Una vez que se tiene el polinomio predictor $A(z)$, los formantes se obtienen de las raíces de $A(z) = 0$: cada par de raíces complejas nos da la frecuencia y ancho de banda correspondiente a un formante. Así, los polos, que están en el interior del círculo unidad, son los elementos responsables de la distribución espectral [Duncan and Jack, 1988, Snell and Milinazzo, 1993].

Como pasa con la frecuencia fundamental, aunque en principio parece un parámetro directamente medible siguiendo alguno de los métodos de las tres clases mencionadas, en la práctica surgen ciertas dificultades. La asignación de máximos espectrales (picos) a formantes no es directa, sobre todo si se usan estimadores espectrales tipo Fourier, que muestran gran cantidad de picos. Y aunque la estimación espectral que resulta de LPC es más suave que con Fourier, los dos o tres picos disponibles en LPC a veces se adaptan a máximos del espectro que no corresponden a formantes. Otro problema complementario es el de dos máximos espectrales que con el tiempo se funden en uno. Un tercer problema aparece en las voces agudas, en las que los picos de la envolvente estimada tienden a acercarse a los picos armónicos, en vez de a los máximos debidos a formante [Parsons, 1987].

Por otro lado, con el modelo de producción de habla antedicho, el filtro del

tracto vocal (responsable en último término de los formantes) puede modelarse mediante LPC. Los parámetros de la función de transferencia que se consideran normalmente son los coeficientes del polinomio predictor. Se puede hallar una estimación del espectro suavizado mediante la evaluación en el círculo unidad descrita en [Rabiner, 1999]. Esta evaluación en el círculo unidad a veces no muestra los máximos espectrales suficientemente separados, al estar los polos del modelo dentro del círculo unidad. [McCandless, 1974, Duncan and Jack, 1988] proponen evaluar el espectro en un circulo interior. [Kang and Coulter, 1976, Parsons, 1987] proponen el método del par de linea (*line pair*), que modifica los coeficientes del polinomio para que sus raíces (polos del modelo) se acerquen al círculo unidad. Ambas aproximaciones pierden efectividad por las interacciones no lineales que se dan entre los coeficientes y las raíces del polinomio.

Un conjunto equivalente de parámetros es el de las raíces del polinomio, los polos del modelo LPC. Como comenta [Olive, 1992], el modelo LPC se adapta al espectro de la señal de voz, pero los coeficientes no tienen un significado físico claro. Por ejemplo no es fácil saber qué coeficientes LPC modificar para que la señal de voz cambie de "e" a "i", o saber a partir de estos parámetros si es vocal o consonante. Los polos sí que son físicamente significativos y se corresponden directamente con formantes, con frecuencia y ancho de banda.

Para decidir qué raíces del polinomio predictor se corresponden a formantes, [Markel, 1976] escoge las tres de mayor módulo. El mismo artículo sugiere que independientemente de la frecuencia de muestreo, es necesario un par de polos complejos para cada banda de 700 Hz, es decir, que el grado de LPC debe ser 4 o 5 (para recoger los formantes) más la cantidad correspondiente a la frecuencia de muestreo (para recoger la forma general del espectro). Asimismo, los tres primeros formantes para las vocales deben estar por debajo de 3 kHz.

Nuestro interés en la extracción de formantes viene del desarrollo de una metodología de reconocimiento de habla en entornos adversos en el proyecto IVORY [Gómez, 1998], y posteriormente en el proyecto DIARCA [Diaz Martin et al., 2001]. En esta metodología, la señal de entrada es filtrada adaptativamente para la cancelación de ruido antes de la fase de reconocimiento. Como subproducto de este filtrado, se obtiene el polinomio LPC (inverso de la función de trasferencia) con un alto grado (como referencia, 32 o 64 coeficientes). La información sobre formantes extraída de este polinomio puede ser usada en posteriores fases de reconocimien-

to. Las dos clases de métodos de extracción citados al principio de este apartado, máximos espectrales y ajuste estadístico, no se pueden aplicar con fiabilidad a este modelo de grado alto debido a los efectos de mezcla de formantes, y a la aparición de picos espurios [McCandless, 1974]. La tercera clase de métodos de localización de formantes, la basada en las raíces del polinomio predictor, es la más adecuada a esta situación.

La factorización de un polinomio (o la extracción de raíces, que es la factorización total, en factores lineales) es un problema que se plantea con frecuencia en procesamiento de señal: con transformadas Z de señales de voz en la construcción de plantillas para reconocimiento de voz [Gómez, 1998], o como descripción de alto nivel de señales de electroencefalograma [W. Philips, 1992], o para usarlos como parámetros para una representación comprimida de una señal sonora [Starer, 1990].

No obstante, un análisis LPC como el descrito en la tercera clase de métodos de detección de formantes requiere la extracción de raíces de un polinomio de grado alrededor de 32, al menos 40 veces por segundo. Los métodos para hallar raíces presentes en los paquetes de cálculo numérico de propósito general no pueden abordar esto en tiempo real en maquinaria no especializada. En realidad, no es necesario hallar todas las raíces: las que más influyen en la forma del espectro son aquellas próximas a la circunferencia unidad. Se puede estimar, con las cifras dadas como referencia, que las raíces significativas tienen radio mayor que 0'95, pues las que tengan menor radio corresponden a filtros con una respuesta trasiente de mayor duración que la ventana de muestreo. Por tanto, sería útil como estimador de la forma del espectro el conjunto de polos de la función de transferencia (de raíces del polinomio denominador) situados en la corona de radio $0'95 < r < 1$. Pero al aplicar métodos clásicos de cálculo numérico, se ve que no es posible condicionar así la búsqueda. Con un algoritmo para encontrar raíces genérico (de biblioteca), es inevitable hallar las 31 raíces (aunque luego se seleccionen solo las más cercanas a la circunferencia unidad). No se puede restringir la búsqueda a regiones específicas del plano complejo. Además, el tener que hallar todas las raíces requiere trabajar con muy alta precisión en los cálculos intermedios [Ralston and Rabinowitz, 1978a]. En general, los métodos basados en raíces tienen una gran carga computacional, y no se suelen usar en la práctica.

Consideremos por ejemplo el método de Newton, que no permite "concentrar el

esfuerzo" en una región del plano. Los resultados teóricos nos aseguran la convergencia a una raíz dada si se inicia el método cerca de esta. Pero en general, aunque se converja a una raíz, esta puede estar arbitrariamente lejos del punto inicial. En contraposición, otros métodos permiten condicionar la búsqueda a una región especificada. Por ejemplo el método de Bisección, que halla una raíz (en la recta real) de una función contenida en un intervalo. Cualquier subintervalo de este, si la función toma valores de distinto signo en sus extremos, contiene al menos una raíz. Aplicando este criterio a las mitades que surgen de dividir el intervalo inicial, se detecta en cuál está contenida alguna de estas. A este subintervalo puede aplicarse recursivamente el método, hasta alcanzar la precisión deseada. En el plano complejo también se puede utilizar un procedimiento análogo: cada región plana se divide en subregiones, en cada una de las cuales se aplica un criterio para la presencia de raíces, y así recursivamente. En nuestro caso, por partición reiterada de la corona $0'95 < r < 1$ aplicando este método, pueden encontrarse las raíces con la condición impuesta. Vamos a ver cómo llevar a cabo este esquema de cálculo.

1.2. Métodos tradicionales para hallar raíces

Hasta mediados del siglo pasado, el proceso de hallar raíces de polinomios seguía el siguiente esquema: una primera fase, de *separación*, divide la recta real en segmentos de modo que en cada uno de ellos solo hay una raíz del polinomio. En una segunda fase, de *aproximación*, la raíz existente en cada segmento de la fase anterior se aproxima por un método iterativo. Procedimientos usados en la fase de separación son la regla de los signos de Descartes, o las sucesiones de Sturm para raíces reales [Ralston and Rabinowitz, 1978a]. También pueden separarse regiones del plano complejo, si se buscan raíces complejas, mediante el algoritmo de Schur-Cohn [Henrici, 1988]. En la fase de aproximación, dada la escasa precisión que era factible obtener manualmente, unos pocos pasos de cualquier método iterativo son suficientes para aproximar la raíz.

A la hora de automatizar este proceso surgen varios inconvenientes: la fase de separación es costosa de llevar a cabo sin una intuición previa del tamaño que pueden tener los segmentos (o regiones) de separación. Sin una separación adecuada, los algoritmos iterativos no convergen a una raíz cercana a la estimación inicial. Para evitar la no convergencia, un recurso frecuente es variar la estimación inicial.

Así, tras varios intentos, es más probable la convergencia a alguna raíz. Pero esto dificulta el análisis del método, y hace que el coste sea imponderable. Además, frecuentemente no se desea una raíz cualquiera, sino las que cumplan cierta propiedad, por ejemplo la de mayor módulo, como sucede en el análisis adaptativo de componentes espectrales [Haykin, 1995] o las próximas al circulo unidad, en el procesado mediante LPC [Rabiner, 1999]. En estos casos es necesario encontrar todas las raíces, y luego filtrarlas por la propiedad deseada. Al aproximar todas las raíces, para evitar converger varias veces a una raíz ya encontrada, se aplica una deflación al polinomio original. La deflación consiste en dividir el polinomio por el factor correspondiente a la raíz encontrada. Con esto se consigue que, al aplicar de nuevo el método al polinomio resultado de la deflacción, la nueva raíz encontrada no sea otra vez la misma. Pero la deflación es un método costoso [Ralston and Rabinowitz, 1978a] y además, la división de polinomios es numéricamente inestable (más adelante se introduce el concepto de estabilidad algorítmica).

Los métodos más extendidos actualmente se basan en este esquema, cuyas limitaciones impiden que se pueda encontrar con seguridad raíces de polinomios de grado mayor que unas pocas decenas [Lang and Frenzel, 1994]. Aunque esto venía siendo suficiente, nuevas aplicaciones demandan métodos con capacidad para mayores grados. También es de interés hallar raíces de polinomios cuyos coeficientes están dados con baja precisión (como pasa en procesado de voz), y la deflación requiere aumentarla artificialmente, lo que además de elevar el coste altera el modelo. Para lograr estos dos objetivos hemos recurrido a los denominados métodos geométricos, y que pueden verse como una automatización de la fase de separación.

En el siguiente apartado se desarrolla la clasificación de los métodos de aproximación de raíces en iterativos o geométricos. Para cada clase, se describe el esquema general y se dan unos métodos de ejemplo. Posteriormente, en el siguiente apartado, se describen los conceptos o herramientas teóricas que aparecen en la literatura para su análisis.

Métodos iterativos y geométricos

Los métodos iterativos (MI) se basan en un bucle de estimación - corrección de error, que converge, en condiciones bien definidas, a un punto complejo tan cerca de una raíz como sea necesario. Los métodos de este tipo son rápidos (convergencia

más que lineal, en general) y su análisis, incluyendo la prueba de su corrección, y la determinación de los recursos necesarios, se basa en herramientas bastante extendidas de análisis matemático (derivada, aplicaciones contractivas o teoría de punto fijo [Ralston and Rabinowitz, 1978a]). Esto hace que se denominen en ocasiones métodos analíticos.

El arquetipo de los MI es el de Newton, con decenas de adaptaciones, para varias variables, variable compleja, evaluación conjunta del polinomio y su derivada, etc. [Marden, 1966]. Junto con los métodos iterativos de Müller, Legendre y otros, constituye la solución más común al problema de hallar raíces de polinomios. Entre los MI más extendidos en bibliotecas numéricas cabe destacar el de Laguerre, o el de Jenkins-Traub. Puede verse una descripción sencilla y una comparativa de estos métodos en [Mekwi, 2001]. El procedimiento incorporado en paquetes matemáticos matriciales (como LAPACK [Anderson et al., 1999] o MATLAB [Smith et al., 1976]) es un MI basado en los autovectores de la matriz compañera del polinomio (*companion matrix*, véase [Fortune, 2002] o [Edelman and Murakami, 1995]).

Un MI, por lo general, converge rápidamente para la mayor parte de polinomios de grado moderado (hasta aproximadamente 50 [Pan, 1997]). Sin embargo, aún para estos grados, los MI son inadecuados para ciertas clases de polinomios (como aquellos con raíces múltiples, o agrupadas -*clusters*-: cada método tiene su clase particular de polinomios para los que no es apto), o para polinomios específicos que muestran un mal condicionamiento, como el polinomio de Wilkinson [Wilkinson, 1965]. Para enfrentarse con este problema, es necesario aplicar reglas heurísticas en caso de dificultades, tales como escoger otras aproximaciones iniciales, cambiar el método aplicado, o usar aritmética de precisión variable [Pan, 1997]. Puede decirse que hay cientos, si no miles [Householder, 1970] de métodos iterativos.

Estos métodos encuentran una raíz solamente. Si se necesita encontrar todas las raíces, o las que cumplen cierta condición, es necesario aplicar una deflación al polinomio original, lo que es costoso. Para ciertas aplicaciones se han desarrollado MI específicos. Por ejemplo en aplicaciones de procesamiento de señal, o robótica, el sistema a estudiar o controlar suele estar descrito por un modelo polinómico que varía con el tiempo, produciendo un flujo de polinomios de alto grado. La frecuencia con que se requiere hallar sus raíces puede ser demasiado grande para los métodos genéricos. Para hallar las raíces de módulo máximo se usa el método de Graeffe [Malajovich and Zubelli, 2001] o el de Bernouilli [Young and Gregory, 2012]). Para

hallar todas las raíces está el método de Durand-Kerner [Aberth, 1973], [Farmer and Loizou, 1975]. En particular en robótica se usa el método de continuación homotópica [Sommese and Wampler, 2005], y otros mixtos simbólico-numéricos [McCarthy, 2011]. Estos métodos adaptados también muestran los problemas de coste y rigor asociados a los MI (como describen las referencias citadas en [Pan, 1997]), que detallamos en el siguiente apartado.

Los métodos geométricos (MG), también llamados de búsqueda, se basan en el refinamiento recursivo de una serie de condiciones que restringen la búsqueda. Como ejemplo, consideremos el método de la bisección para hallar ceros de funciones reales continuas. Se basa en el teorema de Bolzano, que dice que si una función continua $f : [a, b] \to \mathbb{R}$, en un intervalo $[a, b]$ de la recta real tiene signos opuestos en sus extremos (es decir, $f(a) \cdot f(b) < 0$), entonces tiene al menos una raíz en ese intervalo. El MG de bisección se basa en la hipótesis de este teorema. Partiendo de un intervalo inicial que cumpla esta condición, y viendo cuál de las mitades que surgen de dividir el intervalo inicial verifica a su vez la condición, se detecta cuál contiene al menos una raíz. A este subintervalo puede aplicarse recursivamente el método de bisección, hasta alcanzar la precisión deseada.

Otras condiciones distintas a la hipótesis del teorema de Bolzano pueden aplicarse en la recta, dando lugar a otros MG. Por ejemplo suelen aplicarse las sucesiones de Sturm [Henrici, 1988], más costosas de evaluar que el signo en los extremos, pero que dan más información sobre el número de raíces. También hay condiciones que aseguran la existencia de raíces en regiones del plano complejo, y que se puede utilizar en un procedimiento bidimensional análogo: cada región plana se divide en subregiones, en cada una de las cuales se comprueba cierta condición para la presencia de raíces, y así recursivamente. Como los MG se basan en la delimitación de regiones que pueden contener raíces, a menudo se denominan métodos de *bracketing* (horquillado, enmarcado) por analogía con el de bisección.

Los métodos basados en nociones geométricas como las mencionadas son igualmente válidos para todos los polinomios. Esto los diferencia de los MI. Además, la uniformidad, que es la ausencia de casos especiales, facilita el análisis de la complejidad de los MG. Los estudios teóricos de complejidad del problema del cálculo de raíces han motivado el desarrollo de métodos geométricos. Por otro lado, las aplicaciones prácticas en procesamiento de señal o robótica necesitan métodos que permitan enfocar la búsqueda de raíces a una zona pre-especificada del plano com-

plejo, como se ha descrito. Esta acotación de la búsqueda produce un ahorro en cantidad de cómputo con respecto a los MI, que no permiten tal acotación. MG basados en nociones geométricas distintas al recuento del número de raíces en subregiones son el de test de inclusión de Weyl [Henrici, 1988], [Yakoubsohn, 2005], o el procedimiento de escisión (*splitting*) de Graeffe [Bini and Pan, 1996].

En general, los MG tienen un patrón común. Consideremos de nuevo el método de la bisección. Este método consiste en utilizar de forma recursiva los signos distintos en los extremos de un intervalo para construir una sucesión de intervalos anidados (y por lo tanto cada vez más pequeños) que contienen una raíz real. Estos son los componentes prototípicos de un MG: un *test de inclusión*, para decidir si hay alguna raíz en una región, ya sea de la recta o del plano, y un *procedimiento recursivo* para obtener localizaciones cada vez más pequeñas de las raíces deseadas.

Tratamos dos ejemplos en el plano para comparación: el método de Lehmer-Schur y el de Weyl. El método de Lehmer-Schur [Lehmer, 1961] es un método de enmarcado bidimensional basado en un test de inclusión de Schur y Cohn, que dice cuándo un polinomio tiene raíces dentro de un círculo. Así, el área de interés se cubre con círculos, y se aplica este criterio a cada uno de ellos. Después, los círculos que contienen alguna raíz se cubren a su vez por círculos de menor radio, y así sucesivamente, hasta que se alcanza la precisión requerida. El procedimiento recursivo conlleva cierta ineficiencia porque los círculos se superponen y puede encontrarse repetidamente la misma raíz. El método de enmarcado de Weyl [Henrici, 1988] se basa en un test de inclusión que se aplica a regiones cuadradas. Una descomposición en cuadrados de la región de interés conduce de modo natural a un árbol cuadrático (*quadtree* [Aho et al., 1983]). Este consiste en un árbol de búsqueda en el que cada nodo es un cuadrado, enlazado a sus cuatro sub-cuadrados, de la mitad de lado. Se han propuesto varios tests de inclusión para cuadrados, por Weyl y otros [Pan, 1997], para ser utilizados en procedimientos recursivos de búsqueda en árbol cuadrático.

Son menos frecuentes los predicados, que responden afirmativa o negativamente sobre la presencia de raíces en una región, que los tests, que responden afirmativamente o no se deciden. Ejemplos de test de inclusión aparecen en los trabajos citados en [Dickenstein and Emiris, 2005]: [Neff and Reif, 1996a], [Neff, 1994], [Cardinal, 1996], [Stetter, 1996], [Kirrinnis, 1998]. Basados en predicados de inclusión están [Henrici and Gargantini, 1969] o [Dedieu and Yakoubsohn, 1993]. En [Pan,

1996a] se insiste en que un aspecto crucial a efectos de complejidad de estos métodos es que la descomposición "divide y vencerás" se haga en partes equilibradas.

Las pruebas en las que se basan los MG suelen ser geométricas, pero también hay propuestas para usar predicados que usan expresiones algebraicas más generales, como aproximaciones p-ádicas [Lenstra Jr, 1999]. En esta línea es similar [Burr and Krahmer, 2012], donde los predicados se organizan como un árbol sin interpretación geométrica directa, y los métodos algebraico-simbólicos descritos en [Elkadi and Mourrain, 2005].

En este trabajo no usamos test de inclusión, ni predicados con expresiones algebraicas, sino el número de vueltas o índice de curvas planas cerradas. Otros métodos geométricos basados en el número de vueltas, son los de [Ying and Katz, 1988], [Herlocker and Ely, 1995], [Noureddine and Fellah, 2005] o [Yap and Sagraloff, 2011], aproximando la integral de Cauchy con las técnicas generales de acotación del error de integración numérica [Ralston and Rabinowitz, 1978b], en la línea de los trabajos [Delves and Lyness, 1967], [Sakurai et al., 2003]. Comparaciones empíricas de algunos de estos métodos pueden verse en [Kamath, 2010]. Se han propuesto varios técnicas específicas para la integral de Cauchy [Kravanja and Van Barel, 2000], [Suzuki, 2001].

Un enfoque diferente se puede encontrar en [Wilf, 1978] o [Collins, 1977], que usan sucesiones de Sturm para encontrar el número de cruces de una curva por la parte positiva del eje de las abscisas. Hay que considerar como una desventaja de estos métodos que solo sean válidos para formas específicas del contorno que contiene las raíces (circular en Suzuki, rectangular en Wilf y Collins), y muestran varios problemas, como que los métodos de integración numérica necesitan aritmética de precisión arbitraria [Knuth, 1981], y las sucesiones de Sturm requieren usar un paquete de álgebra simbólica.

En este trabajo se sigue otro enfoque, que puede remontarse hasta [Henrici, 1988], basado en una muestra de puntos del contorno de la región que contiene las raíces. El método descrito es aplicable a curvas genéricas, sin necesidad de recurrir a precisión múltiple. Se ha implementado y comparado con varios algoritmos de cálculo de raíces, con resultados favorables [García Zapata and Díaz Martín, 2008]. La comparación se ha realizado también sobre los polinomios de alto grado que surgen en proceso de señal.

Métodos derivados del de Henrici se han usado en el cálculo recursivo de raíces,

ya sea para aplicaciones prácticas [Snell and Milinazzo, 1993] o para estudios teóricos sobre la complejidad del cálculo de raíces [Renegar, 1987], [Pan, 1997]. En los trabajos de [Ying and Katz, 1988] o en [Ko et al., 2008] se encuentran enunciados precisos sobre las condiciones en que se puede usar el índice de curvas planas en algoritmos de cálculo de raíces, y sugerencias sobre cómo gestionar los casos singulares.

Los métodos teóricamente óptimos del problema del cálculo de raíces, en el sentido de la complejidad computacional, son de tipo geométrico [Pan, 1996b], [Neff and Reif, 1996b], [Schönhage, 1982], pero no están muy extendidos en la práctica. Los MG son más difíciles de implementar que los métodos iterativos, ya que requieren tipos de datos para objetos geométricos y variables complejas, y procedimientos de búsqueda en árbol o de *backtracking* para el flujo de control [Brassard and Bratley, 1988]. Sin embargo, los métodos basados en relaciones geométricas son válidos para todos los polinomios. Esta uniformidad permite un análisis de la complejidad de tales métodos. Esta es la razón por la cual los estudios teóricos de complejidad han sido la fuerza motriz en el desarrollo de MG [Renegar, 1987]. Por ejemplo, supongamos que se requieren las raíces de un polinomio hasta b bits, es decir con una precisión de 2^{-b}. El número de multiplicaciones necesarias para extraer todas las raíces de un polinomio de grado n, con esta precisión, es $O(n^2 \log n \log b)$ usando el MG de [Pan, 1997]. Para el método de Newton no existen semejantes estimaciones de coste, ni para otros MI (ver [Traub and Woźniakowski, 1979], [Forster, 1992]). Sin embargo en la práctica la mayoría de las aplicaciones de búsqueda de raíces se basan en MI.

Como se ha comentado, tradicionalmente los métodos geométricos se usaban como una parte heurística en el proceso de hallar raíces de polinomios, en una fase previa de separación. En un fase posterior de aproximación se usaban métodos iterativos. Las implementaciones de MG, por lo general, también incluyen, por eficiencia, algún método iterativo que interviene al final del cálculo, como la aplicación del método de Newton, cuando la búsqueda *quadtree* alcanza una subregión donde se da una convergencia rápida.

Otros cambios que se hacen en las implementaciones prácticas de un MG genérico están relacionados con aspectos numéricos de los tests de inclusión. En las implementaciones con aritmética de coma flotante, es decir, con una precisión finita fija, los errores de redondeo se acumulan y hacen que el cálculo pierda toda

fiabilidad [Gourdon, 1993]. Sin embargo, en implementaciones con aritmética de precisión arbitraria, sin errores de redondeo, la precisión puede crecer indefinidamente [Pan, 1997], y por tanto los recursos computacionales necesarios. Estos problemas de precisión numérica han sido un obstáculo para el uso práctico de MG. En este trabajo se demostrará que el índice de curvas planas puede calcularse con fiabilidad, identificando las curvas que requerirían más recursos de los asignados, y usando el índice en un MG eficiente.

Este paso de métodos iterativos a geométricos, que requiere análisis más elaborados, se da también en otras áreas del cálculo numérico: autovalores mediante divide-y-vencerás [Demmel, 1997], ramificacion y poda en programación lineal [Padberg, 1999], o descomposición de dominio en resolución de ecuaciones diferenciales [Toselli and Widlund, 2005]. Para tener una mayor perspectiva de esta evolución desde los métodos iterativos, basados en un bucle, hacia otros basados en recursividad, vamos a usar un famoso ejemplo no numérico: el problema de la ordenación de una lista. Los diversos algoritmos conocidos para este problema pueden clasificarse también como iterativos (basados en un bucle) o como recursivos. Los métodos de bucle, como el algoritmo de inserción, o el de la burbuja [Aho et al., 1983] producen resultados parciales más ordenados en cada iteración, hasta llegar a la solución (mediante sucesivas aproximaciones; "más ordenado" significa que contiene una sublista ordenada de mayor longitud). Los métodos recursivos (*quicksort* o *mergesort*) llegan a la solución sin pasar por refinamientos sucesivos, y son más efectivos. Como contrapartida, para analizar el coste de algoritmos de ordenación iterativos es suficiente con combinatoria elemental, mientras que demostrar el menor coste de los algoritmos de ordenación recursivos requiere técnicas más avanzadas de análisis de sucesiones de recurrencia, como el teorema maestro, o la fórmula de Akra-Bazzi [Cormen et al., 2001]. Este cambio de paradigma, de iterativo a geométrico, que viene de la mano de una extensión de las herramientas de análisis, se da también en el cálculo numérico, como muestran los ejemplos citados.

1.3. Análisis de algoritmos numéricos

Tras dar una descripción a grandes rasgos de los métodos para hallar raíces de polinomios en el apartado anterior, en este haremos una comparativa en cuanto al coste. Primero precisamos los conceptos relacionados con el coste de algoritmos numéricos (como precisión, condicionamiento y estabilidad). Luego comentamos brevemente el coste de los algoritmos más usados para hallar raíces, y posteriormente exponemos aplicaciones que se dan en la práctica para las que el rendimiento de los métodos actuales no es suficiente, y que motivan el desarrollo del presente trabajo.

Aproximación, estabilidad, condicionamiento

Algunos algoritmos numéricos calculan directamente el resultado deseado con un número finito de operaciones sobre los datos de entrada, como por ejemplo el método de eliminación de Gauss para calcular soluciones de sistemas de ecuaciones lineales. Otros algoritmos numéricos son aproximados, ya que obtienen aproximaciones al resultado, que serán más precisas cuantos más recursos se dediquen. Estos recursos son principalmente tiempo de ejecución del algoritmo. Hablando con propiedad, no se deberían denominar algoritmos ya que pueden no concluir en un número finito de pasos. Sin embargo es habitual llamar algoritmos a estos procedimientos, no finitos, que forman parte del análisis numérico y la teoría de la aproximación [Williamson and Shmoys, 2011]. Los problemas numéricos para los que sólo se conocen algoritmos aproximados son más abundantes que los problemas con algoritmo exacto conocido. Además, incluso para problemas con algoritmo exacto, suele ser preferible usar un algoritmo aproximado [Trefethen, 2010].

Para medir el coste de un algoritmo numérico aproximado no puede usarse el número de operaciones hasta llegar al resultado, pues la finalización puede no llegar. Lo que se usa es el número de operaciones realizadas para obtener una aproximación cuyo error sea menor que una tolerancia máxima preestablecida.

El error de la aproximación $\tilde{x}$ al valor x es $e = |x - \tilde{x}|$, la diferencia (en valor absoluto) entre la aproximación y el valor buscado. Desde luego este error es desconocido (como lo es el valor buscado), pero para cada algoritmo numérico aproximado puede hallarse una cota superior del error de la aproximación obtenida, en función de los recursos dedicados. Esta cota de error es una función decreciente

(a más recursos, menor error). Por ejemplo consideremos el problema de hallar un cero de una función continua en el intervalo $[a, b]$. Un algoritmo aproximado para resolverlo es la bisección sucesiva, cuyo recurso tiempo puede medirse en cantidad de bisecciones realizadas. El error e_n tras la bisección n-ésima verifica $e_n \leq \frac{b-a}{2^{n+1}}$. Otro algoritmo aproximado para este problema es el método de Newton (si converge en ese intervalo $[a, b]$). También puede medirse el tiempo dedicado en iteraciones, y el error e_n tras la iteración n-ésima es menor que $\frac{C}{2^{2^n}}$ para cierta constante C. La cota del error de Newton decrece más rápidamente que la de bisección. La cota de error es una medida de la efectividad del algoritmo: mientras más rápidamente decrezca, el algoritmo requerirá menos operaciones para bajar de un error máximo preestablecido.

En los dos ejemplos anteriores se ha descrito el *error de truncamiento*, que se produce al detener la ejecución del algoritmo, potencialmente infinita. Otra manera de limitar los recursos de un algoritmo es la discretización o muestreo, en la que no se usan todos los datos de entrada, sino sólo los suficientes para producir la aproximación. El *error de discretización* es el cometido en este caso, por ejemplo al aproximar una integral mediante una suma finita de Riemann, o mediante el método de los trapecios. Como se ha comentado, en el campo del análisis numérico es fundamental dar, como medida del coste de un algoritmo, una cota de su error de truncamiento en función del momento de detención, o de su error de discretización en función de la cantidad de datos usados.

Independientemente del error de truncamiento o de discretización, que viene de la limitación de recursos de ejecución del algoritmo, está el *error de redondeo*, que viene de la finitud de recursos usados para representar datos, normalmente datos numéricos. El redondeo es la conversión de un valor en un dato del tipo numérico que se va a usar para representarlo. Se usa un recurso finito, como es la memoria, para representar un conjunto de valores potencialmente infinito, como los enteros, o realmente infinitos (en cuanto a recursos requeridos para su expresión), como $1/3 = 0'\widehat{33}$, o $\sqrt{2}$. El tipo numérico más frecuente es el basado en coma flotante. Una introducción a este tipo de dato es [Goldberg, 1991a] y un estudio en profundidad [Higham, 2002]. Las implementaciones del tipo de dato coma flotante siguen el standard IEEE 754-2008, adoptado internacionalmente como IEC-60559, que permite diversos formatos, caracterizados por su precisión (número de bits). Los formatos más frecuentemente usados son precisión simple y doble. La precisión

de la mantisa es de 24 bits en simple, lo que quiere decir que puede representar valores con un error menor que $1/2^{24}$ (aproximadamente $7'2$ cifras decimales). Por otro lado el exponente tiene 8 bits, con lo que puede tomar valores entre -128 y 127 (aproximadamente 38 órdenes de magnitud decimales).

El standard especifica cinco modos de redondeo [IEEE, 2008]: hacia abajo, al valor mas cercano, etc. Vamos a usar como ejemplo una aritmética decimal, y así $1/3$ redondeado a una precisión $1/100 = 0'01$ de dos cifras decimales, por el modo de redondeo hacia abajo, sería $0'33$. El error de redondeo (la diferencia, en valor absoluto, entre un valor y su redondeo) es $0'00333...$. En general, se consigue error cero solo para ciertos valores. El error de redondeo es menor que la precisión (en el ejemplo, $0'0033... < 0'01$) y, dependiendo del modo de redondeo, puede ser igual. El error de redondeo interviene fundamentalmente en el redondeo de las operaciones en coma flotante (suma, producto, inverso): a partir de unos operandos con un cierto error, el resultado de una operación, con la misma precisión, va a tener en general peor error (es decir, mayor error).

Estas nociones se han recogido tradicionalmente, en teoría de medida y operaciones con magnitudes físicas [Taylor, 1997], considerando tres tipos de error: error de medida de datos (que en esta memoria no vamos a considerar), error de truncamiento (o discretización) y error de redondeo en las operaciones. Esta clasificación en tres tipos refleja la fuente de error: el error de medida viene de la sensibilidad finita de los dispositivos de entrada de datos, el error de truncamiento (o discretización) del finito tiempo disponible para cómputo, y el error de redondeo refleja la finita memoria disponible para representar datos.

Además de la anterior clasificación de las fuentes de error, también es habitual en el análisis numérico hablar de la velocidad (u orden) de convergencia [Ralston and Rabinowitz, 1978b]. Las aproximaciones pueden ponerse como una sucesión $\tilde{x}_n$, si el algoritmo es iterativo ($\tilde{x}_n$ es la aproximación en la iteración n-ésima), o si este procede por discretización ($\tilde{x}_n$ es la aproximación para una muestra de tamaño n). En cualquiera de los dos casos la cota de error, decreciente con n, asegura la existencia de un límite al que converge esa sucesión. Siendo e_n la sucesión de errores de $\tilde{x}_n$, se define el *orden de convergencia* como el exponente q tal que el límite:

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^q} = C$$

pertenece al intervalo $(0, \infty)$. Equivalentemente, si el error e_{n+1} es proporcional a e_n^q, con constante de proporcionalidad independiente de n. El orden de convergencia mide la velocidad a la que converge $\tilde{x}_n$. En el ejemplo anterior de cotas de error, se ve que el método de la bisección tiene orden $q = 1$, mientras que el método de Newton tiene $q = 2$. Se dice que $q = 1$ es un orden de convergencia lineal porque la representación gráfica de $\log(e_n)$ frente a n es una recta, mientras que para orden cuadrático, $q = 2$, esa representación es una parábola.

Como ejemplo, vamos a exponer un caso sencillo de algoritmo para el que vamos a calcular su error de truncamiento y su error de redondeo. (También nos referiremos a este ejemplo posteriormente al hablar de estabilidad y condicionamiento). Consideremos el problema "Dado u, hallar $u/3$", y un algoritmo iterativo que lo resuelve, consistente en aplicar repetidamente $x_n = \dfrac{u - x_{n-1}}{2}$. Evidentemente, tal algoritmo sólo será de interés práctico en un hipotético procesador en cuya aritmética la operación de dividir entre tres sea muy costosa comparada con la división entre dos. Calculamos por este algoritmo 1/3 y 10/3, ambos con condición inicial, por ejemplo, $x_0 = 5$ (en la tabla 1.1).

n	$x_n = \dfrac{1 - x_{n-1}}{2}$	n	$x_n = \dfrac{10 - x_{n-1}}{2}$
0	5.0000	0	5.0000
1	-2.0000	1	2.5000
2	1.5000	2	3.7500
3	-0.2500	3	3.1250
4	0.6250	4	3.4375
5	0.1875	5	3.2813
6	0.4063	6	3.3594
7	0.2969	7	3.3203
8	0.3516	8	3.3398
9	0.3242	9	3.3301
10	0.3379	10	3.3350

Tabla 1.1: Aproximaciones a 1/3 y 10/3

Habría que hacer un análisis para demostrar que este algoritmo converge (para ciertos valores iniciales), y para hallar su cota de error y orden de convergencia (necesaria para saber cuantas iteraciones hacen falta para alcanzar una precisión preestablecida). Por simplicidad no lo hacemos. También por simplicidad redon-

deamos a una precisión fija de cuatro cifras decimales. Al truncarlo en la iteración décima se tiene un error de truncamiento de $0'3379 - 1/3 = 0'0045\widehat{6}$ en el primer ejemplo y $3'3350 - 10/3 = 0'001\widehat{6}$ en el segundo. Por otro lado, para el error de redondeo, aunque prolongásemos el algoritmo todo lo posible, anulando el error de truncamiento, el error de redondeo al expresar el resultado (que es $0'0000\widehat{3}$) va a ser inevitable.

Para analizar algoritmos numéricos de aproximación llevados a cabo manualmente era suficiente estudiar el error de truncamiento (o el de discretización) para presupuestar el coste de ejecución del cálculo. El error de redondeo se evitaba aumentando la precisión de algunas operaciones intermedias del algoritmo, si se veía necesario cuando se estaba llevando a cabo. Con la llegada de los ordenadores, para ejecutar los algoritmos sin supervisión, es decir, como componentes de un sistema a los que solo se accede por su interfaz, es necesario estudiar también la *estabilidad* [Wilkinson, 1964]. Los errores de redondeo en las operaciones que realiza un algoritmo alteran la aproximación, y la cota de error de truncamiento, que supone que no hay errores de redondeo, quizá no es válida. Es decir, los errores de redondeo pueden acumularse en sucesivas iteraciones del algoritmo, impidiendo su convergencia. En tal caso se dice que el algoritmo es inestable. Por ejemplo, el algoritmo propuesto para hallar $u/3$, si redondeamos a dos cifras decimales, no converge a un valor fijo (Tabla 1.2). Pero se puede decir que es estable porque, aún con ese redondeo, oscila entre valores que no se apartan mucho del resultado deseado. Hay algoritmos para los que la oscilación causada por el redondeo no solo retrasa la convergencia, sino que hace sucesivas aproximaciones cada vez más alejadas del valor deseado. Es decir, el redondeo puede disminuir el orden de convergencia, o incluso impedirla.

Por otro lado, el *condicionamiento* es la dependencia del resultado del problema con respecto a los datos de entrada [Wilkinson, 1965]. Una pequeña variación del resultado ante pequeñas variaciones de la entrada significa un buen condicionamiento del problema, y en cambio fuertes variaciones del resultado se consideran un mal condicionamiento. No depende del algoritmo. El problema de hallar $u/3$ está bien condicionado. Un ejemplo de problema similar, pero mal condicionado, es hallar $1/(u-1)$. Este problema está mal condicionado alrededor de $u = 1$, donde un pequeño cambio en u puede producir un cambio muy grande en la respuesta, sea cual sea la precisión con que la representemos o el algoritmo que usemos para

n	$x_n = \frac{1 - x_{n-1}}{2}$ redondeado a dos decimales
0	5.00
1	2.00
2	1.50
3	-0.25
4	0.62
5	0.19
6	0.40
7	0.30
8	0.35
9	0.32
10	0.34
11	0.33
12	0.34

Tabla 1.2: Aproximaciones a 1/3 con precisión de dos dígitos

calcularla.

En resumen, el coste del algoritmo numérico se ve influenciado por el tamaño de entrada pero también por el error máximo que se tolera en la aproximación a la solución. Como se ve tras estas consideraciones, para lograr la respuesta con un cierto error máximo, tenemos que trabajar con una precisión que no solo depende de este error, sino también del condicionamiento del problema y de la estabilidad del algoritmo que usemos. Esto se aparta tanto del análisis de algoritmos clásico como de la teoría de medida de magnitudes [Trefethen and Bau III, 1997]. Un estudio con ejemplos de algoritmos numéricos, con su coste, puede verse en [Emiris et al., 2010].

Coste de algoritmos para raíces

Los algoritmos para hallar raíces de polinomios, al ser de aproximación, están sujetos a estas interrelaciones error máximo demandado-condicionamiento por un lado, precisión de trabajo-estabilidad por otro. Sin embargo, el tratamiento teórico clásico se ha centrado en el estudio de la cota de error (y la correspondiente velocidad de convergencia). Siguiendo las referencias estándar de cálculo numérico (ver [Ralston and Rabinowitz, 1978b] en general o [Henrici, 1988] en particular para

raíces), se pueden comparar los métodos iterativos (MI) y los métodos geométricos (MG) según los siguientes criterios:

- Cotas de error (y velocidad de convergencia): Los MG suelen tener una convergencia lineal (orden $q = 1$), mientras que los MI tienen convergencia más rápida, de orden $q > 1$. Esto es un punto fuerte de los MI, como puede verse en el número de bits correctos de la aproximación n-ésima. Este es el número de posiciones iniciales en la expresión binaria de la aproximación que coinciden con la expresión binaria de la solución. En el ejemplo anterior del método de la bisección, de convergencia lineal, e_{n+1} es la mitad de e_n, es decir en cada paso la aproximación incrementa en uno el número de sus bits correctos. En cambio el método de Newton, de convergencia cuadrática (orden $q = 2$), e_{n+1} es aproximadamente e_n^2, es decir el número de bits correctos se dobla en cada paso.

- Convergencia local o global: Normalmente, para que se verifique el decrecimiento de la cota de error de un MI, es necesario que este se inicie con una estimación previa bastante cercana a la solución. Se dice que converge localmente, o en un entorno de la solución. Los MG, en cambio, suelen tener convergencia global: el error decrece aunque la estimación inicial esté alejada de la solución. Este es el punto fuerte de los MG.

- Convergencia condicional: Los MI suelen tener un conjunto de entradas para los que no convergen. Por ejemplo el método de Newton no es válido para polinomios con raíces múltiples. Esta es otra restricción para la convergencia de los MI (además de la localidad). Los MG en cambio convergen incondicionalmente.

En resumen, los MI suelen ser locales, condicionales, de convergencia de orden 2 o más. Los MG son globales, incondicionalmente convergentes, y de convergencia lineal.

A pesar del fundamento teórico de los MI (por ejemplo para el método de Newton, la teoría clásica de Kantorovich [Ralston and Rabinowitz, 1978a], o la teoría α de Smale [Blum et al., 1998]), la condicionalidad de la convergencia hace que, en general, estos métodos no se consideren completos (es decir, válidos en todos los casos). No se conocían algoritmos completos, es decir, de convergencia

no condicional hasta 1924, con los trabajos de Brouwer y Weyl [Pan, 1997] sobre MG. Para MI, tampoco se ha precisado la globalidad de la convergencia (es decir, que no se necesiten estimaciones previas). Aportes al estudio de la convergencia global han sido los de [Smale, 1981] en el método de Newton, [Pan, 1987], [Toh and Trefethen, 1994] para el método de la matriz compañera, o [Dickenstein and Emiris, 2005] para los métodos de prolongación homotópica.

Al estudiar los algoritmos disponibles, el interés de los investigadores se ha centrado en los puntos anteriores, sobre todo la cota de error, y en cambio el coste (desde el punto de vista de los recursos necesarios) no se ha estimado adecuadamente, como afirma [Pan, 1997]. Intentos de paliar esa situación son los trabajos sobre el "teorema fundamental del álgebra"(que es como se conoce al problema de hallar raíces de polinomios) de [Smale, 1981] y [Schönhage, 1982] para MI y MG respectivamente. Como se ha comentado, analizar la complejidad computacional de este problema ha sido una motivación para el desarrollo de los MG [Renegar, 1987]. Estos estudios son clásicos en el sentido de que no suelen considerar la estabilidad numérica de los métodos y el condicionamiento del problema. La estabilidad de la evaluación de polinomios en coma flotante se analiza en [Toh and Trefethen, 1994]. Ante la dificultad de análisis se recurre a comparativas experimentales [Jenkins and Traub, 1975], como por ejemplo [Bini and Fiorentino, 2000] o [Zeng, 2004].

Desde el punto de vista práctico, la localidad de la convergencia es una importante desventaja de los MI: la posición de la raíz encontrada no esta relacionada con la estimación inicial [Blum et al., 1998]. Esto puede ser representado en una gráfica de cuencas de atracción. La *cuenca de atracción* de una raíz (para el método de Newton) es el conjunto de valores complejos z tales que el método converge a esa raíz, si la iteración comienza con el valor inicial z. Se dice que una raíz *atrae* los puntos que convergen hacia ella. Las cuencas de dos raíces distintas son subconjuntos disjuntos del plano. Representamos las cuencas de las raíces de un polinomio de grado 31 en la figura 1.5. Es la región plana $|\mathrm{Re}(z)| < 1'2, |\mathrm{Im}(z)| < 1'2$. Las raíces están señaladas con asteriscos. El método de Newton se aplica una vez para cada píxel de la figura. La estimación inicial es el valor complejo correspondiente a la posición de este píxel. Luego cada píxel es coloreado con un nivel de gris, según la raíz a la que converge. Así cada cuenca queda coloreada con un nivel de gris diferente.

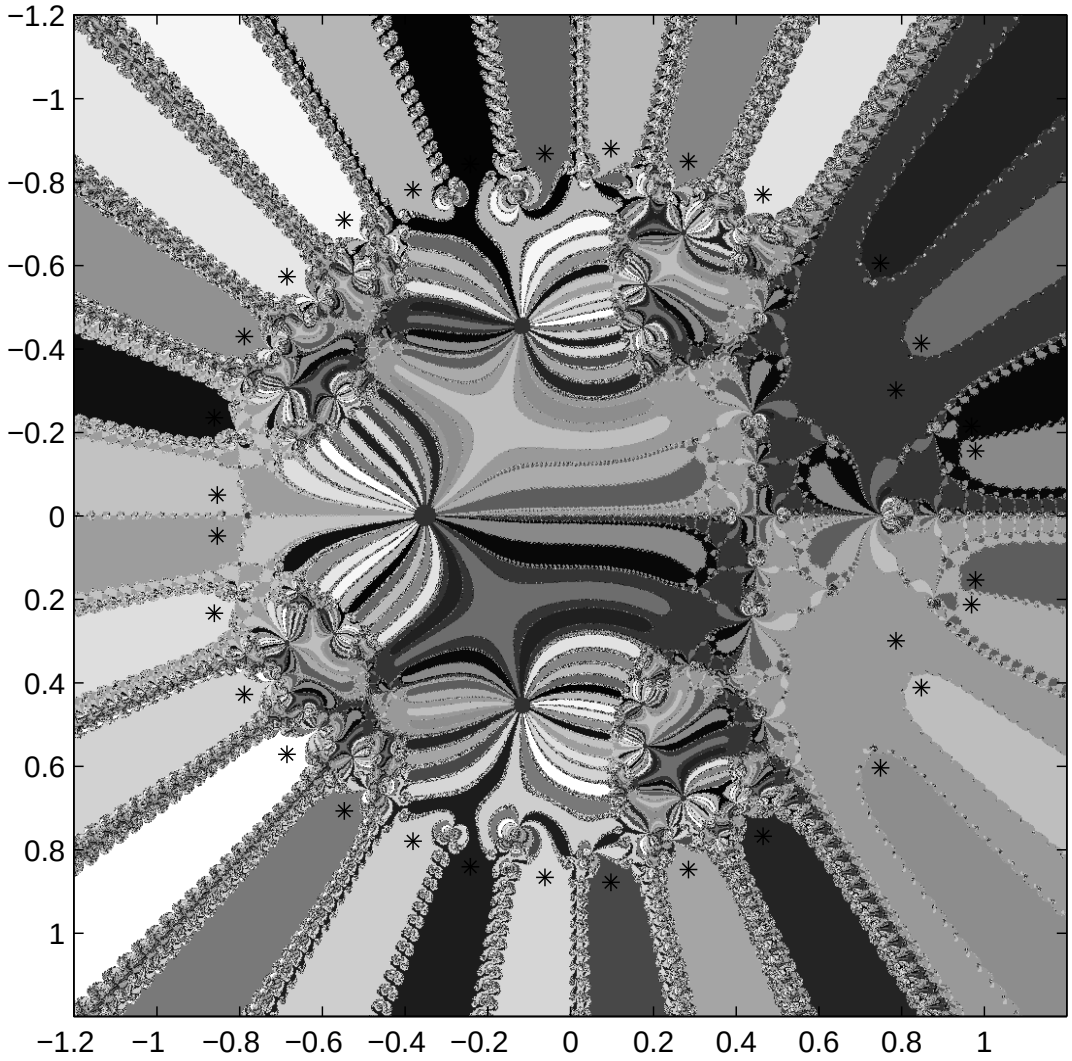


Figura 1.5: Cuencas de atracción para el método de Newton aplicado a un polinomio de grado 31. Los asteriscos marcan las raíces, y los niveles de gris las cuencas.

En la figura, las fronteras entre cuencas muestran puntos cuyo color es diferente del de sus puntos próximos. Un punto tal converge a una raíz situada lejos de las raíces correspondientes a puntos cercanos. Además, las fronteras están imbricadas de un modo fractal: en cualquier línea que una dos puntos de distintas cuencas va a haber puntos pertenecientes a una tercera cuenca. Así, la raíz hacia la que un punto converge es imprevisible. Esta indeterminación de la convergencia es característica

de los MI. Es un fenómeno de dependencia sensible de condiciones iniciales, y no puede ser evitado aumentando la precisión numérica [Kalantari, 2009].

En este contexto, para encontrar, por ejemplo, las raíces cerca de la circunferencia unidad con un MI de uso común, es necesario hallar todas las raíces, y seleccionar las que sean de interés. Para evitar hallar repetidamente la misma raíz, cada una encontrada se elimina del polinomio dividiendo por un factor lineal (procedimiento llamado deflación). Los pasos de deflación requieren aritmética de alta precisión, que es computacionalmente costosa [Ralston and Rabinowitz, 1978a]. En contraste, el uso de pasos de deflación (y por tanto de alta precisión) no es necesario en MG. Además, si el área de interés es relativamente reducida, y contiene pocas raíces, un MG evita gastar cálculo en raíces no deseadas. Muchas aplicaciones prácticas se beneficiarían métodos que permitan centrar la búsqueda de raíces en una región pre-especificada del plano complejo (véase [Sitton et al., 2003, Van Dooren, 1994] y sus referencias). Esta capacidad de los MG, y el ahorro computacional consiguiente con respecto a los MI, es una motivación práctica para su estudio [Pan, 2012].

Con respecto a las demandas actuales, la linealización de los modelos (es decir, reemplazar cada operación o relación por una aproximación mediante matrices) suele servir para sortear la necesidad de hallar raíces de polinomios de grado mayor que 10 [Pan, 1997]. Sin embargo, hay ámbitos donde se requiere el cálculo de raíces de polinomios de alto grado (del orden de varios cientos o miles). Pueden verse referencias sobre la necesidad de hallar raíces de polinomios de alto grado en proceso de señal [Sitton et al., 2003] o en robótica (el problema cinemático inverso [Craig, 2005], [Sommese and Wampler, 2005]) o teoría de antenas [Orchard et al., 1985]. También en las referencias de [Bini and Pan, 1994] y [Huang, 2004]. Con respecto a los sistemas de álgebra computacional, o álgebra por ordenador (*Computer algebra systems, CAS* [Von Zur Gathen and Gerhard, 2013]), surge la necesidad de hallar raíces al resolver sistemas de polinomios mediante el método de eliminación, teóricamente óptimo para esta tarea. Como no se dispone de un algoritmo fiable para raíces, en la práctica se usan las bases de Grobner y el politopo de Newton, que no son métodos óptimos para sistemas de polinomios. Otros problemas algebraicos que se beneficiarían de un algoritmo fiable para raíces son la factorización de polinomios en coeficientes racionales (que se relaciona con la aproximación diofántica), y el cálculo del máximo común divisor (referencias

en [Pan, 1997] y [Dickenstein and Emiris, 2005]).

En nuestro trabajo [García Zapata et al., 2004b] nos enfrentamos a una situación (con polinomios de grado en torno a varias decenas) en la que un método geométrico es mejor que uno iterativo. La ventaja se deriva principalmente de que en muchos casos no se necesita encontrar todas y cada una de las raíces de un polinomio, sino sólo aquellas que satisfacen ciertas condiciones. En aplicaciones de procesado de señales, como se describe en la sección anterior, el análisis LPC produce una función de transferencia $H(z) = \frac{1}{A(z)}$, el modelo de la señal. Las raíces del polinomio denominador $A(z)$ están dentro del círculo unidad, por la estabilidad del modelo, y las raíces más cercanas a la circunferencia están relacionados con los componentes de frecuencia principales de la señal [Oppenheim et al., 1996]. Por ejemplo, el polinomio de la figura 1.5 proviene del análisis LPC de una señal de voz. Estamos interesados en encontrar las raíces complejas de polinomios que estén situadas cerca de la circunferencia de la unidad, hasta una distancia que depende del grado de modelado LPC. Esto nos ha llevado a considerar MG para resolver este problema. En el campo del procesamiento de señal se usan varios métodos iterativos para encontrar las raíces de mayor módulo, como Graeffe o Bernoulli [Ralston and Rabinowitz, 1978a]. Sin embargo, el alto grado de los polinomios de interés para nosotros, y la compleja casuística que requieren estos métodos, son dos características que dificultan su aplicación.

Es el objetivo de esta memoria precisar las condiciones bajo las cuales puede aplicarse eficientemente un procedimiento de extracción de raíces, capaz de restringirse a una zona de interés en el plano complejo, aplicable en el ámbito de la codificación lineal predictiva para la detección y seguimiento de formantes. En el capítulo 2 introducimos un método para calcular el índice, el número de vueltas que da una curva plana alrededor del origen, y en el capítulo 3 se usa el índice como criterio de la presencia de raíces de un polinomio en una región del plano. También se desarrolla un método recursivo de subdivisión de regiones usando este criterio para hallar las raíces. Finalmente en el capítulo 4 se realiza una comparativa del método propuesto con otros actualmente usados en la práctica para tareas similares de proceso de señal.

Capítulo 2

Desarrollo y estudio teórico de un método geométrico para calcular el índice de una curva plana

2.1. El índice de una curva y el cálculo de raíces

El índice de una curva plana cerrada Δ es el número de vueltas que da alrededor del origen. Su valor puede ser calculado aplicando la formula integral de Cauchy (un resultado de Análisis Complejo) como:

$$\mathrm{Ind}(\Delta) = \frac{1}{2\pi i} \oint_{\Delta} \frac{dz}{z}$$

El índice se ha usado en varios procedimientos para calcular las raíces de un polinomio f, basándose en el hecho de que el número de raíces contenidas en una región del plano complejo bordeada por una curva Γ coincide con el índice de la curva $\Delta{=}f(\Gamma)$. Si este índice es positivo, la región contiene una o varias raíces. Si se divide esta región en regiones más pequeñas, y se calcula el índice de la transformación por f del borde de cada una de ellas, se tiene una localización más precisa de las raíces. Aplicando esta subdivisión recursivamente a las regiones que contengan alguna raíz, puede alcanzarse una aproximación a las raíces de f con la precisión que se quiera [Henrici, 1988]. Un procedimiento similar se ha propuesto para hallar los ceros de funciones analíticas [Kravanja and Van Barel, 2000].

En los trabajos de [Ying and Katz, 1988] o en [Ko et al., 2008] se encuentran enunciados precisos sobre las condiciones en que podemos usar el índice en algoritmos de cálculo de raíces, y sugerencias sobre como gestionar los casos singulares. Pero el estudio de la complejidad de estos métodos, esencial para compararlos con los tradicionales, no se ha llevado a cabo. Nos proponemos llenar ese hueco.

Con este propósito, mostraremos que la distancia de la curva Δ al origen de coordenadas es una medida del coste computacional del cálculo del índice. Llamamos a esta distancia el *valor de singularidad* de Δ. En particular, en el uso del índice para calcular las raíces de un polinomio f, el valor de singularidad de $\Delta = f(\Gamma)$ es equivalente a la distancia desde Γ a las raíces de f. Las referencias mencionadas comentan la influencia de esta distancia en el coste, pero carecen de una expresión precisa de su dependencia, como la que damos más adelante. Además, damos cotas del coste del manejo de casos singulares, que surgen cuando se aplica el algoritmo de cálculo del índice como subprocedimiento recursivo en un algoritmo para cálculo de raíces.

El valor de singularidad de Δ es el factor principal en el coste del algoritmo. El recíproco de este valor puede compararse con el número de condición de una matriz, en cálculo numérico lineal. Ambos son una medida del mal condicionamiento de los datos de entrada. Por ejemplo, el coste de una inversión matricial con una precisión predefinida crece proporcionalmente al número de condición de la matriz [Golub and Van Loan, 1996]. De modo similar, el coste de un cálculo de índice crece con el recíproco del valor de singularidad.

Más allá de su interés teórico, es necesario un estudio del valor de singularidad para el uso del índice en aplicaciones de cálculo de raíces, porque surgen contornos singulares en la subdivisión recursiva de la región plana. En el tratamiento de estos casos reemplazamos el contorno singular por otro, situado a corta distancia, con mejor valor de singularidad. Esta técnica de desvío es similar a la propuesta en [Ying and Katz, 1988], pero provistos con nuestros resultados sobre el valor de singularidad podemos aplicarlo en el seno de un procedimiento recursivo, y por tanto podemos tratar curvas con una singularidad arbitraria, como aparece en [García Zapata and Díaz Martín, 2014] y se describe en el capítulo 3.

En la siguiente sección exponemos el algoritmo para cálculo del índice llamado Procedimiento de Inserción (PI, figura 2.6). Luego estudiamos su coste computacional, y lo expresamos en función del valor de singularidad de la curva en la

hipótesis de que esta está uniformemente parametrizada, y en la siguiente sección desarrollamos un estudio similar para curvas Lipschitzianas, más generales. En la sección 2.5 modificamos ese algoritmo base y enunciamos una condición necesaria para que el nuevo algoritmo cuente los giros de cualquier tamaño alrededor del origen (Procedimiento de Inserción Válido para cualquier secuencia inicial, PIV, figura 2.13). La sección 2.6 introduce un control, con el propósito de que el algoritmo resultante pueda aplicarse a curvas con un valor de singularidad arbitrario desconocido a priori (Procedimiento de Inserción con Control de Singularidad, PICS, figura 2.17).

2.2. Definiciones y procedimiento de inserción

Vamos a considerar curvas cerradas definidas paramétricamente, es decir, como aplicaciones de un intervalo real al plano complejo $\Delta : [a, b] \to \mathbb{C}$ con $\Delta(a) = \Delta(b)$. Como es usual en el estudio de curvas parametrizadas, dos curvas $\Delta_1 : [a, b] \to \mathbb{C}$ y $\Delta_2 : [c, d] \to \mathbb{C}$ con la misma imagen $\mathrm{Im}(\Delta_1) = \mathrm{Im}(\Delta_2)$ pueden verse como dos parametrizaciones de un mismo subconjunto de $\mathbb{C}$. Recuérdese que una curva $\Delta : [a, b] \to \mathbb{C}$ es *uniformemente parametrizada* si la longitud de un arco de la curva coincide con la longitud del intervalo de sus valores de parámetro, es decir, si para cada $x, y \in [a, b]$, se tiene que $|y - x| = \int_x^y d\Delta(t)$. Esta última integral es la longitud de arco $\mathrm{longarc}(\Delta([x, y]))$. Toda curva diferenciable a trozos puede parametrizarse uniformemente [Kolmogorov and Fomın, 1975]. Esto es, para cualquier curva $\Delta : [a, b] \to \mathbb{C}$ hay otra curva $\Delta^u : [0, \mathrm{longarc}(\Delta)] \to \mathbb{C}$ con $\mathrm{Im}(\Delta) = \mathrm{Im}(\Delta^u)$ y Δ^u uniformemente parametrizada.

También consideraremos *curvas Lipschitzianas* (esto es, verificando que hay una constante L con $|\Delta(y) - \Delta(x)| \leq L|y - x|$ para cada $x, y \in [a, b]$). Las curvas uniformemente parametrizadas son un caso particular de las Lipschitzianas, con $L = 1$.

El *índice*, o *número de vueltas al origen*, $\mathrm{Ind}(\Delta)$ de una curva $\Delta : [a, b] \to \mathbb{C}$, es el número de rotaciones completas que da la curva alrededor del punto $(0, 0)$ en sentido contrario a las agujas del reloj. Véase la figura 2.1.

Como caso particular de la fórmula de Cauchy de análisis complejo, el índice

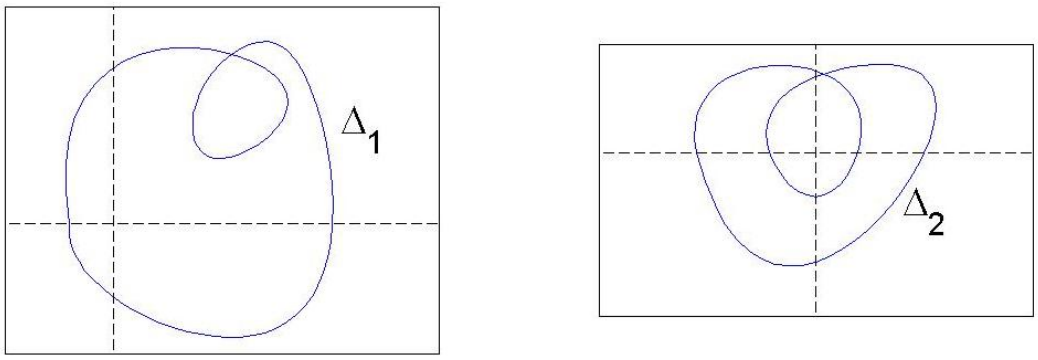


Figura 2.1: Los índices de las curvas Δ_1 y Δ_2 son $\mathrm{Ind}(\Delta_1) = 1$ y $\mathrm{Ind}(\Delta_2) = 2$.

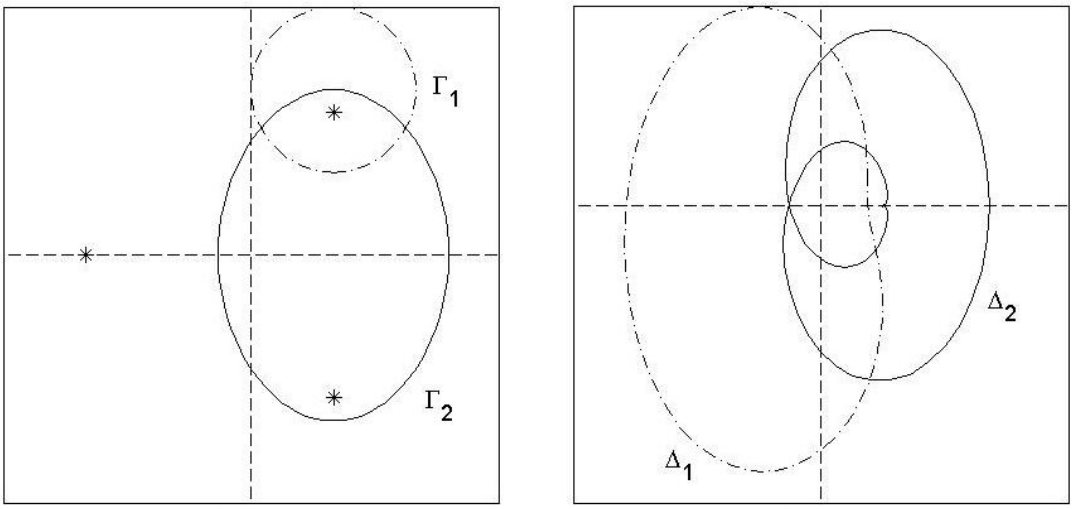
es igual a la siguiente integral de línea:

$$\mathrm{Ind}(\Delta) = \frac{1}{2\pi i} \int_{\Delta} \frac{1}{w}\, dw$$

El concepto de índice se aplica en el principio del argumento [Henrici, 1988], que afirma que el número de ceros (contados con multiplicidad) de una función analítica $f : \mathbb{C} \to \mathbb{C}$, $w = f(z)$, dentro de una región con borde definido por la curva Γ, es igual al índice de la curva $\Delta = f(\Gamma)$ (ver figura 2.2). El principio del argumento puede ser visto como un análogo bidimensional del teorema de Bolzano, y es la base varios métodos recursivos para hallar los ceros de funciones holomorfas y, en particular, ceros de polinomios.

Notemos que el índice de la curva Δ no está definido si Δ pasa por el origen $(0, 0)$ ya que la integral $\int_{\Delta} \frac{1}{w}\, dw$ no existe. En este caso, Δ se dice que es una *curva singular*. Si $\Delta = f(\Gamma)$, esto es equivalente a que Γ pase por un cero de f. Dado un valor real $\varepsilon \geq 0$, decimos que una curva es *ε-singular* si su distancia mínima al origen es ε. Las curvas 0-singulares son las que se han llamado previamente singulares para la integral del índice. La fórmula de Cauchy no es aplicable a curvas 0-singulares.

Como se ha comentado en la introducción, adoptamos el enfoque de Henrici [Henrici, 1988] como alternativa frente a la integración numérica de la integral del índice. Por tanto, trabajaremos con aproximaciones poligonales de la curva Δ, esto es, conjuntos discretos de puntos complejos dispuestos en un cierto orden: Para cualquier secuencia de valores del parámetro $s_i \in [a, b]$, $(a = s_0, \ldots, s_n = b)$ con



Plano z Plano w

Figura 2.2: El números de raíces del polinomio $f(z) = z^3 + 1$ dentro de Γ_1 y Γ_2 es igual a los índices de $\Delta_1 = f(\Gamma_1)$ y $\Delta_2 = f(\Gamma_2)$, respectivamente.

$s_0 < s_1 < \cdots < s_n$, la aproximación poligonal es $\widetilde{\Delta}_n = (\Delta(s_0), \ldots, \Delta(s_n))$. La figura 2.3 muestra una curva con una de sus aproximaciones poligonales.

Dada una secuencia $S = (s_0, \ldots, s_n)$ de valores crecientes, $s_i < s_{i+1}$, $i = 0, \ldots, n-1$, llamamos su *paso máximo* $|S|$ a la máxima diferencia entre valores consecutivos, esto es: $|S| = \max\limits_{0 \le i \le n-1}(s_{i+1} - s_i)$.

El plano complejo se divide en sectores angulares, de ángulo $\dfrac{\pi}{4}$. Hay ocho de tales sectores, llamados $C_0, C_1, C_2, \ldots, C_7$, cada uno la mitad que resulta de cortar un cuadrante por su bisectriz. Para ser preciso, cada borde entre sectores adyacentes, C_x y C_{x+1}, está incluido en C_{x+1} para $x = 0, \ldots, 6$, y el borde entre C_7 y C_0 está incluido en C_0.

Decimos que dos puntos p, q de la curva Δ están *conectados* si están situados en dos sectores adyacentes o en el mismo sector, esto es, si $p, q \in C_x \cup C_{x+1}$ para algún x. Equivalentemente, si cuando $p \in C_x$ y $q \in C_y$ entonces $y = x \pm 1$ (o $y = x$). Como los sectores C_7 y C_0 son adyacentes, las igualdades deben entenderse como una congruencia aritmética módulo 8.

Consideremos los bordes que delimitan los sectores $C_0, \ldots, C_7$. Para cada segmento de curva $\Delta([s_i, s_{i+1}])$, decimos que *cruza N bordes* si el intervalo de parámetros contiene como mucho N valores f_j, $j = 1, \ldots, N$, con $s_i \le f_1 < f_2 < \ldots <$

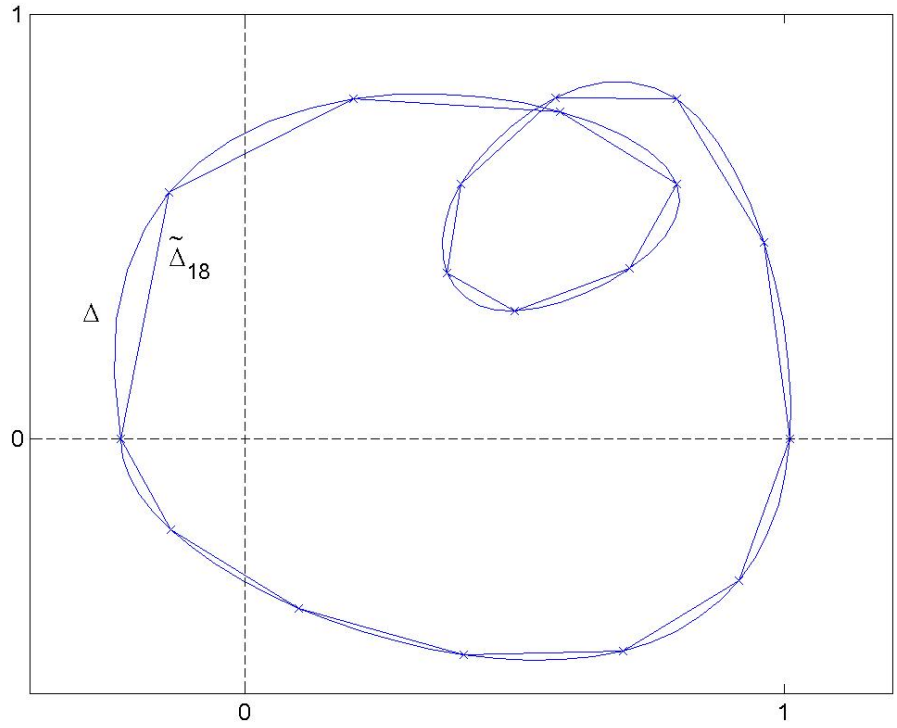


Figura 2.3: La curva Δ es aproximada por la poligonal $\widetilde{\Delta}_{18}$ con una resolución de 18 puntos.

$f_N \leq s_{i+1}$, cuyas imágenes $\Delta(f_j)$ pertenezcan a algún borde. Por ejemplo en la figura 2.4 hay varios segmentos con el número de bordes que cruza cada uno.

Supongamos que la curva Δ tiene un índice definido, esto es, que no cruza el origen. Entonces es seguro que hay una secuencia $(t_0, t_1, \ldots, t_m)$ de valores del parámetro $t \in [a, b]$, $a = t_0 < t_1 < \ldots < t_m = b$ cuyas imágenes por la aplicación Δ están conectadas, como muestra la figura 2.5.

Decimos que un polígono de vértices $\Delta(t_i)$, $i = 0, 1, \ldots, m$ *satisface la propiedad de conexión* si cada par de puntos consecutivos $\Delta(t_i)$ y $\Delta(t_{i+1})$ están conectados. Si conseguimos por algún método una secuencia $(\Delta(t_0), \Delta(t_1), \ldots, \Delta(t_m))$ de puntos definiendo un polígono $\widetilde{\Delta}_m$ que verifica la propiedad de conexión, su índice $\mathrm{Ind}(\widetilde{\Delta}_m)$ es igual al número de puntos $\Delta(t_i)$ en C_7 que están seguidos por un punto $\Delta(t_{i+1})$ en C_0. La ocurrencia de un sector C_0 seguido por C_7 debe contarse negativamente. Esto es, $\mathrm{Ind}(\widetilde{\Delta}_m) = \#(\text{cruces de } C_7 \text{ a } C_0) - \#(\text{cruces de } C_0 \text{ a } C_7)$. Este es el método de Henrici para el cálculo del índice.

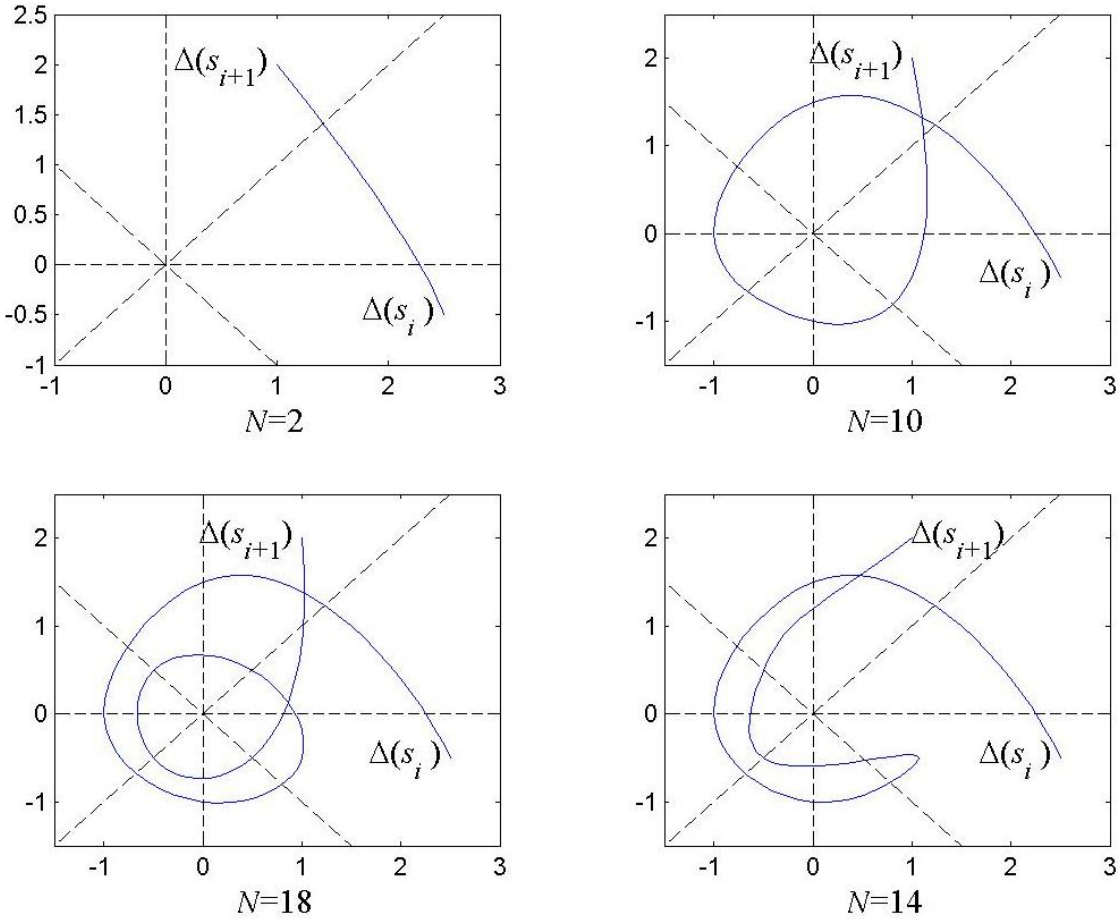


Figura 2.4: Segmentos de curva con el número N de bordes que cruzan.

Henrici no especificó el procedimiento para encontrar los valores del parámetro $(t_0, t_1, \ldots, t_m)$, ni precisó las condiciones bajo las cuales el índice del polígono de aproximación $\widetilde{\Delta}_m$ coincide con el de la curva original Δ. En el trabajo [Ying and Katz, 1988] se propone tal procedimiento, con un coste computacional razonable. Consiste en construir la secuencia buscada a partir de una secuencia inicial de valores del parámetro, $(a = s_0, \ldots, s_n = b)$, de la curva Δ, cuyas imágenes no verifican necesariamente la propiedad de conexión, esto es, que quizás, para algún i, las imágenes de s_i y s_{i+1} no están conectadas. Se recorre la secuencia de valores $(\ldots, s_i, \ldots)$ desde su inicio s_0, hasta que se encuentra un par (s_i, s_{i+1}) de valores consecutivos cuyas imágenes $\Delta(s_i) \in C_x$ y $\Delta(s_{i+1}) \in C_y$ no estén conectadas. En esta situación se inserta un valor de interpolación $\dfrac{s_i + s_{i+1}}{2}$ en la secuencia de parámetros $(s_0, \ldots, s_n)$ entre s_i y s_{i+1}. Después se recorre de nuevo la secuencia

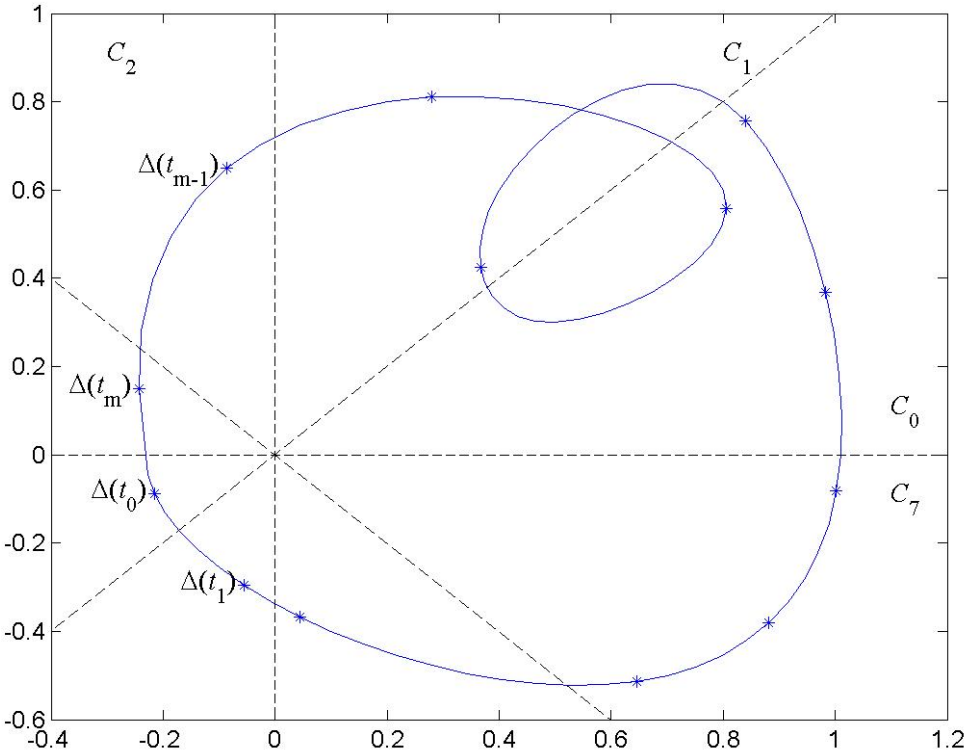


Figura 2.5: Las imágenes de los valores sucesivos t_i, t_{i+1} están conectadas.

resultante $(s'_0, \ldots, s'_{n+1})$ hasta que se encuentre otro par (s'_j, s'_{j+1}) cuyas imágenes no estén conectadas, y otra vez se inserta un punto intermedio como se ha descrito. Iterando este proceso, se llega finalmente a una secuencia $T = (t_0, ..., t_m)$, $m \geq n$, cuyas imágenes verifican la propiedad de la conexión. Este procedimiento está definido en la figura 2.6.

El Procedimiento de Inserción recorre la secuencia de izquierda a derecha, de modo que los puntos necesarios para conectar s_i y s_{i+1} se insertan antes que los que van entre s_{i+1} y s_{i+2}.

Este procedimiento de inserción de puntos de interpolación presenta dos inconvenientes que dificultan su aplicación práctica en el cálculo del índice. En primer lugar, no acaba en un número finito de pasos en ciertos casos. Discutimos esto en las secciones 2.3 y 2.4. En segundo lugar, $\mathrm{Ind}(\widetilde{\Delta}_m)$ coincide con el índice de Δ solo bajo ciertas suposiciones. La sección 2.5 trata esta cuestión de la coincidencia de

Procedimiento de Inserción: Para hallar el índice de una curva $\Delta : [a, b] \to \mathbb{C}$

Parámetros de entrada: La definición analítica de la curva Δ, y una secuencia $S^{(0)} = (s_0, \ldots, s_n)$, de valores crecientes $s_0 < s_1 < \ldots < s_m$, de $[a, b]$ con $a = s_0$ y $b = s_n$.

Salida: Una secuencia $(t_0, \ldots, t_m)$ cuyas imágenes verifican la propiedad de la conexión.

Método:

Asignar a k el valor 0. (k es un contador de iteraciones);

Mientras $S^{(k)} = (s'_0, \ldots, s'_i, s'_{i+1}, \ldots, s'_{n+k})$ tenga dos valores consecutivos cuyas imágenes $\Delta(s'_i)$ y $\Delta(s'_{i+1})$ no estén conectadas (encontrados recorriendo la secuencia de izquierda a derecha) hacer:

{ Insertar $\dfrac{s'_i + s'_{i+1}}{2}$ entre s'_i y s'_{i+1}; esto es,

$$S^{(k+1)} = \left(s'_0, \ldots, s'_i, \frac{s'_i + s'_{i+1}}{2}, s'_{i+1}, \ldots, s'_{n+k}\right) ;$$

Incrementar k;

}

Retornar la secuencia resultante.

Figura 2.6: Procedimiento de inserción (Ying y Katz).

$\mathrm{Ind}(\widetilde{\Delta}_m)$ y $\mathrm{Ind}(\Delta)$. En la sección final de este capítulo modificamos el procedimiento de inserción para producir un algoritmo, finito y analizable, para el cálculo del índice.

2.3. Coste del procedimiento de inserción para curvas uniformemente parametrizadas

Si la curva Δ cruza el origen (es decir, Δ es una curva singular) el procedimiento de inserción no puede acabar, porque no hay ninguna secuencia de puntos con la propiedad de conexión. Además, aunque no sea singular, si la curva pasa lo suficientemente cerca del origen, el número de inserciones puede ser arbitrariamente alto. El propósito de este apartado es dar un enunciado preciso de este hecho, en el teorema 1 más adelante.

Supongamos que se aplica el procedimiento de inserción con secuencia inicial

$S^{(0)} = (s_0, \ldots, s_n)$. Concentrémonos en un intervalo $[s_i, s_{i+1}]$ cuya imagen, el segmento de curva $\Delta([s_i, s_{i+1}])$, tenga sus extremos no conectados (ver la figura 2.7). Denotaremos con C_x al sector que contiene $\Delta(s_i)$, y p_k al punto resultante de la inserción k-ésima del procedimiento entre $\Delta(s_i)$ y $\Delta(s_{i+1})$, $k = 1, 2, \ldots$. También denotaremos u_k al valor del parámetro tal que $p_k = \Delta(u_k)$, así que $S^{(k)} = (s_0, \ldots, s_i, \ldots, u_k, \ldots, s_{i+1}, \ldots, s_n)$.

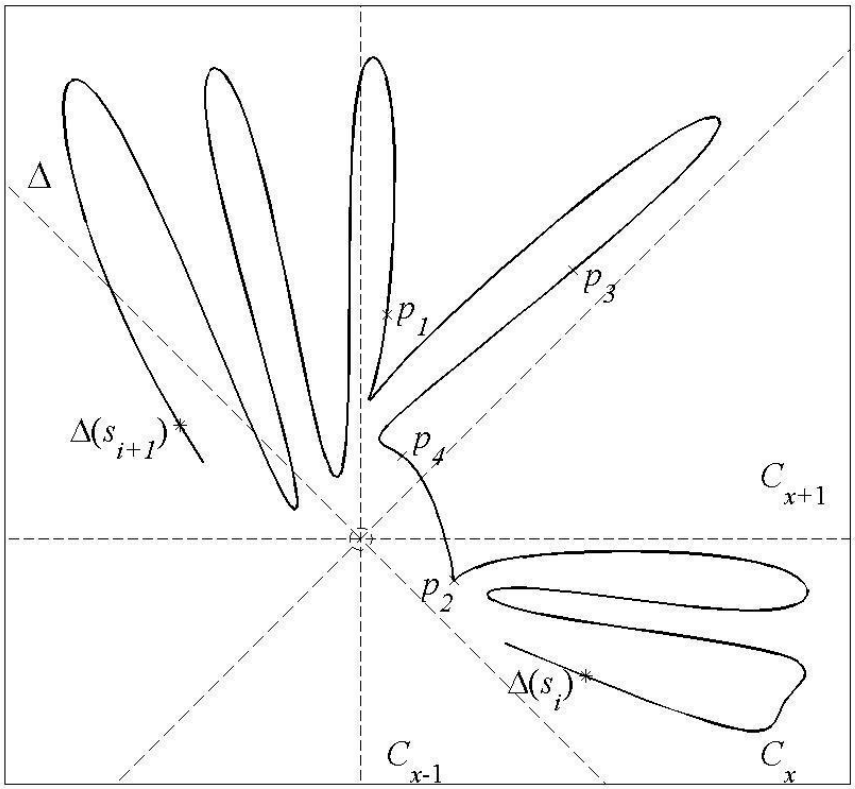


Figura 2.7: La secuencia de parámetros producida por el procedimiento de inserción en esta curva es $S^{(4)} = (\ldots, s_i, u_2, u_4, u_3, u_1, s_{i+1}, \ldots)$. El siguiente punto de inserción p_5 va entre p_2 y p_4.

Notemos que las inserciones u_k en la secuencia de parámetros no se hacen necesariamente en orden creciente, es decir, no necesariamente $u_k < u_{k+1}$. Para gestionar esta impredicibilidad del punto de inserción, la secuencia de puntos se denotará $\Delta(S^{(k)}) = (\ldots, \Delta(s_i), \ldots, p_k, \ldots, \Delta(s_{i+1}), \ldots)$, y llamaremos q_1 y q_2 a los primeros puntos no conectados encontrados en esta secuencia $\Delta(S^{(k)}) = (\ldots, \Delta(s_i), \ldots, q_1, q_2, \ldots, \Delta(s_{i+1}), \ldots)$ en el escaneo izquierda-derecha que hace el procedimiento de inserción. A pesar de esta expresión, es posible que $q_1 = \Delta(s_i)$

o $q_2 = \Delta(s_{i+1})$, o que q_1 o q_2 sean iguales a p_k. Supongamos que ningún punto de la secuencia $\Delta(S^{(k)})$ entre $\Delta(s_i)$ y $\Delta(s_{i+1})$ pertenece a C_{x-1} ni a C_{x+1}, los sectores adyacentes al que contiene $\Delta(s_i)$. En tal caso tenemos que $q_1 \in C_x$ y $q_2 \in (C_{x-1} \cup C_x \cup C_{x+1})^c$. Esto es porque el procedimiento de inserción recorre la secuencia en orden creciente del valor del parámetro, y por tanto los puntos del segmento $(\Delta(s_i), \ldots, q_1)$ de $\Delta(S^{(k)})$ deben pertenecer todos a C_x (ya que están conectados, $\Delta(s_i) \in C_x$, y no hay puntos en C_{x-1} ni en C_{x+1}). Y por tanto el punto no adyacente q_2 debe pertenecer a un sector no adyacente, esto es, $q_2 \in (C_{x-1} \cup C_x \cup C_{x+1})^c$.

La siguiente proposición formaliza este razonamiento, llamando I_k al segmento de curva que une p_k con el punto previo de la secuencia, y I'_k al segmento de curva que une p_k con el punto siguiente. Se define $I_0 = I'_0 = \Delta([s_i, s_{i+1}])$. Por ejemplo, como $p_1 = \Delta(u_1)$, tenemos que $I_1 = \Delta([s_i, u_1])$ y $I'_1 = \Delta([u_1, s_{i+1}])$ (ver figura 2.8).

Como notación, en la siguiente proposición el conjunto $\{p_1, p_2, \ldots, p_{k-1}\}$ de puntos debe entenderse como $\{p_1, p_2\}$ cuando $k = 3$, $\{p_1\}$ cuando $k = 2$ y el conjunto vacío cuando $k = 1$.

Proposición 1. *Supongamos que $\Delta(s_i)$ y $\Delta(s_{i+1})$ no están conectados. Para $k = 1, 2, \ldots,$ si los sectores C_{x-1} y C_{x+1} no contienen ningún punto de $p_1, p_2, \ldots, p_{k-1}$, entonces se verifica:*

a) Si p_k pertenece a C_k, entonces I'_k tiene un extremo en C_x y el otro en $(C_{x-1} \cup C_x \cup C_{x+1})^c$.

b) Si p_k pertenece a $(C_{x-1} \cup C_x \cup C_{x+1})^c$, entonces I_k tiene un extremo en C_x y el otro en $(C_{x-1} \cup C_x \cup C_{x+1})^c$.

Demostración. Como los puntos de inserción $p_1, p_2, \ldots, p_{k-1}$ no están contenidos en C_{x-1} ni en C_{x+1}, por el razonamiento anterior los puntos $q_1 = \Delta(v_1)$ y $q_2 = \Delta(v_2)$ que primero encuentra el escaneo izquierda-derecha del procedimiento de inserción deben verificar $q_1 \in C_x$ y $q_2 \in (C_{x-1} \cup C_x \cup C_{x+1})^c$. Por tanto, si $p_k \in C_x$, entonces $I'_k = \Delta([u_k, v_2])$ verifica la afirmación *a*). De modo parecido si $p_k \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, entonces $I_k = \Delta([v_1, u_k])$ verifica la afirmación *b*).

□

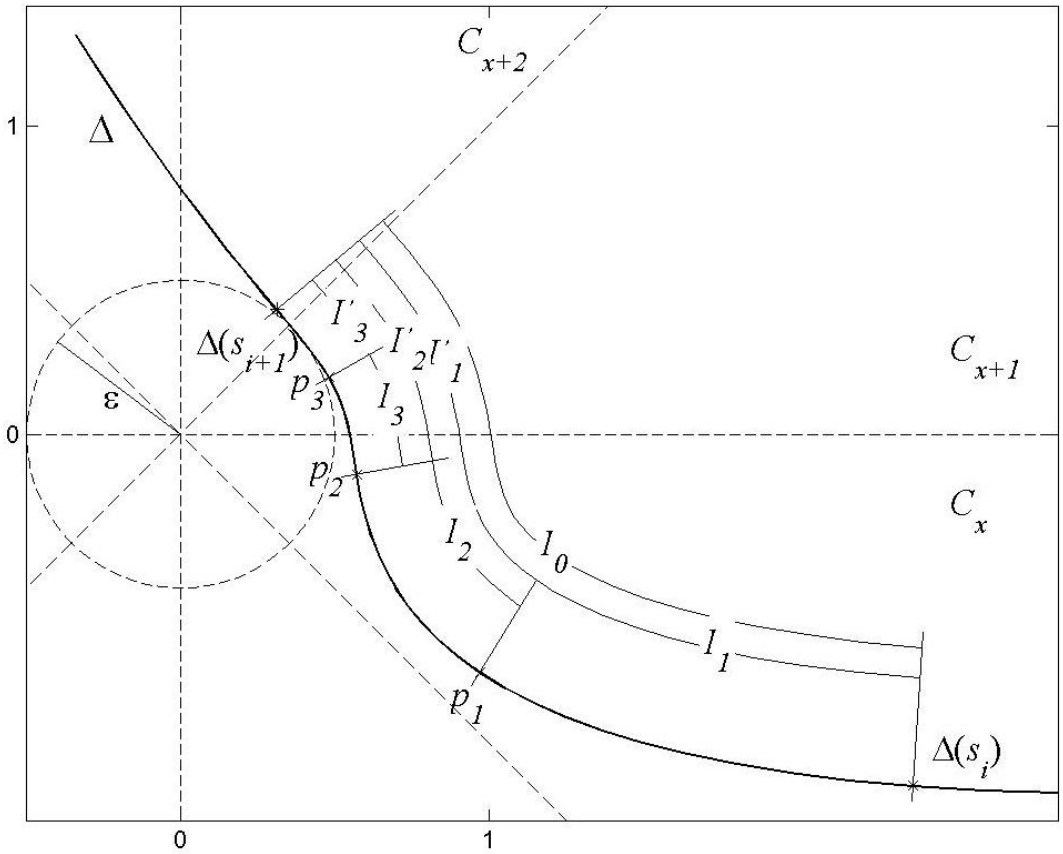


Figura 2.8: Los puntos $\Delta(s_i)$ y $\Delta(s_{i+1})$ y los intervalos I_k y I'_k están marcados.

Para proseguir la argumentación, necesitamos una hipótesis adicional sobre Δ. En esta sección supondremos que $\Delta : [a, b] \to \mathbb{C}$ es uniformemente parametrizada (esto es, para cada $x, y \in [a, b]$, $|y - x| = \int_x^y d\Delta(t)$).

Llamamos M a la longitud de arco del segmento, $M = \text{longarc}\,(\Delta([s_i, s_{i+1}]))$. Si suponemos que Δ es uniformemente parametrizada, tenemos que $M = (s_{i+1} - s_i)$ [Kolmogorov and Fomın, 1975].

Proposición 2. *Si k es tal que los sectores C_{x-1} y C_{x+1} no contienen ningún punto de $p_1, p_2, \ldots, p_{k-1}, p_k$, entonces p_{k+1} es el punto medio o de I_k o de I'_k, el que tenga sus extremos no conectados. Además, si Δ es uniformemente parametrizada, entonces para $j = 1, 2, \ldots, k+1$,* $\text{longarc}(I_j) = \text{longarc}(I'_j) = \dfrac{M}{2^j}$. *En particular,* $\text{longarc}(I_k) = \text{longarc}(I'_k) = \dfrac{M}{2^k}$.

Demostración. Por inducción: para $k = 0$, p_1 es el punto medio de $I_0 = I'_0 = \Delta([s_i, s_{i+1}])$ por la parametrización uniforme. Para k general, la hipótesis de inducción es que p_k es el punto medio de I_{k-1} o de I'_{k-1}. La secuencia que resulta después de esta k-ésima inserción es $(\ldots, \Delta(s_i), \ldots, e_1, p_k, e_2, \ldots, \Delta(s_{i+1}), \ldots)$, siendo e_1, e_2 los extremos no conectados de uno de los dos segmentos I_{k-1} o I'_{k-1}. Notemos que como p_k no está en C_{x-1} ni en C_{x+1}, entonces debe ser $p_k \in C_x$ o $p_k \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, y por la proposición 1 los extremos de o bien I_k (que son e_1, p_k) o bien I'_k (que son p_k , e_2) están no conectados. El siguiente punto de inserción p_{k+1} lo introduce el procedimiento tras encontrar dos puntos q_1 y q_2 no conectados. Por el recorrido izquierda-derecha de la secuencia, debemos concluir que estos puntos son o bien $q_1 = e_1$ y $q_2 = p_k$, o bien $q_1 = p_k$ y $q_2 = e_2$. Esto es, p_{k+1} es el punto medio de I_k o de I'_k.

Para la afirmación sobre las longitudes, notemos que $\text{longarc}(I_0) = \text{longarc}(I'_0) = M$, y que, para $j = 1, 2, \ldots, k+1$, p_j divide o a I_j o a I'_j en dos mitades, I_{j+1} y I'_{j+1}. Por tanto $\text{longarc}(I_{j+1}) = \text{longarc}(I'_{j+1})$, y $\text{longarc}(I_{j+1}) = \dfrac{\text{longarc}(I_j)}{2}$, y por inducción se deriva la expresión para la longitud de arco.

□

Recordemos que una curva es ε-singular si su distancia mínima al origen es ε. Tenemos que:

Proposición 3. *La longitud de arco de un segmento de una curva ε-singular con sus extremos no conectados es estrictamente mayor que $\dfrac{\pi}{4}\varepsilon$.*

Demostración. Esto puede verse considerando una circunferencia de radio ε, como la mostrada con una linea rayada en la figura 2.8. Es la curva con menor longitud de arco entre las ε-singulares. Y sus puntos en sectores no adyacentes tienen una diferencia angular de al menos $\dfrac{\pi}{4}$, correspondiente a una longitud de arco de $\dfrac{\pi}{4}\varepsilon$.

□

Con el decrecimiento en la longitud de arco de los intervalos (si se cumplen las condiciones de la proposición 2) y el concepto de ε-singularidad, podemos progresar en la prueba de la finitud del procedimiento de inserción entre s_i y s_{i+1}. Se puede conjeturar que el número de inserciones va a estar acotado por una fórmula que involucre logaritmos, ya que cada inserción divide por la mitad la diferencia entre dos parámetros consecutivos de la secuencia. La proposición 4 nos

da la cota concreta de $\left\lceil \lg_2 \left(\frac{4M}{\pi\varepsilon} \right) \right\rceil$ para el número de iteraciones requeridas hasta que un punto de inserción está conectado con $\Delta(s_i)$. $\lceil x \rceil$ es el menor entero mayor o igual que x (el redondeo hacia arriba).

Proposición 4. *Supongamos que Δ es uniformemente parametrizada y ε-singular. Si $\Delta(s_i)$ y $\Delta(s_{i+1})$ no están conectados, siendo C_x el sector que contiene a $\Delta(s_i)$, entonces hay un punto de inserción p_K verificando $p_K \in C_{x-1} \cup C_{x+1}$, con $K \leq \left\lceil \lg_2 \left(\frac{4M}{\pi\varepsilon} \right) \right\rceil$.*

Demostración. Construiremos una cota del número de iteraciones del procedimiento de inserción que se requieren hasta que un punto de inserción pertenece a $C_{x-1} \cup C_{x+1}$. Definimos k_0 como el entero verificando $\frac{M}{2^{k_0}} \leq \frac{\pi}{4}\varepsilon < \frac{M}{2^{k_0-1}}$. Esto es equivalente a $\frac{4M}{\pi\varepsilon} \leq 2^{k_0} < 2\frac{4M}{\pi\varepsilon}$, es decir $\lg_2 \left(\frac{4M}{\pi\varepsilon} \right) \leq k_0 < \lg_2 \left(\frac{4M}{\pi\varepsilon} \right) + 1$, por tanto $k_0 = \left\lceil \lg_2 \left(\frac{4M}{\pi\varepsilon} \right) \right\rceil$. Notemos que por la proposición 3 la longitud de arco de $\Delta([s_i, s_{i+1}])$, M, verifica $M > \frac{\pi\varepsilon}{4}$, es decir $\frac{4M}{\pi\varepsilon} > 1$, luego $\lg_2 \left(\frac{4M}{\pi\varepsilon} \right) > 0$ y $\left\lceil \lg_2 \left(\frac{4M}{\pi\varepsilon} \right) \right\rceil \geq 1$. Por tanto $k_0 \geq 1$.

Si para algún K con $K < k_0$ el punto de inserción p_K verifica $p_K \in C_{x-1} \cup C_{x+1}$, entonces se verifica la afirmación de la proposición, porque $K < k_0 = \left\lceil \lg_2 \left(\frac{4M}{\pi\varepsilon} \right) \right\rceil$. Si, por el contrario, todos los puntos de inserción p_K con $K < k_0$ verifican $p_K \notin C_{x-1} \cup C_{x+1}$ (o equivalentemente, los sectores C_{x-1} y C_{x+1} no contienen ningún punto de $p_1, p_2, \ldots, p_{k_0-1}$) estamos en la hipótesis de la proposición 1 (con $k = k_0$). Si suponemos, buscando una contradicción, que $p_{k_0} \notin C_{x-1} \cup C_{x+1}$ (esto es, $p_{k_0} \in C_x$ o $p_{k_0} \in (C_{x-1} \cup C_x \cup C_{x+1})^c$), entonces o bien I_{k_0} o bien I'_{k_0} tiene sus extremos no conectados. Además, tenemos que $\text{longarc}(I_{k_0}) = \frac{M}{2^{k_0}}$ por la proposición 2, y $\frac{M}{2^{k_0}} \leq \frac{\pi}{4}\varepsilon$ por definición de k_0. Y no es posible que $\text{longarc}(I_{k_0}) = \text{longarc}(I'_{k_0}) \leq \frac{\pi}{4}\varepsilon$ con sus extremos no conectados, porque esto contradice la proposición 3. Debemos concluir por tanto que $p_{k_0} \in C_{x-1} \cup C_{x+1}$.

□

Notemos que siendo p_K el primer punto de inserción con $p_K \in C_{x-1} \cup C_{x+1}$, los puntos en el segmento $(\Delta(s_i), \ldots, q_1)$ de la secuencia $S = (\ldots, \Delta(s_i), \ldots, q_1, p_K,$

$q_2, \ldots, \Delta(s_{i+1}), \ldots)$ pertenecen a C_x. Esto es porque el procedimiento de inserción recorre la secuencia de izquierda a derecha, y por tanto no es posible que uno de los puntos de $(\Delta(s_i), \ldots, q_1)$ pertenezca a $(C_{x-1} \cup C_x \cup C_{x+1})^c$, porque en tal caso el punto de inserción K-ésimo habría sido insertado en una posición anterior a la de q_1. Tampoco es posible que alguno de estos puntos pertenezca a $C_{x-1} \cup C_{x+1}$, porque p_K es el primer punto insertado con esta propiedad. Por tanto los puntos de $(\Delta(s_i), \ldots, q_1)$ deben pertenecer a C_x.

Usando la proposición 4 podemos encontrar una cota al número de puntos de inserción requeridos para conectar $\Delta(s_i)$ y $\Delta(s_{i+1})$. Recuérdese que $\Delta([s_i, s_{i+1}])$ cruza N bordes entre sectores si el intervalo de parámetros contiene N valores f_j, $j = 1, \ldots, N$, con $s_i \leq f_1 < f_2 < \ldots < f_N \leq s_{i+1}$, cuyas imágenes $\Delta(f_j)$ pertenecen a algún borde.

Notemos que si $\Delta([s_i, s_{i+1}])$ cruza $N = 0$ bordes, el número de puntos de inserción requeridos en este segmento es cero porque $\Delta(s_i)$ y $\Delta(s_{i+1})$ están en el mismo sector. De modo parecido, si $N = 1$, entonces $\Delta(s_i)$ y $\Delta(s_{i+1})$ están conectados, y no se requiere ningún punto de inserción. En el caso de que el número de cruces N sea mayor, se verifica la siguiente afirmación:

Lema 1. *Supongamos que Δ es uniformemente parametrizada y ε-singular, que $\Delta(s_i)$ y $\Delta(s_{i+1})$ no están conectados, y que $\Delta([s_i, s_{i+1}])$ cruza N bordes. Para $N \geq 2$, el número de puntos de inserción entre $\Delta(s_i)$ y $\Delta(s_{i+1})$ está acotado por* $(N-1)\left\lceil \lg_2\left(\frac{4M}{\pi\varepsilon}\right)\right\rceil$.

Demostración. Probamos la afirmación por inducción en N.

Supongamos primero que $N = 2$; como $\Delta(s_i)$ y $\Delta(s_{i+1})$ no están conectados, los puntos de $\Delta([s_i, s_{i+1}])$ están contenidos en tres sectores. Si C_x denota el sector que contiene a $\Delta(s_i)$, entonces hay dos casos posibles: o bien $\Delta(s_i) \in C_x$, $\Delta(s_{i+1}) \in C_{x+2}$ y $\Delta([s_i, s_{i+1}]) \subset C_x \cup C_{x+1} \cup C_{x+2}$ (en sentido antihorario), o bien $\Delta(s_i) \in C_x$, $\Delta(s_{i+1}) \in C_{x-2}$ y $\Delta([s_i, s_{i+1}]) \subset C_x \cup C_{x-1} \cup C_{x-2}$ (sentido horario). La figura 2.8 más arriba representa el primer caso. Por la proposición 4, en menos de $\left\lceil \lg_2\left(\frac{4M}{\pi\varepsilon}\right)\right\rceil$ iteraciones el procedimiento de inserción pone un punto p_K en $C_{x-1} \cup C_{x+1}$. Con esta inserción la secuencia $(\Delta(s_i), \ldots, p_K, \ldots, \Delta(s_{i+1}))$ verifica la propiedad de conexión y el procedimiento acaba (entre $\Delta(s_i)$ y $\Delta(s_{i+1})$).

Este razonamiento para $N = 2$ es el caso base de la inducción que demuestra el lema. Para el paso inductivo, supongamos ahora que el segmento de curva

$\Delta([s_i, s_{i+1}])$, de longitud M, cruza N bordes, con $N > 2$. La hipótesis de inducción, que suponemos cierta, es que cualquier segmento de curva cruzando $N-1$ bordes requiere como mucho $(N-2)\left\lceil \lg_2\left(\frac{4M'}{\pi\varepsilon}\right)\right\rceil$ puntos de inserción, siendo M' su longitud. Demostraremos que $\Delta([s_i, s_{i+1}])$ requiere como mucho $(N-1)\left\lceil \lg_2\left(\frac{4M}{\pi\varepsilon}\right)\right\rceil$ puntos de inserción, dividiendo este segmento en dos partes, una primera que va de $\Delta(s_i)$ a un punto en C_{x-1} o en C_{x+1}, y una segunda parte que va de este punto a $\Delta(s_{i+1})$.

Por la proposición 4, tenemos que para cierto $K \leq \left\lceil \lg_2\left(\frac{4M}{\pi\varepsilon}\right)\right\rceil$, el punto $p_K = \Delta(u_K)$ está en un sector que es adyacente a $\Delta(s_i)$, ya sea C_{x-1} o C_{x+1}. Además, según la nota tras la prueba de la proposición 4, los puntos en el segmento $(\Delta(s_i), \ldots, q_1)$ de la secuencia $(\ldots, \Delta(s_i), \ldots, q_1, p_K, q_2, \ldots, \Delta(s_{i+1}), \ldots)$ pertenecen a C_x. La figura 2.9 muestra un ejemplo con $K = 3$ y $p_K \in C_{x+1}$.

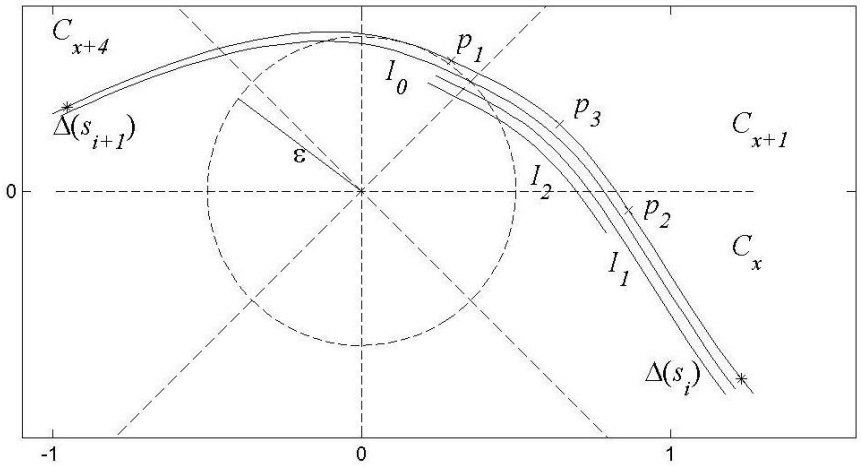


Figura 2.9: El segmento de curva entre $\Delta(s_i)$ y $\Delta(s_{i+1})$ tiene una longitud de arco de M, y $\Delta([s_i, s_{i+1}])$ cruza cuatro bordes.

Dividimos el segmento $\Delta([s_i, s_{i+1}])$ en dos subsegmentos consecutivos, $\Delta([s_i, u_K])$ y $\Delta([u_K, s_{i+1}])$. Notemos que $\Delta([s_i, u_K])$ cruza al menos un borde (el que está entre C_x y C_{x-1} o C_{x+1}), pero no necesariamente bordes distintos en cada cruce, como se puede ver por ejemplo en las posiciones de la figura 2.10. Además, después de la K-ésima iteración el procedimiento de inserción no inserta puntos en $\Delta([s_i, u_K])$, porque todos los puntos de la secuencia en este segmento, $(\Delta(s_i), \ldots, q_1, p_K)$ pertenecen a C_x excepto p_K que pertenece a C_{x-1} o a C_{x+1}.

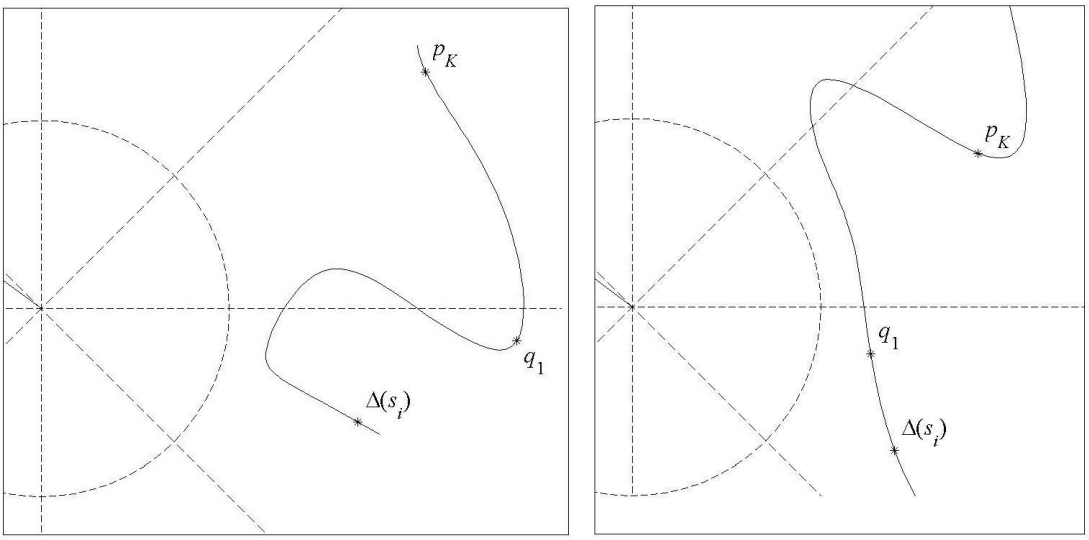


Figura 2.10: El segmento $\Delta([s_i, u_K])$ cruza al menos un borde. q_1 es el punto previo a p_K en la secuencia.

Llamemos α al número de bordes cruzados por $\Delta([s_i, u_K])$. El resto del segmento de curva original, $\Delta([u_K, s_{i+1}])$, cruza $N - \alpha$ bordes, con $\alpha \geq 1$. Consideremos $\Delta([u_K, s_{i+1}])$ como una curva que cruza $N - 1$ bordes o menos. Recuérdese que la hipótesis de inducción es que cualquier segmento de curva cruzando $N - 1$ bordes requiere como mucho $(N-2)\left\lceil \lg_2\left(\frac{4M'}{\pi\varepsilon}\right)\right\rceil$ puntos de inserción, siendo M' su longitud. Por tanto $\Delta([u_K, s_{i+1}])$ requiere como mucho $(N-2)\left\lceil \lg_2\left(\frac{4M'}{\pi\varepsilon}\right)\right\rceil$ inserciones, siendo la longitud de arco $M' = (s_{i+1} - u_K)$. Para concluir, tenemos que con K inserciones (siendo $K \leq \left\lceil \lg_2\left(\frac{4M}{\pi\varepsilon}\right)\right\rceil$), el segmento $\Delta([s_i, s_{i+1}])$ da lugar a un subsegmento $\Delta([u_K, s_{i+1}])$, que requiere como mucho $(N-2)\left\lceil \lg_2\left(\frac{4M'}{\pi\varepsilon}\right)\right\rceil$ inserciones. En total no más de $\left\lceil \lg_2\left(\frac{4M}{\pi\varepsilon}\right)\right\rceil + (N-2)\left\lceil \lg_2\left(\frac{4M'}{\pi\varepsilon}\right)\right\rceil$ inserciones. Como $M' < M$, esto es menor o igual que $(N-1)\left\lceil \lg_2\left(\frac{4M}{\pi\varepsilon}\right)\right\rceil$, que es lo que se pretendía demostrar.

□

Obtenemos el siguiente teorema, que evita la dependencia de N en el lema anterior, acotando el número N de bordes cruzados por un segmento de longitud

de arco M de una curva ε-singular.

Recuérdese que para una secuencia $S = (s_0, \ldots, s_n)$, su paso máximo es $|S| = \max_{0 \le i \le n-1} (s_{i+1} - s_i)$.

Teorema 1. *Si $\Delta : [a, b] \to \mathbb{C}$ es ε-singular con $\varepsilon > 0$, uniformemente parametrizada, el procedimiento de inserción para la curva Δ con una secuencia inicial $S^{(0)} = (s_0, \ldots, s_n)$ de valores crecientes con $s_0 = a$, $s_n = b$, concluye en menos de*

$$\frac{4(b-a)}{\pi\varepsilon} \left\lceil \lg_2 \left(\frac{4|S^{(0)}|}{\pi\varepsilon} \right) \right\rceil$$

inserciones.

Demostración. Consideremos una curva circular situada a distancia constante ε del origen. La distancia a lo largo de la curva entre puntos en bordes distintos es de $\frac{\pi\varepsilon}{4}$. Cualquier otra curva ε-singular tiene una distancia entre puntos situados en bordes distintos mayor o igual que este valor. Por tanto, entre las curvas de longitud de arco M, la curva circular cruza el máximo número de bordes. Para calcular este número, N_{max}, llamemos f_j, $j = 1, \ldots, N_{max}$, a los valores del parámetro correspondientes a puntos $\Delta(f_j)$ en un borde, en orden creciente, $f_j < f_{j+1}$. Dos puntos consecutivos $\Delta(f_j)$, $\Delta(f_{j+1})$ están a una distancia a lo largo de la curva de $\frac{\pi\varepsilon}{4}$, y por la parametrización uniforme, $(f_{j+1} - f_j) = \frac{\pi\varepsilon}{4}$. En general, para calcular el número x de puntos a distancia d dentro de un intervalo de longitud m, notemos que con x puntos igualmente espaciados a distancia d cubrimos una longitud de $(x-1)d$. Por tanto tenemos que $(x-1)d \le m < xd$, esto es $x = \left\lfloor \frac{m}{d} \right\rfloor + 1$. En nuestro caso, el número N_{max} de valores del parámetro f_j a una distancia de $\frac{\pi\varepsilon}{4}$ es $\left\lfloor \frac{M}{\frac{\pi\varepsilon}{4}} \right\rfloor + 1 = \left\lfloor \frac{4M}{\pi\varepsilon} \right\rfloor + 1$. Por tanto el número de bordes cruzados por cualquier curva debe ser menor o igual que este valor. En particular, siendo N el número de bordes cruzados por $\Delta([s_i, s_{i+1}])$, $N \le N_{max} = \left\lfloor \frac{4M}{\pi\varepsilon} \right\rfloor + 1$, luego $N - 1 \le \frac{4M}{\pi\varepsilon}$. Aplicando el lema previo, el máximo número de puntos de inserción en $\Delta([s_i, s_{i+1}])$ está acotado por $\frac{4M}{\pi\varepsilon} \left\lceil \lg_2 \left(\frac{4M}{\pi\varepsilon} \right) \right\rceil$.

Hemos deducido una cota $\frac{4M}{\pi\varepsilon} \left\lceil \lg_2 \left(\frac{4M}{\pi\varepsilon} \right) \right\rceil$ del número de puntos de inserción requeridos en $\Delta([s_i, s_{i+1}])$. Esto es válido para cada i, $0 \le i \le n-1$, siendo $S^{(0)} = (s_0, \ldots, s_n)$ la secuencia inicial, salvo que la distancia M entre

$\Delta(s_i)$ y $\Delta(s_{i+1})$ puede variar con i. En cualquier caso, esta distancia es igual a $(s_{i+1} - s_i)$ por la parametrización uniforme. La suma de estos máximos nos da $\sum_{i=0}^{n-1} \frac{4(s_{i+1} - s_i)}{\pi\varepsilon} \left\lceil \lg_2 \left(\frac{4(s_{i+1} - s_i)}{\pi\varepsilon} \right) \right\rceil$. Además, los $(s_{i+1} - s_i)$ son menores o iguales a $|S^{(0)}|$ por la definición de paso máximo. Por tanto $\lg_2 \left(\frac{4(s_{i+1} - s_i)}{\pi\varepsilon} \right) \leq \lg_2 \left(\frac{4|S^{(0)}|}{\pi\varepsilon} \right)$ para $i = 0, 1, 2, \ldots, n-1$, y el sumatorio anterior es menor o igual que $\left(\sum_{i=0}^{n-1} \frac{4(s_{i+1} - s_i)}{\pi\varepsilon} \right) \left\lceil \lg_2 \left(\frac{4|S^{(0)}|}{\pi\varepsilon} \right) \right\rceil$. Esto puede simplificarse porque $\sum_{i=0}^{n-1} (s_{i+1} - s_i) = (b-a)$, y entonces el total de inserciones es menor o igual que $\frac{4(b-a)}{\pi\varepsilon} \left\lceil \lg_2 \left(\frac{4|S^{(0)}|}{\pi\varepsilon} \right) \right\rceil$. □

2.4. Cota del procedimiento de inserción para curvas Lipschitzianas

Debemos generalizar el teorema anterior para manejar curvas ε-singulares no necesariamente uniformemente parametrizadas. Debe notarse que la secuencia resultante del proceso de inserción depende de la parametrización: esto es, dos parámetros Δ y Δ' con la misma curva imagen en $\mathbb{C}$ pueden producir diferentes puntos de inserción. El análisis anterior del coste del procedimiento de inserción para curvas uniformemente parametrizadas puede extenderse a las curvas Lipschitzianas, más generales. Estas curvas son las que verifican que hay una constante L con $|\Delta(y) - \Delta(x)| \leq L|y - x|$ para cada $x, y \in [a, b]$.

Este relajamiento de las hipótesis es necesario para la aplicación que tenemos en mente, el cálculo del índice de curvas $\Delta = f(\Gamma)$ para un polinomio f y una curva Γ bordeando un área de interés. La curva Γ normalmente se define concatenando segmentos uniformemente parametrizados, y esto hace que Γ sea uniformemente parametrizada. Pero su transformación $\Delta = f(\Gamma)$ por un polinomio f no es en general uniformemente parametrizada, aunque sí es Lipschitziana [Kolmogorov and Fomın, 1975].

El número de puntos de inserción en cualquier curva Lipschitziana tiene como cota la determinada en el teorema 2. La ruta seguida para probarlo es similar a la

del teorema 1.

Para una secuencia inicial $S^{(0)} = (s_0, \ldots, s_n)$ consideremos un segmento de curva $\Delta([s_i, s_{i+1}])$. En este caso $M = (s_{i+1} - s_i)$ no es la longitud de arco del segmento. Sin embargo, en una curva Lipschitziana con constante L, se verifica que $\text{longarc}(\Delta[s,t]) \leq L(t-s)$ (ver [Kolmogorov and Fomın, 1975]). En particular $\text{longarc}(\Delta[s_i, s_{i+1}]) \leq L(s_{i+1} - s_i)$. Con esta cota de la longitud de arco podemos probar el teorema 2 de modo similar al teorema 1.

Llamemos p_k al k-ésimo punto de inserción entre $\Delta(s_i)$ y $\Delta(s_{i+1})$, y u_k a su parámetro, $k = 1, 2, \ldots$, como en el anterior apartado. También, para $k = 1, 2, \ldots$, llamemos I_k al segmento de curva Δ que une p_k con el punto previo en la secuencia, y I'_k al segmento que une p_k con el siguiente. Ver figura 2.8.

La proposición 1 de la sección previa se aplica a curvas con cualquier tipo de parámetro. La siguiente proposición es específica para curvas Lipschitzianas, y es análoga a la proposición 2 de la sección previa.

Proposición 5. *Supongamos que Δ es Lipschitziana con constante L. Si k es tal que los sectores C_{x-1} y C_{x+1} no contienen ningún punto de $p_1, p_2, \ldots, p_{k-1}, p_k$, entonces p_{k+1} tiene, como parámetro, la media de los parámetros de los extremos de o bien I_k o bien I'_k, aquel que tenga sus extremos no conectados. Además, para $j = 1, 2, \ldots, k+1$, $\text{longarc}(I_j) \leq \dfrac{LM}{2^j}$ y $\text{longarc}(I'_j) \leq \dfrac{LM}{2^j}$.*

Demostración. Por inducción: para $k = 0$, $I_0 = I'_0 = \Delta([s_i, s_{i+1}])$ y p_1 tiene de parámetro $u_1 = \dfrac{s_i + s_{i+1}}{2}$, que es lo que afirma la proposición. Para k general, la hipótesis de inducción es que $p_k = \Delta(u_k)$ siendo u_k la media de los parámetros de los extremos de I_{k-1} o de I'_{k-1}, lo que implica que la secuencia tras esta inserción queda $(\ldots, \Delta(s_i), \ldots, e_1, p_k, e_2, \ldots, \Delta(s_{i+1}), \ldots)$, siendo e_1, e_2 los extremos no conectados de uno de los dos segmentos. El siguiente punto de inserción p_{k+1} lo inserta el procedimiento después de encontrar dos puntos q_1 y q_2 no conectados, y p_{k+1} tiene como parámetro la media de los parámetros de estos puntos. Notemos que como p_k no está en C_{x-1} ni en C_{x+1}, entonces debe ser $p_k \in C_x$ o $p_k \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, y por la proposición 1 los extremos de ya sea I_k (que son e_1, p_k) ya sea I'_k (que son p_k, e_2) están no conectados. Como el procedimiento recorre la secuencia de izquierda a derecha, los puntos no conectados encontrados son o bien $q_1 = e_1$ y $q_2 = p_k$, o bien $q_1 = p_k$ y $q_2 = e_2$. Esto es, p_{k+1} tiene como parámetro la media de los parámetros de los extremos de I_k o de I'_k.

Este razonamiento es similar al de la proposición 2 de la sección anterior, pero ahora el punto de inserción p_k no es necesariamente el punto medio del segmento de curva entre los puntos no conexos.

Para las longitudes, tenemos por inducción que la diferencia entre los parámetros de los extremos de I_j es $\frac{M}{2^j}$, para $j = 1, 2, \ldots, k+1$. Lo mismo con I'_j. Como se comentó más arriba, en una curva Lipschitziana con constante L, la longitud de arco de un segmento de curva verifica $\text{longarc}(\Delta[s,t]) \leq L(t-s)$. Luego $\text{longarc}(I_j) \leq L\frac{M}{2^j}$ y $\text{longarc}(I'_j) \leq L\frac{M}{2^j}$.

□

Finalmente, probamos:

Proposición 6. *Supongamos que Δ es Lipschitziana de constante L y ε-singular. Si $\Delta(s_i)$ y $\Delta(s_{i+1})$ no están conectados, siendo C_x el sector que contiene a $\Delta(s_i)$, entonces el primer punto insertado verificando $p_K \in C_{x-1} \cup C_{x+1}$ es tal que $K \leq \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$.*

Demostración. Definamos k_0 como el entero verificando $\frac{LM}{2^{k_0}} \leq \frac{\pi}{4}\varepsilon < \frac{LM}{2^{k_0-1}}$. Tenemos que $k_0 = \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil \geq 1$.

Si el primer punto de inserción perteneciente a $C_{x-1} \cup C_{x+1}$ es p_K con $K < k_0$, entonces la afirmación $K \leq \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$ de la proposición es obvia. En caso contrario, todos los puntos $p_1, p_2, \ldots, p_{k_0-1}$ están fuera de $C_{x-1} \cup C_{x+1}$, y por tanto el procedimiento de inserción alcanza la iteración k_0-ésima. Así estamos en la hipótesis de la proposición 1 (con $k = k_0$), y si suponemos que $p_{k_0} \notin C_{x-1} \cup C_{x+1}$, entonces o bien I_{k_0} o bien I'_{k_0} tiene sus extremos no conectados. Además por la proposición 5, $\text{longarc}(I_{k_0}) = \text{longarc}(I'_{k_0}) \leq \frac{LM}{2^{k_0}}$. por definición de k_0 tenemos que $\frac{LM}{2^{k_0}} \leq \frac{\pi}{4}\varepsilon$, así que $\text{longarc}(I_{k_0}) = \text{longarc}(I'_{k_0}) \leq \frac{\pi}{4}\varepsilon$. Esto sin embargo contradice la proposición 3. Debemos pues concluir que $p_{k_0} \in C_{x-1} \cup C_{x+1}$.

□

Con esta proposición, podemos seguir un argumento similar al de la sección anterior para mostrar que, si el segmento de curva $\Delta([s_i, s_{i+1}])$ cruza N bordes, el número de puntos de inserción requeridos no puede ser mayor que $(N-1)\left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$.

Lema 2. *Supongamos que Δ es Lipschitziana con constante L y ε-singular. Para $N \geq 2$, si $\Delta(s_i)$ y $\Delta(s_{i+1})$ no están conectados, el número de puntos de inserción está acotado por $(N-1)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil$.*

Demostración. Se puede repetir la prueba del lema de la sección anterior con la cota $\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil$ de la proposición 6 en vez de $\left\lceil \lg_2\left(\frac{4M}{\pi\varepsilon}\right)\right\rceil$, concluyendo con la afirmación deseada.

□

Teorema 2. *Si $\Delta : [a,b] \to \mathbb{C}$ es ε-singular con $\varepsilon \geq 0$, y Lipschitziana con constante L, entonces el procedimiento de inserción para la curva Δ con secuencia inicial $S^{(0)} = (s_0, \ldots, s_n)$, $s_0 = a$, $s_n = b$, concluye en menos de $\frac{4L(b-a)}{\pi\varepsilon}\left\lceil \lg_2\left(\frac{4L|S^{(0)}|}{\pi\varepsilon}\right)\right\rceil$ inserciones.*

Demostración. Vamos a acotar el número N de bordes cruzados por un segmento $\Delta([s_i, s_{i+1}])$ de una curva ε-singular Lipschitziana. Consideremos los valores del parámetro f_j, $j = 1, \ldots, N$, correspondientes a los puntos $\Delta(f_j)$ que estén en un borde. Dos de tales puntos que estén consecutivos, $\Delta(f_j)$ y $\Delta(f_{j+1})$, están separados por un segmento de curva de longitud de arco mayor o igual que $\frac{\pi\varepsilon}{4}$. Esto es, $\text{longarc}(\Delta(f_{j+1} - f_j)) \geq \frac{\pi\varepsilon}{4}$. Además, por Lipschitzianidad, $\text{longarc}(\Delta(f_{j+1} - f_j)) \leq L(f_{j+1} - f_j)$, y encadenando las desigualdades deducimos que $\frac{\pi\varepsilon}{4} \leq L(f_{j+1} - f_j)$ para $j = 1, \ldots, N$. El máximo número de valores f_j a una distancia de $\frac{\pi\varepsilon}{4L}$, dentro de un intervalo de longitud M es $\left\lfloor \frac{M}{\frac{\pi\varepsilon}{4L}} \right\rfloor + 1 = \left\lfloor \frac{4LM}{\pi\varepsilon} \right\rfloor + 1$. Por tanto $N \leq \left\lfloor \frac{4LM}{\pi\varepsilon} \right\rfloor + 1$.

Aplicando el lema previo, como $N - 1 \leq \frac{4LM}{\pi\varepsilon}$, el número máximo de puntos de inserción en $\Delta([s_i, s_{i+1}])$ está acotado por $\frac{4LM}{\pi\varepsilon}\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil$.

En general, como $M = (s_{i+1} - s_i)$, el número de puntos de inserción en cada $\Delta([s_i, s_{i+1}])$, $i = 0, 1, 2, \ldots, n-1$, de la secuencia inicial $S^{(0)} = (s_0, \ldots, s_n)$ está acotado por $\frac{4L(s_{i+1} - s_i)}{\pi\varepsilon}\left\lceil \lg_2\left(\frac{4L(s_{i+1} - s_i)}{\pi\varepsilon}\right)\right\rceil$. Para sumar estas cotas, tenemos que la expresión $\sum_{i=0}^{n-1} \frac{4L(s_{i+1} - s_i)}{\pi\varepsilon}\left\lceil \lg_2\left(\frac{4L(s_{i+1} - s_i)}{\pi\varepsilon}\right)\right\rceil$ es menor o igual

que $\sum_{i=0}^{n-1} \frac{4L(s_{i+1}-s_i)}{\pi\varepsilon} \left\lceil \lg_2 \left(\frac{4L|S^{(0)}|}{\pi\varepsilon} \right) \right\rceil$ porque $(s_{i+1}-s_i) \leq |S^{(0)}|$. Además, como $\sum_{i=0}^{n-1} (s_{i+1}-s_i) = (b-a)$, tenemos que el total de inserciones es menor que $\frac{4L(b-a)}{\pi\varepsilon} \left\lceil \lg_2 \left(\frac{4L|S^{(0)}|}{\pi\varepsilon} \right) \right\rceil$.

□

2.5. Evitando giros perdidos

Debemos evitar situaciones como la que se muestra en la figura 2.11 para asegurar el correcto cálculo del índice mediante el procedimiento de inserción. Los puntos de la secuencia $S = (s_0, \ldots, s_n)$ verifican la propiedad de conexión, pero el índice de la poligonal $\widetilde{\Delta}$ no es igual al índice de Δ, ya que $\mathrm{Ind}(\widetilde{\Delta}) = 1$ y $\mathrm{Ind}(\Delta) = 2$.

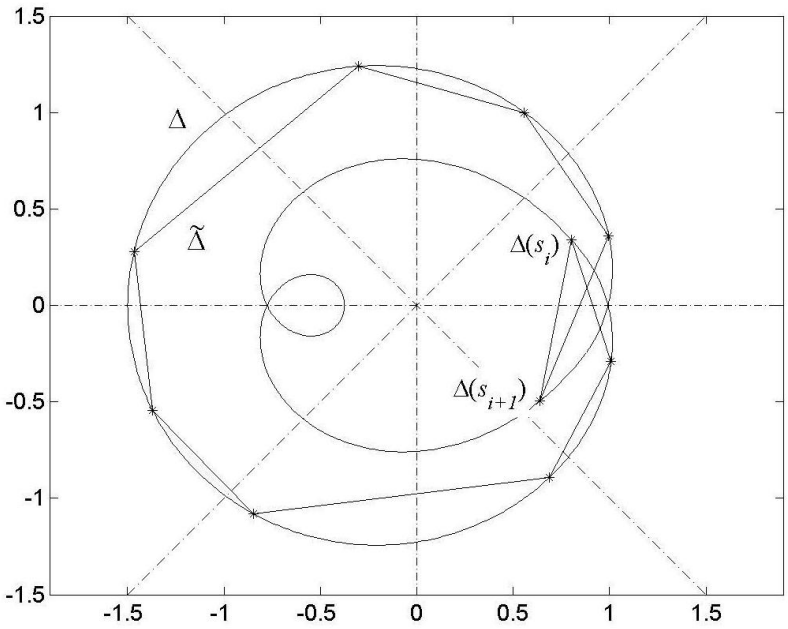


Figura 2.11: La curva Δ y su aproximación poligonal $\widetilde{\Delta}$ tienen distinto índice, 2 y 1 respectivamente.

Cada par de puntos consecutivos en la secuencia $S = (s_0, \ldots, s_n)$ define un segmento de curva $\Delta_i : [s_i, s_{i+1}] \to \mathbb{C}$, para $i = 0, 1, 2, \ldots, n-1$. Si cualquiera de estos segmentos abarca un ángulo mayor que $\frac{3\pi}{2}$, pasando sobre siete bordes diferentes, los puntos $\Delta_i(s_i)$ y $\Delta_i(s_{i+1})$ están en sectores adyacentes. Por tanto el

procedimiento de inserción no inserta ningún punto entre ellos. Por ejemplo, en el caso de la figura 2.11, hay un cruce de C_0 a C_7 en $\widetilde{\Delta}$ (el segmento de recta de $\Delta_i(s_i)$ a $\Delta_i(s_{i+1})$) que no está en Δ (cuyo segmento curvo Δ_i sigue un camino C_0-C_1-C_2-$\cdots$-C_6-C_7). Por consiguiente, en la secuencia el contador de cruces de C_0 a C_7 se incrementa de s_i a s_{i+1}, cuando en realidad el segmento Δ_i no tiene un cruce de C_0 a C_7. Esto provoca que $\mathrm{Ind}(\widetilde{\Delta}) \neq \mathrm{Ind}(\Delta)$. Un segmento de curva que abarque un ángulo negativo (esto es, que vaya en sentido horario) menor que $-\frac{3\pi}{2}$ produce una situación similar.

Para evitar tales situaciones usamos un hecho conocido de análisis complejo: el ángulo cubierto por una curva, no necesariamente cerrada, $\gamma : [x, y] \to \mathbb{C}$ es la integral de linea $\frac{1}{i}\int_\gamma \frac{1}{w}dw$. Un *giro perdido* para la secuencia $S = (s_0, \ldots, s_n)$ es un segmento de curva $\Delta_i : [s_i, s_{i+1}] \to \mathbb{C}$, verificando que $\left|\frac{1}{i}\int_\gamma \frac{1}{w}dw\right| > \frac{3\pi}{2}$. El siguiente teorema nos da una condición suficiente para evitar giros perdidos.

Teorema 3. *Si $\Delta : [a, b] \to \mathbb{C}$ es ε-singular con $\varepsilon \neq 0$ y Lipschitziana con constante L, y se aplica el procedimiento de inserción con una secuencia inicial $S^{(0)} = (s_0, \ldots, s_n)$ verificando que su paso máximo es $|S^{(0)}| \leq \frac{3\pi\varepsilon}{2L}$, entonces no hay giros perdidos.*

Demostración. Para asegurar que no hay giros perdidos (es decir, que $\left|\frac{1}{i}\int_\gamma \frac{1}{w}dw\right| \leq \frac{3\pi}{2}$ para cada Δ_i) consideramos dos hechos geométricos: primero, que un segmento $\Delta_i = \Delta([s_i, s_{i+1}])$ de una curva ε-singular subtiende un ángulo máximo de $\frac{\mathrm{longarc}(\Delta_i)}{\varepsilon}$ (equivalentemente, que $\left|\frac{1}{i}\int_\gamma \frac{1}{w}dw\right| \leq \frac{\mathrm{longarc}(\Delta_i)}{\varepsilon}$). Esto es porque la curva ε-singular con una longitud de arco dada que subtiende el máximo ángulo es un segmento de la circunferencia de radio ε.

Segundo, que en una curva Lipschitziana con constante L, se cumple que

$$\mathrm{longarc}(\Delta_i) \leq L(s_{i+1} - s_i).$$

Luego si $|S^{(0)}| \leq \frac{3\pi\varepsilon}{2L}$, y recordando que $(s_{i+1} - s_i) \leq |S^{(0)}|$, se tiene:

$$\left|\frac{1}{i}\int_\gamma \frac{1}{w}dw\right| \leq \frac{\mathrm{longarc}(\Delta_i)}{\varepsilon} \leq \frac{L(s_{i+1} - s_i)}{\varepsilon} \leq \frac{L|S^{(0)}|}{\varepsilon} \leq \frac{L3\pi\varepsilon}{\varepsilon 2L} = \frac{3\pi}{2}$$

□

Si no hay giros perdidos, el índice de Δ coincide con el de $\widetilde{\Delta}$, y entonces, al final del procedimiento de inserción, se calcula $\mathrm{Ind}(\Delta)$ correctamente.

Resumimos los teoremas 2 y 3 sobre el procedimiento de inserción: para una curva Lipschitziana con constante L, ε-singular, $\Delta : [a,b] \to \mathbb{C}$, el procedimiento de inserción de Ying-Katz mostrado en la figura 2.6, con secuencia inicial $S^{(0)}$, verifica:

a) Si $|S^{(0)}| \leq \dfrac{3\pi\varepsilon}{2L}$, la secuencia retornada nos da $\mathrm{Ind}(\Delta)$.

b) Acaba en menos de $\dfrac{4L(b-a)}{\pi\varepsilon}\left\lceil \lg_2(\dfrac{4L|S^{(0)}|}{\pi\varepsilon})\right\rceil$ iteraciones.

Con esto, la aplicación del procedimiento de inserción requiere, en una primera fase, el cálculo de una secuencia inicial $S^{(0)}$ con $|S^{(0)}| \leq \dfrac{3\pi\varepsilon}{2L}$. La secuencia $S = (a = s_0, \ldots, s_n = b)$ de $n+1$ valores uniformemente espaciados en el intervalo $[a,b]$ verifica $|S| = \dfrac{(b-a)}{n}$. Luego tomando n tal que $\dfrac{(b-a)}{n} \leq \dfrac{3\pi\varepsilon}{2L}$, tenemos un array S que verifica $|S| \leq \dfrac{3\pi\varepsilon}{2L}$. El menor de estos n es $\left\lfloor \dfrac{2L(b-a)}{3\pi\varepsilon}\right\rfloor$. En una segunda fase, se necesitan al menos $\dfrac{4L(b-a)}{\pi\varepsilon}\left\lceil \lg_2(\dfrac{4L|S^{(0)}|}{\pi\varepsilon})\right\rceil \leq \dfrac{4L(b-a)}{\pi\varepsilon}\left\lceil \lg_2\left(\dfrac{4L\frac{3\pi\varepsilon}{2L}}{\pi\varepsilon}\right)\right\rceil = \dfrac{4L(b-a)}{\pi\varepsilon}\lceil \lg_2(6)\rceil = \dfrac{12L(b-a)}{\pi\varepsilon}$ iteraciones del bucle. Cada iteración requiere la inserción de un punto, lo que implica una evaluación de Δ en un valor del parámetro (para saber a qué sector pertenece). Consecuentemente, tenemos la siguiente expresión simplificada para el número de evaluaciones de Δ que se necesitan para el cálculo del índice: $\left\lfloor \dfrac{2L(b-a)}{3\pi\varepsilon}\right\rfloor$ para obtener una secuencia inicial $S^{(0)}$ que verifique la cota del teorema 3, más como mucho $\dfrac{12L(b-a)}{\pi\varepsilon}$ evaluaciones por el teorema 2.

En cualquier caso, el valor de ε es desconocido en general, con lo que no podemos construir a priori la secuencia $S^{(0)}$ verificando la hipótesis del teorema 3. Desarrollamos ahora una modificación del procedimiento de inserción que no requiere del conocimiento previo de ε. La modificación es tal que evita giros perdidos con cualquier secuencia inicial. Se basa en el siguiente lema:

Lema 3. *Si $(s_0, \ldots, s_m)$ es una secuencia de valores del parámetro de la curva Δ, Lipschitziana con constante L, y i es tal que Δ_i es un giro perdido, entonces*

$$(s_{i+1} - s_i) \geq \frac{|\Delta(s_i)| + |\Delta(s_{i+1})|}{L}.$$

Demostración. Recuérdese que un giro perdido en una secuencia $S = (s_0, \ldots, s_m)$ es un segmento de curva $\Delta_i : [s_i, s_{i+1}] \to \mathbb{C}$ que verifica $\left|\frac{1}{i}\int_\gamma \frac{1}{w}dw\right| > \frac{3\pi}{2}$. Si el ángulo subtendido por Δ_i, $\left|\frac{1}{i}\int_\gamma \frac{1}{w}dw\right|$, es mayor que $\frac{3\pi}{2}$, entonces su longitud de arco debe verificar $|\Delta(s_i)| + |\Delta(s_{i+1})| \leq \text{longarc}(\Delta_i)$. En realidad, para verificar esta desigualdad es suficiente que el ángulo subtendido por Δ_i sea mayor que π. La figura 2.12 puede reemplazar un razonamiento riguroso basado en envolventes convexas [Ball, 1997].

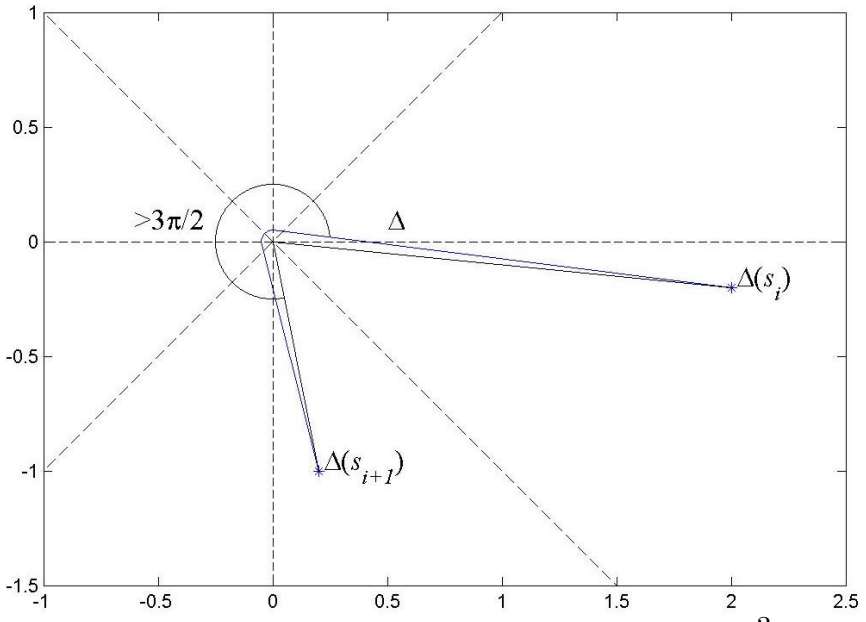


Figura 2.12: Si la curva Δ entre $\Delta(s_i)$ y $\Delta(s_{i+1})$ recorre más de $\frac{3\pi}{2}$ radianes, debe tener longitud de arco mayor que $|\Delta(s_i)| + |\Delta(s_{i+1})|$.

Además, por Lipchitzianidad, $\text{longarc}(\Delta_i) \leq L(s_{i+1} - s_i)$. Encadenando las desigualdades tenemos $|\Delta(s_i)|+|\Delta(s_{i+1})| \leq L(s_{i+1}-s_i)$, equivalentemente $(s_{i+1} - s_i) \geq \frac{|\Delta(s_i)| + |\Delta(s_{i+1})|}{L}$.

□

En vez de este lema vamos a usar más adelante el contrarrecíproco, que afirma que " Si $(s_{i+1} - s_i) < \dfrac{|\Delta(s_i)| + |\Delta(s_{i+1})|}{L}$, entonces Δ_i no es un giro perdido".

Para ser preciso, cambiamos ahora ligeramente la notación con respecto a las anteriores demostraciones: $S^{(k)}$ continúa significando el valor de la secuencia al final de la k-ésima iteración del bucle, pero renombramos las entradas de la secuencia, de modo que la i-ésima entrada se denota por $s_i^{(k)}$. La secuencia inicial es $S^{(0)} = (s_0^{(0)}, s_1^{(0)}, \ldots, s_n^{(0)})$ y, después de k puntos de inserción, la secuencia es $S^{(k)} = (s_0^{(k)}, s_1^{(k)}, \ldots, s_{n+k}^{(k)})$. También llamamos $p(s_i^{(k)})$ a la aserción "Los valores $s_i^{(k)}$ y $s_{i+1}^{(k)}$ en la secuencia $S^{(k)}$ tienen sus imágenes $\Delta(s_i^{(k)})$ y $\Delta(s_{i+1}^{(k)})$ no conectadas", y $q(s_i^{(k)})$ a la aserción "Los valores $s_i^{(k)}$ y $s_{i+1}^{(k)}$ en la secuencia $S^{(k)}$ verifican $(s_{i+1}^{(k)} - s_i^{(k)}) \geq \dfrac{|\Delta(s_i^{(k)})| + |\Delta(s_{i+1}^{(k)})|}{L}$". Con esta notación, consideremos el siguiente procedimiento (figura 2.13).

Procedimiento de inserción válido para cualquier secuencia inicial: Para encontrar el índice de una curva $\Delta : [a, b] \to \mathbb{C}$

Parámetros de entrada: La curva Δ de constante de Lipschitz L, y una secuencia $S^{(0)} = (s_0^{(0)}, s_1^{(0)}, \ldots, s_n^{(0)})$, muestreo de $[a, b]$.

Salida: Una secuencia que es válida para calcular $\mathrm{Ind}(\Delta)$.

Método:

Asignar a k el valor 0.
Mientras haya un $s_i^{(k)}$ en $S^{(k)}$ con $p(s_i^{(k)})$ o con $q(s_i^{(k)})$ hacer:
{ Insertar $\dfrac{s_i^{(k)} + s_{i+1}^{(k)}}{2}$ entre $s_i^{(k)}$ y $s_{i+1}^{(k)}$;
Incrementar k;
}
Retornar la secuencia resultante.

Figura 2.13: Procedimiento de Inserción Válido para cualquier secuencia inicial (PIV). Notemos que la línea "Insertar" produce $S^{(k)}$ a partir de $S^{(k-1)}$ en la iteración k-ésima ($k > 0$), y $s_i^{(k)} = s_i^{(k-1)}$, $s_{i+1}^{(k)} = \dfrac{s_i^{(k-1)} + s_{i+1}^{(k-1)}}{2}$, $s_{i+2}^{(k)} = s_{i+1}^{(k-1)}$.

Llamamos PIV a este Procedimiento de Inserción Válido para cualquier secuencia inicial. Supongamos que el PIV concluye tras K inserciones, retornando la secuencia $S^{(K)} = (s_0^{(K)}, s_1^{(K)}, \ldots, s_{n+K}^{(K)})$. Esta secuencia es válida para calcular $\mathrm{Ind}(\Delta)$, porque verifica, para cada $i = 0, \ldots, n + K + 1$, "no $p(s_i^{(K)})$ y no $q(s_i^{(K)})$", que es la negación de la condición del bucle. Esto implica que la secuencia $S^{(K)}$

verifica la propiedad de conexión (esto es, "no $p(s_i^{(K)})$") y que no tiene giros perdidos, porque el contrarrecíproco del lema 3 aplicado a $S^{(K)}$ es "si no $q(s_i^{(K)})$ (esto es, si $(s_{i+1}^{(K)} - s_i^{(K)}) < \dfrac{|\Delta(s_i^{(K)})| + |\Delta(s_{i+1}^{(K)})|}{L}$), entonces $\Delta([s_i^{(K)}, s_{i+1}^{(K)}])$ no es un giro perdido". Luego $S^{(K)}$ es válido para calcular $\mathrm{Ind}(\Delta)$ correctamente.

El PIV calcula el índice con cualquier secuencia inicial, mientras que el procedimiento de inserción anterior requería una secuencia que verificase una restricción en función del desconocido ε. Sin embargo, el número de iteraciones de PIV no puede deducirse del teorema 2 porque este se aplica sólo al procedimiento de inserción. Probaremos el teorema 4 más adelante, que nos da una cota al número de iteraciones de PIV. Es una prueba más complicada que la anterior por la interrelación de la propiedades p y q.

Con el cambio de notación, al final de la iteración k-ésima, la secuencia producida es $S^{(k)}$, y el último punto insertado es $\Delta(s_{i+1}^{(k)})$. Llamamos I_k al segmento de curva que une el punto de inserción de la k-ésima iteración con el punto previo de la secuencia $S^{(k)}$, y I'_k al que lo une con el siguiente. Esto es, $I_k = \Delta([s_i^{(k)}, s_{i+1}^{(k)}])$ y $I'_k = \Delta([s_{i+1}^{(k)}, s_{i+2}^{(k)}])$. Decimos que el k-ésimo punto de inserción es *decreciente* si pertenece a I_{k-1} o a I'_{k-1}. Esto es equivalente a decir que el valor de su parámetro $s_{i+1}^{(k)}$ va, en la secuencia $S^{(k)}$, inmediatamente antes o después del parámetro de la inserción $(k-1)$-ésima (ver figura 2.14). El nombre "decreciente" viene de que en tales inserciones, la diferencia de los parámetros de los extremos de I_k es la mitad de esta diferencia en I_{k-1}.

Usaremos el hecho de que, si el valor del parámetro de la inserción $(k+1)$-ésima es menor que el de la k-ésima, entonces la inserción $(k+1)$-ésima es decreciente. Para cerciorarse de este hecho, notemos que la inserción del punto $\Delta(s_{i+1}^{(k)})$ en la k-ésima iteración requiere que se haya verificado "$p(s_i^{(k-1)})$ o $q(s_i^{(k-1)})$" (la condición de entrada del bucle). Como el PIV realiza un recorrido izquierda-derecha para comprobar esta condición, también se verifica que cualquier valor en el segmento inicial de la secuencia, $(s_0^{(k-1)}, s_1^{(k-1)}, \ldots, s_{i-1}^{(k-1)})$, no cumple ni p ni q. Luego $(s_0^{(k)}, s_1^{(k)}, \ldots, s_{i-1}^{(k)}) = (s_0^{(k-1)}, s_1^{(k-1)}, \ldots, s_{i-1}^{(k-1)})$. Además $s_i^{(k)} = s_i^{(k-1)}$ por el modo de insertar el punto. Por tanto en la $(k+1)$-ésima inserción los valores del segmento inicial $(s_0^{(k)}, s_1^{(k)}, \ldots, s_{i-1}^{(k)})$ permanecen sin verificar ni p ni q, y el primer valor de la secuencia que puede verificar p o q es $s_i^{(k)}$. Luego si el valor del parámetro de la inserción $(k+1)$-ésima es menor que $s_{i+1}^{(k)}$, solo puede caer entre $s_i^{(k)}$ y $s_{i+1}^{(k)}$, los

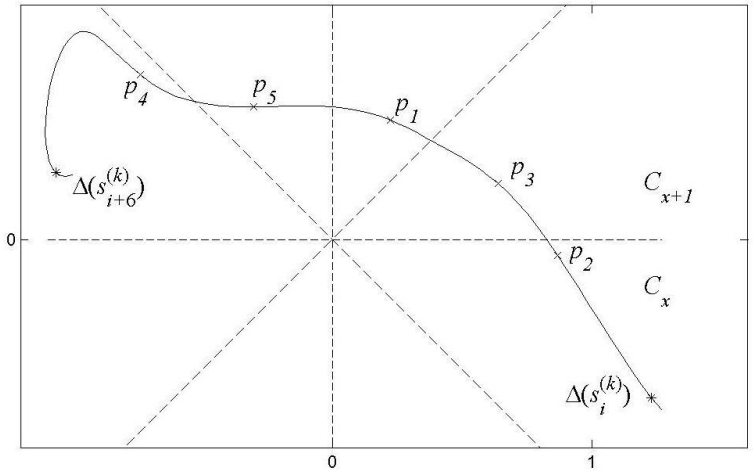


Figura 2.14: Los puntos p_1 a p_5 se han insertado en iteraciones consecutivas. Los puntos p_1, p_2, p_3 y p_5 son decrecientes pero p_4 no es decreciente.

extremos de I_k, y por tanto la inserción es decreciente.

Decimos que una iteración es una *p-inserción* si se realiza porque se verifica la propiedad $p(s_i^{(k-1)})$, y que es una *q-inserción* si se realiza porque se cumple la propiedad "no $p(s_i^{(k-1)})$ y $q(s_i^{(k-1)})$". Así cualquier iteración puede clasificarse como p-inserción o como q-inserción, pero no ambos tipos a la vez. Se verifica el siguiente hecho:

Lema 4. *Supongamos que Δ es Lipschitziana de constante L, ε-singular, y $S^{(k)}$ es la secuencia de parámetros al final de la iteración k-ésima de PIV aplicado a Δ, en la que se ha insertado el punto $\Delta(s_{i+1}^{(k)})$. Si* $\text{longarc}(I_k) \leq \dfrac{\pi\varepsilon}{4}$ *y* $\text{longarc}(I'_k) \leq \dfrac{\pi\varepsilon}{4}$, *entonces la inserción $(k+1)$-ésima no puede ser p-inserción y decreciente.*

Demostración. La proposición 3 de la sección anterior dice que la longitud de arco entre puntos no conexos es mayor que $\dfrac{\pi\varepsilon}{4}$. Contrarrecíprocamente, si $\text{longarc}(I_k) = \text{longarc}(\Delta([s_i^{(k)}, s_{i+1}^{(k)}]))$ es menor o igual que esta cantidad, los puntos $s_i^{(k)}$ y $s_{i+1}^{(k)}$ deben estar conectados. De modo similar, $s_{i+1}^{(k)}$ y $s_{i+2}^{(k)}$ están conectados también porque $I'_k = \Delta([s_{i+1}^{(k)}, s_{i+2}^{(k)}])$. Luego la siguiente iteración, si es decreciente, no puede ser una p-inserción, porque los extremos de I_k y los de I'_k están conectados.

□

La manera como se va a usar el lema anterior es para afirmar que, si se dan varias condiciones ($\text{longarc}(I_k) \leq \dfrac{\pi\varepsilon}{4}$, $\text{longarc}(I'_k) \leq \dfrac{\pi\varepsilon}{4}$ y $(s_{i+1}^{(k)} - s_i^{(k)}) <$

$\frac{|\Delta(s_i^{(k)})| + |\Delta(s_{i+1}^{(k)})|}{L}$), entonces la iteración $(k+1)$-ésima no puede ser decreciente de tipo p ni decreciente de tipo q. En consecuencia no puede ser decreciente.

Las siguiente proposiciones son pasos en la prueba de la subsiguiente proposición 9. Notemos que si llamamos $\Delta(s_j^{(k+K)})$ al $(k+K)$-ésimo punto de inserción, entonces j depende de K. Ver figura 2.15.

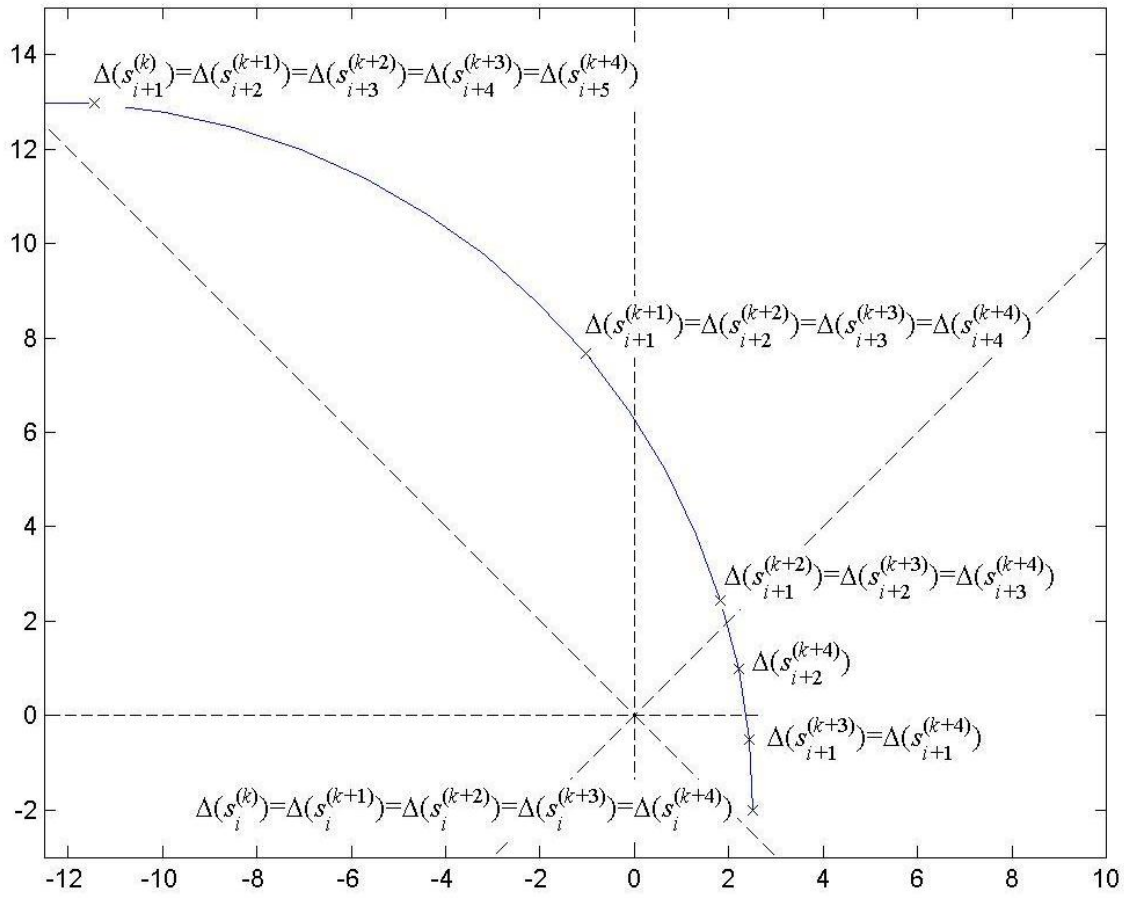


Figura 2.15: Los puntos de inserción $\Delta(s_j^{(k+K)})$ para $K = 1, 2, 3, 4$ son $\Delta(s_{i+1}^{(k+1)})$, $\Delta(s_{i+1}^{(k+2)})$, $\Delta(s_{i+1}^{(k+3)})$ y $\Delta(s_{i+2}^{(k+4)})$ respectivamente.

Proposición 7. *Supongamos que Δ es Lipschitziana de constante L, que $S^{(k)}$ es la secuencia de parámetros al final de la k-ésima iteración del PIV aplicado a Δ, y que $\Delta(s_i^{(k)}) \in C_x$ y $\Delta(s_{i+1}^{(k)}) \in (C_{x-1} \cup C_x \cup C_{x+1})^c$. Para $K = 1, 2, \ldots,$ denotemos con $p_K = \Delta(s_j^{(k+K)})$ el punto de inserción $(k+K)$-ésimo. Si los sectores C_{x-1} y C_{x+1} no contienen ningún punto de $p_1, p_2, \ldots, p_K$, entonces hay una inserción*

$(k+K+1)$-ésima, p_{K+1}, entre $\Delta(s_i^{(k)})$ y $\Delta(s_{i+1}^{(k)})$. Además si p_K es de tipo p entonces p_{K+1} es de tipo p y decreciente.

Demostración. Si C_{x-1} y C_{x+1} no contienen ningún punto de $p_1, p_2, \ldots, p_K$, tenemos que, en la secuencia $S^{(k+K)}$, las imágenes de los valores en el segmento $(s_i^{(k+K)}, s_{i+1}^{(k+K)}, \ldots, s_{i+K+1}^{(k+K)})$ no pertenecen a $C_{x-1} \cup C_{x+1}$, y los puntos extremos de este segmento, $\Delta(s_i^{(k+K)}) = \Delta(s_i^{(k)}) \in C_x$ y $\Delta(s_{i+K+1}^{(k+K)}) = \Delta(s_{i+1}^{(k)}) \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, no están conectados. Luego al menos uno de los $\Delta(s_h^{(k+K)})$ con $i \leq h < i+K+1$ debe verificar $p(s_h^{(k+K)})$, por tanto se verifica al menos una condición suficiente para provocar una inserción $(k+K+1)$-ésima.

Para ver que la inserción p_{K+1} es decreciente si p_K es de tipo p, consideremos los puntos $q_1 = \Delta(s_{j-1}^{(k+K-1)})$ y $q_2 = \Delta(s_j^{(k+K-1)})$ que causaron la inserción $p_K = \Delta(s_j^{(k+K)})$. Estos son los primeros puntos encontrados que verifican o bien "q_1 y q_2 están no conectados" (esto es $p(s_{j-1}^{(k+K-1)})$), o bien "$(s_j^{(k+K-1)} - s_{j-1}^{(k+K-1)}) \geq \dfrac{|\Delta(s_{j-1}^{(k+K-1)})| + |\Delta(s_j^{(k+K-1)})|}{L}$" (esto es $q(s_{j-1}^{(k+K-1)})$). Además, debido al escaneo izquierda-derecha que hace el PIV, se verifica que los puntos anteriores $(s_i^{(k+K-1)}, s_{i+1}^{(k+K-1)}, \ldots, s_{j-3}^{(k+K-1)}, s_{j-2}^{(k+K-1)})$ en la secuencia $S^{(k+K-1)}$ verifican "no p y no q", porque en caso contrario el punto de inserción $(k+K)$-ésimo tendría índice menor que j, contradiciendo que $p_K = \Delta(s_j^{(k+K)})$. Como los puntos del segmento $(\Delta(s_{i+1}^{(k+K-1)}), \Delta(s_{i+2}^{(k+K-1)}), \ldots, \Delta(s_{j-1}^{(k+K-1)}) = q_1)$ son todos del conjunto $\{p_1, p_2, \ldots, p_K\}$ (que está fuera de $C_{x-1} \cup C_{x+1}$), y $\Delta(s_i^{(k+K-1)}) = \Delta(s_i^{(k)}) \in C_x$, y se cumple "no $p(s_h^{(k+K-1)})$" para $i \leq h < j-1$, deducimos que todos estos puntos pertenecen a C_x, en particular $q_1 \in C_x$. Luego, como p_K es de tipo p (esto es, $p(s_{j-1}^{(k+K-1)})$), entonces $q_2 \in (C_{x-1} \cup C_x \cup C_{x+1})^c$. Finalmente, el punto p_K pertenece a C_x o a $(C_{x-1} \cup C_x \cup C_{x+1})^c$ (por hipótesis), luego la $(k+K+1)$-ésima inserción se realiza en $S^{(k+K)} = (\ldots, q_1, p_K, q_2, \ldots)$ entre p_K, q_2 (si $p_K \in C_x$) o entre q_1, p_K (si $p_K \in (C_{x-1} \cup C_x \cup C_{x+1})^c$). En cualquier caso es de tipo p y decreciente.

□

Notemos que por la proposición anterior, los puntos de inserción sucesivos entre $s_i^{(k)}$ y $s_{i+1}^{(k)}$ son de tipo p hasta que uno de ellos pertenezca a $C_{x-1} \cup C_{x+1}$, después de lo cual pueden venir uno o varios de tipo q.

Proposición 8. *Supongamos que Δ es Lipschitziana de constante L, ε-singular, y que $S^{(k)}$ es la secuencia de parámetros al final de la iteración k-ésima del PIV apli-*

cado a Δ. *Llamando* $M = (s_{i+1}^{(k)} - s_i^{(k)})$, *si el intervalo* $[s_i^{(k)}, s_{i+1}^{(k)}]$ *verifica* $\Delta(s_i^{(k)}) \in C_x$ *y* $\Delta(s_{i+1}^{(k)}) \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, *entonces para algún* $K \leq \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$ *el* $(k+K)$*-ésimo punto de inserción* p_K *verifica* $p_K \in C_{x-1} \cup C_{x+1}$.

Demostración. Definamos k_0 como el entero verificando $\frac{LM}{2^{k_0}} \leq \frac{\pi}{4}\varepsilon \leq \frac{LM}{2^{k_0-1}}$, esto es $k_0 = \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$. En general, no es posible tener $k_0 + 1$ p-inserciones decrecientes consecutivas. Esto es porque empezamos con un segmento $I_k = \Delta([s_i^{(k)}, s_{i+1}^{(k)}])$, cuya longitud de arco es menor que LM con $M = (s_{i+1}^{(k)} - s_i^{(k)})$, y después de k_0 inserciones decrecientes, llamando $s_j^{(k+k_0)}$ al parámetro del $(k+k_0)$-ésimo punto de inserción $p_{k_0} = \Delta(s_j^{(k+k_0)})$, tenemos que la diferencia de parámetros de los extremos de $I_{k+k_0} = \Delta([s_{j-1}^{(k+k_0)}, s_j^{(k+k_0)}])$ es $(s_j^{(k+k_0)} - s_{j-1}^{(k+k_0)}) = \frac{M}{2^{k_0}}$, y $\text{longarc}(I_{k+k_0}) \leq \frac{LM}{2^{k_0}}$, que es menor o igual que $\frac{\pi\varepsilon}{4}$ por definición de k_0. Luego por el lema 4 la iteración $(k+k_0+1)$-ésima, si es de tipo p, no puede ser una inserción decreciente: debe ser no decreciente.

Notemos que si para algún K, menor o igual que k_0, se tiene que $p_K \in C_{x-1} \cup C_{x+1}$, concluimos porque es la afirmación de la proposición. En caso contrario alcanzamos una contradicción, porque entonces para cada K de 1 a k_0, $p_K \notin C_{x-1} \cup C_{x+1}$. Luego podemos aplicar la proposición 7 para $K = 1$ (Esto es, usando la hipótesis de que $p_1 \notin C_{x-1} \cup C_{x+1}$), porque p_1 es de tipo p por la hipótesis "$\Delta(s_i^{(k)}) \in C_x$ y $\Delta(s_{i+1}^{(k)}) \in (C_{x-1} \cup C_x \cup C_{x+1})^c$", para deducir que p_2 es decreciente y de tipo p; También la aplicamos para $K = 2$ (sabiendo que $p_2 \notin C_{x-1} \cup C_{x+1}$ y de tipo p) para concluir que p_3 es decreciente y de tipo p, y así sucesivamente para $K = 1, 2, \ldots, k_0$, concluyendo que las inserciones $(k+K)$-ésimas p_K (para $K = 2, 3, \ldots, k_0+1$) son decrecientes y de tipo p. Esto es una contradicción con la observación anterior, de que la iteración $(k+k_0+1)$-ésima no puede ser decreciente y de tipo p.

□

La siguiente proposición juega un papel similar al de la proposición 6 de la sección anterior. Mientras que una nos daba una cota del número de inserciones que pueden hacerse hasta que se inserta un punto en un sector conectado con el de $\Delta(s_i)$, la proposición 9 nos da una cota para el número de inserciones hasta que un punto verifica no p y no q. Ambas proposiciones sirven como base para

lemas posteriores que acotan el números de inserciones que realizan el proceso de inserción y el PIV entre dos valores del parámetro.

El número de inserciones que requiere el PIV es mayor que el del procedimiento de inserción porque una inserción de tipo q, aún en las hipótesis de la proposición 7, puede ser no decreciente. Sin embargo, el número de q-inserciones verifica lo siguiente: si Δ es ε-singular entonces $\frac{|\Delta(s_i^{(k)})| + |\Delta(s_{i+1}^{(k)})|}{L} \geq \frac{2\varepsilon}{L}$ porque $|\Delta(t)| \geq \varepsilon$ para cualquier parámetro t. Por tanto, si se verifica $q(s_i^{(k)})$ (esto es, $(s_{i+1}^{(k)} - s_i^{(k)}) \geq \frac{|\Delta(s_i^{(k)})| + |\Delta(s_{i+1}^{(k)})|}{L}$, que es mayor o igual que $\frac{2\varepsilon}{L}$), entonces el punto insertado tiene un valor de parámetro de $s_{i+1}^{(k+1)} = \frac{s_i^{(k)} + s_{i+1}^{(k)}}{2}$ que verifica $(s_{i+1}^{(k+1)} - s_i^{(k)}) = \frac{s_{i+1}^{(k)} - s_i^{(k)}}{2} \geq \frac{|\Delta(s_i^{(k)})| + |\Delta(s_{i+1}^{(k)})|}{2L} \geq \frac{\varepsilon}{L}$ y $(s_{i+1}^{(k)} - s_{i+1}^{(k+1)}) = \frac{s_{i+1}^{(k)} - s_i^{(k)}}{2} \geq \frac{\varepsilon}{L}$. Esto es, $s_{i+1}^{(k+1)}$ está situado a una distancia mayor que $\frac{\varepsilon}{L}$ de $s_i^{(k)}$ y de $s_{i+1}^{(k)}$. Repitiendo el razonamiento, x inserciones de tipo q se extienden sobre una longitud mayor o igual que $(x-1)\frac{\varepsilon}{L}$. Dentro de un segmento de parámetros de longitud M (y a una distancia mayor que $\frac{\varepsilon}{L}$ de sus dos extremos) el número x de q-inserciones debe verificar $(x-1)\frac{\varepsilon}{L} + 2\frac{\varepsilon}{L} = (x+1)\frac{\varepsilon}{L} \leq M$. Esto es, no puede haber más de $\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor$ q-inserciones, si esta expresión es mayor o igual que 0, o ninguna q-inserción en otro caso.

Proposición 9. *Supongamos que Δ es Lipschitziana de constante L, ε-singular, y que $S^{(k)}$ es la secuencia de parámetros al final de la k-ésima iteración del PIV aplicado a Δ. Llamando $M = (s_{i+1}^{(k)} - s_i^{(k)})$, si $\Delta(s_i^{(k)}) \in C_x$, $\Delta(s_{i+1}^{(k)}) \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, y x al número de inserciones de tipo q que hay en el intervalo $[s_i^{(k)}, s_{i+1}^{(k)}]$, entonces para algún $K' \leq (x+1)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + x$ hay un punto $s_g^{(k+K')}$ de la secuencia $S^{(k+K')}$ verificando que $\Delta(s_g^{(k+K')}) \in C_{x-1} \cup C_{x+1}$, y que para cada h con $i \leq h < g$ no se verifica ni $p(s_h^{(k+K')})$ ni $q(s_h^{(k+K')})$.*

Demostración. El enunciado es más complicado que el de la proposición 6 porque el punto de interés $s_g^{(k+K')}$ no es necesariamente el punto $p_{K'}$ insertado en la iteración $(k+K')$-ésima. Probaremos la afirmación por inducción completa en x.

Con $x = 0$, el caso base, tenemos, por la proposición 8, que para algún $K \leq \left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil$ el $(k+K)$-ésimo punto de inserción $p_K = \Delta(s_j^{(k+K)})$ verifica $p_K \in$

$C_{x-1} \cup C_{x+1}$. Podemos suponer que este es el primer punto insertado perteneciente a $C_{x-1} \cup C_{x+1}$, porque un punto de inserción anterior perteneciente a esta región también tendría su subíndice acotado por $\left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$. No se verifica $p(s_h^{(k+K-1)})$ para $i \leq h < j-1$, porque si así fuese la inserción $(k+K)$-ésima no tendría un parámetro con índice j. Notemos que, para $i \leq h < j-1$, son equivalentes "no $p(s_h^{(k+K-1)})$" y "no $p(s_h^{(k+K)})$", porque los puntos implicados en estas aserciones en $S^{(k+K-1)}$ y en $S^{(k+K)}$ son los mismos. Luego los puntos imagen del segmento inicial $(s_i^{(k+K)}, s_{i+1}^{(k+K)}, \ldots, s_{j-2}^{(k+K)}, s_{j-1}^{(k+K)})$ de la secuencia $S^{(k+K)}$ deben pertenecer a C_x, porque están conectados ("no $p(s_h^{(k+K)})$"), y $p_K = \Delta(s_j^{(k+K)})$ es el primer punto de inserción perteneciente a $C_{x-1} \cup C_{x+1}$. Además, como p_K pertenece a $C_{x-1} \cup C_{x+1}$, no se verifica $p(s_h^{(k+K)})$ para $h = j-1$. Tampoco se verifica $q(s_j^{(k+K)})$ para $i \leq h < j$ porque en ese caso tendríamos una q-inserción, y eso no es posible con $x = 0$. Luego llegamos a la afirmación con $K' = K \leq \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$ y $g = j$.

Para $x > 0$, el caso general, supongamos que la primera q-inserción es la $(k+T)$-ésima, y que y es el número de q-inserciones consecutivas que se producen después de esta, incluyéndola. Esto significa que p_T es la primera q-inserción, y también que $p_{T+1}, p_{T+2}, \ldots$ hasta p_{T+y-1} son de tipo q, pero p_{T+y} (si existe) es una p-inserción. Por la proposición 8, sabemos que en no más de $K \leq \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$ inserciones, $p_K = \Delta(s_j^{(k+K)}) \in C_{x-1} \cup C_{x+1}$. Se cumple que $T = K+1$, por el siguiente razonamiento: como esta inserción, la $(k+K)$-ésima, tiene lugar en la j-ésima entrada de la sucesión, entonces no se verifica ni $p(s_h^{(k+K-1)})$ ni $q(s_h^{(k+K-1)})$ para $i \leq h < j-1$ (esto es equivalente a ni $p(s_h^{(k+K)})$ ni $q(s_h^{(k+K)})$ para $i \leq h < j-1$). Como $\Delta(s_j^{(k+K)}) \in C_{x-1} \cup C_{x+1}$, tampoco se verifica $p(s_{j-1}^{(k+K)})$. Si además tampoco se verificase $q(s_{j-1}^{(k+K)})$, concluiríamos con la afirmación deseada: "$\Delta(s_g^{(k+K')}) \in C_{x-1} \cup C_{x+1}$ y para cada h con $i \leq h < g$ no se verifica ni $p(s_h^{(k+K')})$ ni $q(s_h^{(k+K')})$" con $K' = K \leq \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil$ y $g = j$, como en el caso base. Falta considerar el caso de que sí se verificase $q(s_{j-1}^{(k+K)})$. Pero entonces la inserción $(k+K+1)$-ésima es de tipo q, y $T = K+1$.

Por tanto hay y inserciones consecutivas de tipo q, p_{K+1} a p_{K+y}, pero p_{K+y+1} (si existe) es de tipo p. Llamemos $K_1 = K+y$, y $s_f^{(k+K_1)}$ al valor del parámetro de la última de esta racha inicial de q-inserciones consecutivas: $p_{K_1} = \Delta(s_f^{(k+K_1)})$. Como esta última inserción, la $(k+K_1)$-ésima, tiene lugar en la entrada f-ésima

de la secuencia, no se verifica ni $p(s_h^{(k+K_1-1)})$ ni $q(s_h^{(k+K_1-1)})$ para $i \leq h < f-1$ (equivalentemente, ni $p(s_h^{(k+K_1)})$ ni $q(s_h^{(k+K_1)})$ para $i \leq h < f-1$). Además, como la siguiente inserción $p_{K_1+1} = p_{K+y+1}$ no es de tipo q (o no existe), no puede verificarse $q(s_{f-1}^{(k+K_1)})$. Resumiendo, tenemos "no $p(s_h^{(k+K_1)})$" para $i \leq h < f-1$, y "no $q(s_h^{(k+K_1)})$" para $i \leq h \leq f-1$. Este hecho será usado varias veces más adelante.

Ahora discutimos dos casos diferentes. Consideremos primero que algún valor del segmento inicial $(s_i^{(k+K_1)}, s_{i+1}^{(k+K_1)}, \ldots, s_{f-2}^{(k+K_1)}, s_{f-1}^{(k+K_1)})$ de la secuencia $S^{(k+K_1)}$ es tal que su imagen no pertenece a C_x. Llamemos g al menor índice de estos valores, es decir, el primer índice g con $i \leq g \leq f-1$ y que la imagen $\Delta(s_g^{(k+K_1)})$ no pertenezca a C_x. Debe verificar $\Delta(s_g^{(k+K_1)}) \in C_{x-1} \cup C_{x+1}$, porque tenemos "no $p(s_h^{(k+K_1)})$" para $i \leq h < f-1$, luego las imágenes de $(s_i^{(k+K_1)}, s_{i+1}^{(k+K_1)}, \ldots, s_{g-2}^{(k+K_1)}, s_{g-1}^{(k+K_1)})$ deben estar en sectores conectados, y $\Delta(s_g^{(k+K_1)})$ es la primera que no está en C_x. Y para cada h con $i \leq h < g$ no se verifica ni $p(s_h^{(k+K_1)})$ ni $q(s_h^{(k+K_1)})$ (por el hecho mostrado anteriormente "no $p(s_h^{(k+K_1)})$" y "no $q(s_h^{(k+K_1)})$" para $i \leq h < f-1$, y porque $g \leq f-1$). Como $y \leq x$, tenemos un punto $\Delta(s_g^{(k+K_1)})$ que satisface la afirmación deseada con $K' = K_1 = K + y \leq \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil + x \leq (x+1) \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil + x$.

Consideremos ahora el caso alternativo, que todos los puntos imagen del segmento inicial $(s_i^{(k+K_1)}, s_{i+1}^{(k+K_1)}, \ldots, s_{f-2}^{(k+K_1)}, s_{f-1}^{(k+K_1)})$ de la secuencia $S^{(k+K_1)}$ pertenecen a C_x. En particular $\Delta(s_{f-1}^{(k+K_1)}) \in C_x$. Para $p_{K_1} = \Delta(s_f^{(k+K_1)})$ hay tres posibles opciones: $p_{K_1} \in C_x$, $p_{K_1} \in C_{x-1} \cup C_{x+1}$, o $p_{K_1} \in (C_{x-1} \cup C_x \cup C_{x+1})^c$.

En la primera opción, como $p_{K_1} = \Delta(s_f^{(k+K_1)}) \in C_x$ y $\Delta(s_{i+K_1+1}^{(k+K_1)}) = \Delta(s_{i+1}^{(k)}) \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, y llamando $M_2 = (s_{i+1}^{(k)} - s_f^{(k+K_1)})$, se verifica la hipótesis de inducción en el intervalo $[s_f^{(k+K_1)}, s_{i+1}^{(k)}]$, que solo puede tener $x-y$ inserciones de tipo q, y deducimos que en $K_2 \leq (x-y+1) \left\lceil \lg_2 \left(\frac{4LM_2}{\pi\varepsilon} \right) \right\rceil + (x-y)$ inserciones, tenemos cierto punto $\Delta(s_{g_2}^{(k+K_1+K_2)}) \in C_{x-1} \cup C_{x+1}$ tal que para cada h con $f \leq h < g_2$ no se verifica ni $p(s_h^{(k+K_1+K_2)})$ ni $q(s_h^{(k+K_1+K_2)})$. Luego concluimos con la afirmación deseada tomando $K' = K_1 + K_2$ y $g = g_2$, porque como $M_2 \leq M$ y $x - y + 1 \leq x$, tenemos que $K_1 + K_2 = K + y + K_2 \leq \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil + y + (x-y+1) \left\lceil \lg_2 \left(\frac{4LM_2}{\pi\varepsilon} \right) \right\rceil + (x-y) \leq (x+1) \left\lceil \lg_2 \left(\frac{4LM}{\pi\varepsilon} \right) \right\rceil + x$. También tenemos que "no $p(s_h^{(k+K_1+K_2-1)})$ y

no $q(s_h^{(k+K_1+K_2-1)})$" para $i \leq h \leq f-1$ (porque en caso contrario no alcanzaríamos la inserción $\Delta(s_{g_2}^{(k+K_1+K_2)})$ de índice g_2 con $f < g_2$), y $s_h^{(k+K_1+K_2-1)} = s_h^{(k+K_1+K_2)}$ para $i \leq h \leq f-1$. Reuniendo $i \leq h \leq f-1$ con $f \leq h < g_2$, tenemos que ni $p(s_h^{(k+K_1+K_2)})$ ni $q(s_h^{(k+K_1+K_2)})$ para cada h con $i \leq h < g_2$.

En la segunda opción $p_{K_1} \in C_{x-1} \cup C_{x+1}$ también obtenemos lo deseado tomando $g = f$ y $K' = K_1 = K + y \leq (x+1)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + x$, porque $\Delta(s_f^{(k+K_1)})$ pertenece a $C_{x-1} \cup C_{x+1}$, y por el hecho mostrado anteriormente "no $p(s_h^{(k+K_1)})$ y no $q(s_h^{(k+K_1)})$" para $i \leq h < f$.

Finalmente, en la tercera opción, que es $p_{K_1} \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, como $\Delta(s_{f-1}^{(k+K_1)}) \in C_x$ y $\Delta(s_f^{(k+K_1)}) \in (C_{x-1} \cup C_x \cup C_{x+1})^c$, llamando $M_2 = (s_f^{(k+K_1)} - s_{f-1}^{(k+K_1)})$, el intervalo $[s_{f-1}^{(k+K_1)}, s_f^{(k)}]$ verifica la hipótesis de inducción, y solo puede tener $(x-y)$ inserciones de tipo q, y por tanto deducimos que en menos de $K_2 \leq (x-y+1)\left\lceil \lg_2\left(\frac{4LM_2}{\pi\varepsilon}\right)\right\rceil + (x-y)$ inserciones, se obtiene cierto punto $\Delta(s_{g_2}^{(k+K_1+K_2)}) \in C_{x-1} \cup C_{x+1}$ tal que para cada h with $f-1 \leq h < g_2$ no se verifica ni $p(s_h^{(k+K_1+K_2)})$ ni $q(s_h^{(k+K_1+K_2)})$. De modo parecido a lo anterior, concluimos tomando $K' = K_1 + K_2 = K + y + K_2 \leq \left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + y + (x - y + 1)\left\lceil \lg_2\left(\frac{4LM_2}{\pi\varepsilon}\right)\right\rceil + (x-y) \leq (x+1)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + x$ y tenemos que "no $p(s_h^{(k+K_1+K_2)})$ y no $q(s_h^{(k+K_1+K_2)})$" para $i \leq h < g_2$.

□

Esta laboriosa proposición nos permite demostrar el siguiente lema. Denotaremos $\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0 = \text{máx}\left(0, \left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor\right)$.

Lema 5. *Supongamos que Δ es Lipschitziana de constante L, ε-singular, y que $S^{(k)}$ es la secuencia de parámetros al final de la iteración k-ésima del PIV aplicado a Δ, y que $M = (s_{i+1}^{(k)} - s_i^{(k)})$ siendo $\Delta(s_i^{(k)})$ el k-ésimo punto de inserción. Si N es el número de bordes cruzados por $\Delta([s_i^{(k)}, s_{i+1}^{(k)}])$, para $N \geq 2$ el número de puntos de inserción entre $\Delta(s_i^{(k)})$ y $\Delta(s_{i+1}^{(k)})$ está acotado por*

$$(N-1)\left(\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0 + 1\right)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + N\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0.$$

Demostración. Probamos la afirmación por inducción en N. Si $N = 2$, por la

proposición 9 en K' inserciones, con $K' \leq (x+1)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + x$, siendo x el número de q-inserciones, tenemos un punto de inserción $\Delta(s_g^{(k+K')}) \in C_{x-1} \cup C_{x+1}$ tal que para cada h con $i \leq h < g$ no se verifica ni $p(s_h^{(k+K')})$ ni $q(s_h^{(k+K')})$. Recordemos que el número de q-inserciones está acotado por $\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0$, así que $K' \leq \left(\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0 + 1\right)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + \left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0$. Como se verifica "ni p ni q", el PIV no producirá ninguna inserción adicional en el segmento de curva $\Delta([s_i^{(k)}, s_g^{(k+K')}])$. Además no producirá ninguna p-inserción en $\Delta([s_g^{(k+K')}, s_{i+1}^{(k)}])$ (porque los puntos extremos están conectados ya que $N = 2$). El número de q-inserciones en $\Delta([s_g^{(k+K)}, s_{i+1}^{(k)}])$ está acotado por $\left\lfloor \frac{L(s_{i+1}^{(k)} - s_g^{(k+K)})}{\varepsilon} - 1 \right\rfloor_0 \leq \left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0$, luego el total de inserciones en $\Delta([s_i^{(k)}, s_{i+1}^{(k)}])$ es menor o igual que

$$K' + \left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0 \leq \left(\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0 + 1\right)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + 2\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0,$$

que es nuestra afirmación.

Para el caso general $N > 2$, la hipótesis de inducción es que cualquier segmento de curva, con una diferencia de parámetros de los extremos de M', cruzando $(N-1)$ bordes, requiere como mucho

$$(N-2)\left(\left\lfloor \frac{LM'}{\varepsilon} - 1 \right\rfloor_0 + 1\right)\left\lceil \lg_2\left(\frac{4LM'}{\pi\varepsilon}\right)\right\rceil + (N-1)\left\lfloor \frac{LM'}{\varepsilon} - 1 \right\rfloor_0$$

inserciones. Por la proposición 9, tenemos que para cierto K' menor o igual que $K' \leq (x+1)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + x \leq \left(\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0 + 1\right)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + \left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0$, el punto de inserción $\Delta(s_g^{(k+K')}) \in C_{x-1} \cup C_{x+1}$ es tal que para cada h con $i \leq h < g$ no se verifica ni $p(s_h^{(k+K')})$ ni $q(s_h^{(k+K')})$. Luego el segmento $\Delta([s_i^{(k)}, s_g^{(k+K')}])$ no tendrá más inserciones, y $\Delta([s_g^{(k+K')}, s_{i+1}^{(k)}])$, que tiene una diferencia entre los parámetros de los extremos de $M' = (s_{i+1}^{(k)} - s_g^{(k+K')}) < M$, por hipótesis de inducción requerirá como mucho de $(N-2)\left(\left\lfloor \frac{LM'}{\varepsilon} - 1 \right\rfloor_0 + 1\right)\left\lceil \lg_2\left(\frac{4LM'}{\pi\varepsilon}\right)\right\rceil + (N-1)\left\lfloor \frac{LM'}{\varepsilon} - 1 \right\rfloor_0 \leq (N-2)\left(\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0 + 1\right)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + (N-1)\left\lfloor \frac{LM}{\varepsilon} - 1 \right\rfloor_0$ inserciones. Sumando las cotas de ambos segmentos, tenemos que el total es menor

o igual que $(N-1)\left(\left\lfloor \frac{LM}{\varepsilon}-1\right\rfloor_0+1\right)\left\lceil \lg_2\left(\frac{4LM}{\pi\varepsilon}\right)\right\rceil + N\left\lfloor \frac{LM}{\varepsilon}-1\right\rfloor_0$.

□

Finalmente tenemos:

Teorema 4. *Si $\Delta:[a,b]\to\mathbb{C}$ es ε-singular con $\varepsilon\neq 0$, con una parametrización Lipschitziana de constante L, entonces el PIV aplicado a la curva Δ concluye en menos de*

$$\frac{4L(b-a)}{\pi\varepsilon}\left(\left\lfloor \frac{L|S^{(0)}|}{\varepsilon}-1\right\rfloor_0+1\right)\left\lceil \lg_2\left(\frac{4L|S^{(0)}|}{\pi\varepsilon}\right)\right\rceil+\left(\frac{4L|S^{(0)}|}{\pi\varepsilon}+1\right)\left\lfloor \frac{L(b-a)}{\varepsilon}-1\right\rfloor_0$$

inserciones.

Demostración. El número N de bordes cruzados por cada segmento $\Delta([s_i^{(0)},s_{i+1}^{(0)}])$ de la secuencia inicial es tal que $N-1\leq \frac{4L(s_{i+1}^{(0)}-s_i^{(0)})}{\pi\varepsilon}$, como se ve en la prueba del teorema 2. Aplicando el lema previo tenemos que el número máximo de inserciones en cada segmento es $\frac{4L(s_{i+1}^{(0)}-s_i^{(0)})}{\pi\varepsilon}\left(\left\lfloor \frac{L(s_{i+1}^{(0)}-s_i^{(0)})}{\varepsilon}-1\right\rfloor_0+1\right)\left\lceil \lg_2\left(\frac{4L(s_{i+1}^{(0)}-s_i^{(0)})}{\pi\varepsilon}\right)\right\rceil+\left(\frac{4L(s_{i+1}^{(0)}-s_i^{(0)})}{\pi\varepsilon}+1\right)\left\lfloor \frac{L(s_{i+1}^{(0)}-s_i^{(0)})}{\varepsilon}-1\right\rfloor_0$. Sumando estos valores, y usando que $\left\lfloor\sum_{i=0}^{n-1}\frac{4L(s_{i+1}^{(0)}-s_i^{(0)})}{\varepsilon}\right\rfloor=\left\lfloor\frac{4L(b-a)}{\varepsilon}\right\rfloor$, $\left\lfloor \frac{L(s_{i+1}^{(0)}-s_i^{(0)})}{\varepsilon}-1\right\rfloor_0\leq\left\lfloor \frac{L|S^{(0)}|}{\varepsilon}-1\right\rfloor_0$ y $\lg_2\left(\frac{4L(s_{i+1}^{(0)}-s_i^{(0)})}{\pi\varepsilon}\right)\leq\lg_2\left(\frac{4L|S^{(0)}|}{\pi\varepsilon}\right)$, para los factores del primer sumando, y dos desigualdades similares para los factores del segundo sumando, se tiene que el total de inserciones es menor o igual que $\frac{4L(b-a)}{\pi\varepsilon}\left(\left\lfloor \frac{L|S^{(0)}|}{\varepsilon}-1\right\rfloor_0+1\right)\left\lceil \lg_2\left(\frac{4L|S^{(0)}|}{\pi\varepsilon}\right)\right\rceil++\left(\frac{4L|S^{(0)}|}{\pi\varepsilon}+1\right)\left\lfloor \frac{L(b-a)}{\varepsilon}-1\right\rfloor_0$.

□

Notemos que esta cota del número de inserciones requeridas por el PIV para el cálculo del índice es de orden $O\left(\frac{1}{\varepsilon^2}\lg_2\left(\frac{1}{\varepsilon}\right)+\frac{1}{\varepsilon}\lg_2\left(\frac{1}{\varepsilon}\right)\right)=O\left(\frac{1}{\varepsilon^2}\lg_2\left(\frac{1}{\varepsilon}\right)\right)$.

2.6. Cota del coste independientemente de la ε-singularidad

Al usar el PIV, un cálculo del índice requiere $\frac{4L(b-a)}{\pi\varepsilon}\left(\left\lfloor\frac{L|S^{(0)}|}{\varepsilon}-1\right\rfloor_0+1\right)\left\lceil\lg_2\left(\frac{4L|S^{(0)}|}{\pi\varepsilon}\right)\right\rceil+\left(\frac{4L|S^{(0)}|}{\pi\varepsilon}+1\right)\left\lfloor\frac{L(b-a)}{\varepsilon}-1\right\rfloor_0$ inserciones. Sin embargo, si ε no es previamente conocido, esta fórmula no puede aplicarse para presupuestar el número de inserciones que se necesitarán. Puede ser arbitrariamente alto si la distancia de la curva Δ al origen es cercana a cero. El PIV, aplicado a una curva con valor de singularidad ε desconocido, se enfrenta a un coste impredecible. Para controlar esto, modificamos el PIV, de tal modo que podemos acotar el coste del cálculo del índice, retornando con error cuando se excede esta cota. La herramienta usada es el teorema 5 más adelante. Está basado en el hecho de que dos puntos de inserción con valores de parámetro cercanos implican un valor de singularidad bajo. Primero probamos este hecho para p-inserciones, en el lema 6, y luego para q-inserciones, en el lema 7.

Lema 6. *Si s_i, s_{i+1}, s_{i+2} son tres valores del parámetro de la curva Δ, Lipschitziana con constante L, ε-singular, y s_{i+1} verifica que $s_{i+1}=\frac{s_i+s_{i+2}}{2}$ con $\Delta(s_i)$ y $\Delta(s_{i+2})$ en sectores no conectados, y además $s_{i+1}-s_i\leq\delta$ para cierto δ positivo, entonces o bien $|\Delta(s_i)|$ o bien $|\Delta(s_{i+2})|$ es menor o igual que $\frac{L\delta}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$. Consecuentemente $\varepsilon\leq\frac{L\delta}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$.*

Demostración. Como $s_{i+1}=\frac{s_i+s_{i+2}}{2}$ entonces $s_{i+2}-s_{i+1}=s_{i+1}-s_i$ y por hipótesis $s_{i+2}-s_i=s_{i+2}-s_{i+1}+s_{i+1}-s_i=2(s_{i+1}-s_i)\leq 2\delta$. Además por la propiedad de Lipschitz, $|\Delta(s_{i+2})-\Delta(s_i)|\leq L(s_{i+2}-s_i)\leq L2\delta$. Luego se tiene que los puntos $\Delta(s_i)$ y $\Delta(s_{i+2})$ están en sectores no conectados, pero a una distancia menor que $L2\delta$. Consideremos el triángulo formado por estos puntos y el origen O. Tiene un ángulo α en O mayor que $\frac{\pi}{4}$, pero menor o igual que π porque es el ángulo interno de un triángulo. El lado opuesto a α es el segmento $\overline{\Delta(s_i)\Delta(s_{i+2})}$, de longitud menor o igual que $L2\delta$ (ver figura 2.16). Mostraremos que o bien $\Delta(s_i)$ o bien $\Delta(s_{i+2})$ está a distancia menor o igual que $\frac{L\delta}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$ del origen. Supongamos

que $|\Delta(s_{i+2})| \leq |\Delta(s_i)|$ (en caso de que $|\Delta(s_{i+2})| > |\Delta(s_i)|$, el razonamiento es similar). Sea D el punto del segmento $\overline{O\Delta(s_i)}$ a la misma distancia de O que $\Delta(s_{i+2})$.

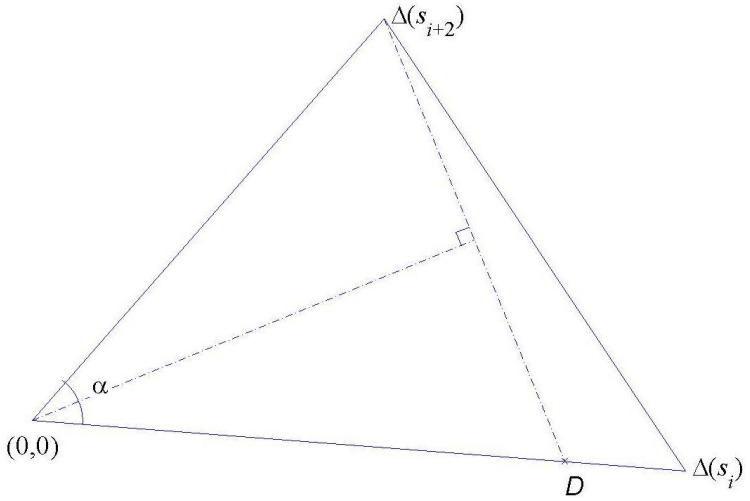


Figura 2.16: El punto D está a la misma distancia del origen que $\Delta(s_{i+2})$.

Se verifica que $\text{long}(\overline{\Delta(s_{i+2})D}) \leq L2\delta$, porque el triángulo isósceles de vértices O, $\Delta(s_{i+2})$, D tiene el lado mínimo entre aquellos triángulos que tienen un ángulo de α y el menor lado adyacente a este tiene longitud $|\Delta(s_{i+2})|$. Además, considerando el triángulo que surge de la bisección del ángulo α, tenemos que $\text{sen}\left(\frac{\alpha}{2}\right) = \frac{\frac{\text{long}(\overline{\Delta(s_{i+2})D})}{2}}{|\Delta(s_{i+2})|}$. Finalmente notemos que como $\frac{\pi}{4} \leq \alpha \leq \pi$, entonces $\frac{\pi}{8} \leq \frac{\alpha}{2} \leq \frac{\pi}{2}$ y por tanto $\frac{\pi}{8}$ y $\frac{\alpha}{2}$ están en un intervalo de crecimiento de la función seno, y verifican $\text{sen}\left(\frac{\pi}{8}\right) \leq \text{sen}\left(\frac{\alpha}{2}\right)$. Encadenando estas desigualdades tenemos:

$$\text{sen}\left(\frac{\pi}{8}\right) \leq \text{sen}\left(\frac{\alpha}{2}\right) = \frac{\frac{\text{long}(\overline{\Delta(s_{i+2})D})}{2}}{|\Delta(s_{i+2})|} \leq \frac{L\delta}{|\Delta(s_{i+2})|}$$

Y entonces $|\Delta(s_{i+2})| \leq \frac{L\delta}{\text{sen}\left(\frac{\pi}{8}\right)}$. Como ε es el mínimo $|\Delta(s)|$ con $s \in [a,b]$, tenemos que $\varepsilon \leq |\Delta(s_{i+2})| \leq \frac{L\delta}{\text{sen}\left(\frac{\pi}{8}\right)}$.

□

Lema 7. *Si s_i, s_{i+1}, s_{i+2} son tres valores del parámetro de la curva Δ, Lipschitziana con constante L, ε-singular, y s_{i+1} verifica que $s_{i+1} = \dfrac{s_i + s_{i+2}}{2}$ con $(s_{i+2} - s_i) \geq \dfrac{|\Delta(s_i)| + |\Delta(s_{i+2})|}{L}$, y además $s_{i+1} - s_i \leq \delta$ para cierto δ positivo, entonces o bien $|\Delta(s_i)|$ o bien $|\Delta(s_{i+2})|$ es menor o igual que $L\delta$. Consecuentemente $\varepsilon \leq L\delta$.*

Demostración. Como $s_{i+1} = \dfrac{s_i + s_{i+2}}{2}$ entonces $s_{i+2} - s_{i+1} = s_{i+1} - s_i$ y por hipótesis, como en el lema 6, $s_{i+2} - s_i = s_{i+2} - s_{i+1} + s_{i+1} - s_i = 2(s_{i+1} - s_i) \leq 2\delta$. Encadenando esto con $(s_{i+2} - s_i) \geq \dfrac{|\Delta(s_i)| + |\Delta(s_{i+2})|}{L}$, se tiene $2\delta \geq \dfrac{|\Delta(s_i)| + |\Delta(s_{i+2})|}{L}$, esto es $2L\delta \geq |\Delta(s_i)| + |\Delta(s_{i+2})| \geq 2\,\text{mín}(|\Delta(s_i)|, |\Delta(s_{i+2})|)$. Luego $\text{mín}(|\Delta(s_i)|, |\Delta(s_{i+2})|) \leq L\delta$, lo que implica la conclusión.

□

Notemos que estos lemas son válidos para tres valores cualquiera del parámetro, independientemente de que formen parte de una secuencia. Pero en particular, si $s_{i+1}^{(k)}$ es el valor insertado en una iteración del PIV en la secuencia $S^{(k)} = (s_0^{(k)}, s_1^{(k)}, \ldots, s_{n+k}^{(k)})$, entonces $s_{i+1}^{(k)} = \dfrac{s_i^{(k-1)} + s_{i+1}^{(k-1)}}{2} = \dfrac{s_i^{(k)} + s_{i+2}^{(k)}}{2}$. Luego tenemos que:

Corolario. *Si $S^{(k)} = (s_0^{(k)}, s_1^{(k)}, \ldots, s_{n+k}^{(k)})$ es el estado de la secuencia al final de cualquier iteración del PIV, y si $\Delta(s_{i+1}^{(k)})$ es el último punto de inserción, con $s_{i+1}^{(k)} - s_i^{(k)} \leq \delta$ para cierto δ positivo, entonces $\varepsilon \leq \dfrac{L\delta}{\text{sen}\left(\frac{\pi}{8}\right)}$.*

Demostración. Notemos que, en la k-ésima iteración del bucle "mientras", las condiciones p y q se evalúan en la secuencia $S^{(k-1)}$ que resulta de la iteración previa. Luego se verifica "$p(s_i^{(k-1)})$ o $q(s_i^{(k-1)})$", y el punto insertado tiene de parámetro a $s_{i+1}^{(k)} = \dfrac{s_i^{(k-1)} + s_{i+1}^{(k-1)}}{2} = \dfrac{s_i^{(k)} + s_{i+2}^{(k)}}{2}$. Por un lado, si $p(s_i^{(k-1)})$, entonces $\Delta(s_i^{(k-1)}) = \Delta(s_i^{(k)})$ y $\Delta(s_{i+1}^{(k-1)}) = \Delta(s_{i+2}^{(k)})$ no están conectados, y los valores $s_i^{(k)}$, $s_{i+1}^{(k)}$ y $s_{i+2}^{(k)}$ están en la hipótesis del lema 6. Por tanto $\varepsilon \leq \dfrac{L\delta}{\text{sen}\left(\frac{\pi}{8}\right)}$. Por otro lado, si $q(s_i^{(k-1)})$, entonces $(s_{i+1}^{(k-1)} - s_i^{(k-1)}) \geq \dfrac{|\Delta(s_i^{(k-1)})| + |\Delta(s_{i+1}^{(k-1)})|}{L}$, i.e. $(s_{i+2}^{(k)} - s_i^{(k)}) \geq \dfrac{|\Delta(s_i^{(k)})| + |\Delta(s_{i+2}^{(k)})|}{L}$, y podemos aplicar el lema 7 para concluir que

$\varepsilon \leq L\delta$, que es menor que $\dfrac{L\delta}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$. En cualquier caso, si $p(s_i^{(k-1)})$ o $q(s_i^{(k-1)})$, y $s_{i+1}^{(k)} - s_i^{(k)} \leq \delta$, entonces se tiene que $\varepsilon \leq \dfrac{L\delta}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$.

□

Con esta cota del valor de singularidad ε (que es $\varepsilon \leq \dfrac{L\delta}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$ si $s_{i+1}^{(k)} - s_i^{(k)} \leq \delta$ en la inserción de $s_{i+1}^{(k)}$), modificaremos el PIV de tal manera que podremos acotar el número de iteraciones que realiza. El Procedimiento de Inserción con Control de la Singularidad (PICS) (mostrado en la figura 2.17) tiene como entradas una curva analíticamente definida Δ, una sucesión $S^{(0)} = (s_0^{(0)}, s_1^{(0)}, \ldots, s_n^{(0)})$ y un parámetro real Q. Recordemos que $S^{(k)} = (s_0^{(k)}, s_1^{(k)}, \ldots, s_{n+k}^{(k)})$ es el valor de la secuencia de parámetros después de la inserción k-ésima. Las aserciones $p(s_i^{(k)})$ y $q(s_i^{(k)})$ significan lo mismo que en la sección anterior, y $r(s_i^{(k)}, Q)$ es "los valores $s_i^{(k)}$ y $s_{i+1}^{(k)}$ en la secuencia $S^{(k)}$ verifican $s_{i+1}^{(k)} - s_i^{(k)} \leq Q$".

Procedimiento de inserción con control de singularidad:
Para hallar el índice de una curva $\Delta : [a, b] \to \mathbb{C}$

Parametros de entrada: La curva Δ de constante de Lipschitz L, una secuencia $S^{(0)} = (s_0, s_1, \ldots, s_n)$, muestreo del $[a, b]$, y un parámetro real $Q > 0$.

Salida: Una secuencia que es válida para calcular $\operatorname{Ind}(\Delta)$, si se sale normalmente, o un valor t en $[a, b]$ tal que $|\Delta(t)| < LQ$, si se sale con error.

Método:

Asignar a k el valor 0.
Mientras haya un $s_i^{(k)}$ en $S^{(k)}$ con $p(s_i^{(k)})$ o con $q(s_i^{(k)})$ hacer:
{ Insertar $s_i^{(k+1)} = \dfrac{s_i^{(k)} + s_{i+1}^{(k)}}{2}$ entre $s_i^{(k)}$ y $s_{i+1}^{(k)}$;
Si $r(s_i^{(k)}, Q)$, retornar $t = s_i^{(k)}$ o $t = s_{i+1}^{(k)}$ según sea mín$(|\Delta(s_i^{(k)})|, |\Delta(s_{i+1}^{(k)})|)$; [Salida con error]
Incrementar k;
}
Retornar la secuencia resultante; [Salida normal]

Figura 2.17: Procedimiento de Inserción con Control de Singularidad (PICS). Notemos que la línea "Insertar" produce $S^{(k)}$ a partir de $S^{(k-1)}$ en la k-ésima iteración.

Informalmente, podemos decir que PICS consiste en un bucle "mientras" que se repite hasta que se obtiene una secuencia conexa (i.e., que verifica la condición "no p") y sin giros perdidos ("no q"), comprobando (con r) que este bucle no se ejecuta indefinidamente.

Notemos que la aserción $r(s_i^{(k)}, \delta)$ es equivalente a $s_{i+1}^{(k)} - s_i^{(k)} \leq \delta$, la hipótesis del corolario anterior, y en caso de retorno con error podemos aplicarlo para deducir una cota superior del valor de singularidad.

El uso de la secuencia de salida para calcular $\mathrm{Ind}(\Delta)$, y el coste de PICS, se describen en el siguiente:

Teorema 5. *Si $\Delta : [a,b] \to \mathbb{C}$ es Lipschitziana con constante L, $S^{(0)}$ una secuencia muestreo de $[a,b]$, y Q es un real positivo, el procedimiento de inserción con control de singularidad aplicado a Δ, $S^{(0)}$ y Q verifica:*

a) Retorna en menos de $\left\lfloor \dfrac{b-a}{Q} - 1 \right\rfloor_0$ *iteraciones.*

b) Si sale normalmente, la secuencia retornada da $\mathrm{Ind}(\Delta)$.

c) Si sale con error, el valor del parámetro t verifica $|\Delta(t)| \leq \dfrac{LQ}{\mathrm{sen}\left(\frac{\pi}{8}\right)}$.

Demostración. Acotaremos el número de valores del parámetro insertados entre a y b si se empieza con la secuencia inicial mínima ($a = s_0^{(0)}, s_1^{(0)} = b$). Al final de cada iteración, los valores ($a = s_0^{(k)}, s_1^{(k)}, \ldots, s_{k+1}^{(k)} = b$) deben verificar $s_{i+1}^{(k)} - s_i^{(k)} > Q$, porque en caso contrario la condición de error causa la salida. Para contar el máximo número de valores $s_i^{(k)}$ que se pueden dar con diferencia mayor que Q, consideremos que en un intervalo de longitud $b-a$ se pueden insertar m valores con una diferencia mayor que Q si $(m+1)Q < b-a$, sin contar valores insertados en los extremos. Luego el número k de puntos interiores $s_1^{(k)}, \ldots, s_k^{(k)}$ debe verificar $(k+1)Q < b-a$, y el máximo valor posible para k es $\left\lfloor \dfrac{b-a}{Q} - 1 \right\rfloor_0$. Luego como mucho $\left\lfloor \dfrac{b-a}{Q} - 1 \right\rfloor_0$ iteraciones se realizan hasta que se alcanza la condición de error $r(s_i^{(k)}, Q)$, si no hay antes una salida normal.

Para *b*), el caso de salida normal, la traza de ejecución del PICS coincide con la del PIV, y por tanto la secuencia retornada es válida para calcular el $\mathrm{Ind}(\Delta)$ por el teorema 4.

Respecto al valor del parámetro t retornado si hay error, notemos que en tal caso se verifica que "$p(s_i^{(k-1)})$ o $q(s_i^{(k-1)})$, y $r(s_i^{(k)}, Q)$", que son las condiciones requeridas para entrar en el bucle y salir con error. En el caso de que se verifique "$p(s_i^{(k-1)})$ y $r(s_i^{(k)}, Q)$", la última iteración ha sido de tipo p (esto es, $s_{i+1}^{(k)} = \dfrac{s_i^{(k)} + s_{i+2}^{(k)}}{2}$ con $\Delta(s_i^{(k)})$ y $\Delta(s_{i+2}^{(k)})$ en sectores no conectados) y $s_{i+1}^{(k)} - s_i^{(k)} \leq Q$. Por el lema 6, tenemos que o bien $|\Delta(s_i^{(k)})|$ o bien $|\Delta(s_{i+2}^{(k)})|$ es menor o igual que $\dfrac{LQ}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$. En el caso de que se verifique "$q(s_i^{(k-1)})$ y $r(s_i^{(k)}, Q)$", la última iteración ha sido de tipo q (esto es, $s_{i+1}^{(k)} = \dfrac{s_i^{(k)} + s_{i+2}^{(k)}}{2}$ con $(s_{i+2}^{(k)} - s_i^{(k)}) \geq \dfrac{|\Delta(s_i^{(k)})| + |\Delta(s_{i+2}^{(k)})|}{L}$) y $s_{i+1}^{(k)} - s_i^{(k)} \leq Q$. Luego por el lema 7 o bien $|\Delta(s_i^{(k)})|$ o bien $|\Delta(s_{i+2}^{(k)})|$ es menor que LQ. Como t es precisamente $t = s_i^{(k)}$ or $t = s_{i+1}^{(k)}$ dependiendo de mín$(|\Delta(s_i^{(k)})|, |\Delta(s_{i+1}^{(k)})|)$, en cualquiera de los dos casos el valor de retorno t verifica $|\Delta(t)| \leq \dfrac{LQ}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$.

□

Resumiendo, el procedimiento de inserción con control de singularidad evita un número excesivo de iteraciones, controlado por un parámetro de entrada Q. Así, PICS calcula de modo efectivo el índice de curvas Δ con $\varepsilon > \dfrac{LQ}{\operatorname{sen}\left(\frac{\pi}{8}\right)}$, pero si el valor de singularidad está por debajo de este nivel, el procedimiento retorna con error, señalando este hecho, o puede retornar normalmente con una secuencia cuyo número de cruces coincide con el índice de la curva, siempre en menos de $\left\lfloor \dfrac{b-a}{Q} - 1 \right\rfloor_0$ iteraciones.

Para encuadrar este resultado en el marco de la literatura existente, hay que considerar que hemos desarrollado un algoritmo para el cálculo del índice, usando la discretización de la curva para obtener una secuencia a la que se puede aplicar con fiabilidad el método de Henrici de contar el número de pasos por el eje de las abscisas positivo. También damos una cota de su coste computacional en casos no singulares, en el teorema 4. Esta cota es del orden de $O\left(\dfrac{1}{\varepsilon^2}\log\left(\dfrac{1}{\varepsilon}\right)\right)$ evaluaciones de la curva, siendo ε su distancia mínima al origen. También damos otro algoritmo para calcular el índice que se puede aplicar a curvas cuya distancia mínima es desconocida, pero asegurando que, antes de invertir una cantidad predefinida de cálculo, retornará con el índice calculado o con un certificado de que la curva es casi singular.

Como conexión con otros trabajos, y sugerencias de desarrollo futuro, podemos decir que el coste de $O\left(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\varepsilon}\right)\right)$ es coherente con los procedimientos analizados en [Chou and Ko, 1995], que dependen de un parámetro n para calcular el índice de curvas con valor de singularidad mayor que 2^{-n}.

El valor de singularidad ε se relaciona con el condicionamiento en cálculo numérico descrito en la introducción. Dentro de una clase dada de problemas, el número de condición de un problema específico mide la dificultad de su resolución comparada con los restantes de su clase. El recíproco del valor de singularidad, $\frac{1}{\varepsilon}$, puede verse como un "número de condición" similar al usado en análisis numérico lineal. Esta analogía de $\frac{1}{\varepsilon}$ con un número de condición es subrayada por el hecho de que coincide con la distancia de Δ a la curva más próxima mal planteada (*ill-posed*, [Demmel, 1987, Blum et al., 1998]) para el problema del índice, para cierta métrica definida en el conjunto de curvas planas. Intuitivamente, mide la proximidad del problema específico a un problema mal planteado de la clase. En nuestro caso, un problema mal planteado es una curva que cruza el origen, sin un índice definido.

Capítulo 3

Método geométrico para calcular el número de raíces

Siguiendo con el programa expuesto en la introducción para desarrollar un método geométrico, ahora que ya disponemos del método PICS para el cálculo del índice de curvas planas del capítulo anterior, en este se particulariza, en la sección 3.1, para curvas imagen por un polinomio, y se incorpora a un procedimiento recursivo de descomposición en subregiones, en la sección 3.2. Como se ha comentado, la principal aportación de nuestro método es que permite evitar las curvas singulares, lo es crucial para la descomposición pues asegura la finitud del proceso. Que la gestión de los errores producidos por curvas singulares es correcta se demuestra en la sección 3.3. También se hace un análisis de coste en la sección 3.4.

Resumimos el capítulo anterior: calculamos el índice de una curva utilizando un muestreo discreto obtenido por un procedimiento iterativo. Es el procedimiento básico de Ying-Katz (figura 2.6), con un bucle de recorrido-inserción. Como aportación, hemos demostrado la convergencia de este procedimiento para curvas no singulares. También hemos introducido el Procedimiento de Inserción Válido para cualquier secuencia inicial (PIV, figura 2.13). Es una modificación de IP con una condición adicional de entrada en el bucle, que asegura el cálculo correcto del índice de la curva mediante el número total de cruces $C_7 - C_0$ de la secuencia resultante. Informalmente, se puede decir que PIV involucra un bucle *mientras* que es repetido hasta que se obtiene una secuencia conectada (es decir, que verifica la

condición "no p") y sin giros perdidos ("no q"). En el teorema 4 se ha demostrado la validez de la secuencia de salida para calcular $\mathrm{Ind}(\Delta)$, analizando también el coste del PIV usando la noción de ε-singularidad.

Sin tener en cuenta las constantes, esta cota del coste es de orden $O\left(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\varepsilon}\right)\right)$. Sin embargo, este resultado es de escaso interés para la aplicación práctica de PIV, debido a que el valor ε no se conoce previamente. El teorema anterior nos dice que el número de iteraciones es menor que una cota que crece proporcionalmente a $\frac{1}{\varepsilon}$ (esto es, disminuyendo con ε). Recíprocamente, podemos deducir que si el número de iteraciones supera cierto umbral, el valor ε debe ser bajo, es decir, la curva debe estar cerca del origen. Siguiendo este razonamiento, hemos definido otro procedimiento que incluye una prueba para detectar un valor pequeño de ε y salir del procedimiento en ese caso (que corresponde con una curva próxima a ser singular). Se ha denominado Procedimiento de Inserción con Control de Singularidad (PICS), que evita un número excesivo de iteraciones, controlado por el parámetro de entrada Q. De este modo se mantienen acotados los requisitos de cómputo cuando se aplica a curvas singulares o próximas a singulares [García Zapata and Díaz Martín, 2012].

El teorema 5 establece que PICS calcula en efecto el índice de las curvas Δ con $\varepsilon > \frac{LQ}{\mathrm{sen}\left(\frac{\pi}{8}\right)}$. Para curvas cuya distancia al origen esté por debajo de este nivel, PICS puede o bien retornar normalmente con una secuencia válida para calcular el índice, o bien retornar con error, indicando que la curva de entrada es ε-singular. En cualquier caso, el procedimiento retorna en menos de $\left\lfloor\frac{b-a}{Q}\right\rfloor$ iteraciones.

3.1. Procedimiento de inserción para curvas de la forma $\Delta = f(\Gamma)$

El coste computacional (es decir, el número de iteraciones) del cálculo del índice de una curva usando PIV está relacionado con la distancia de la curva al origen $\mathrm{d}(O, \Delta)$ (teorema 4). El procedimiento PICS, por el contrario, cuando el coste sobrepasa un umbral, señala error y retorna una cota de esta distancia (teorema 5). En este capítulo nos centramos en curvas de la forma $\Delta = f(\Gamma)$ para un polinomio f. Estas curvas surgen cuando calculamos el número de raíces de f

dentro de Γ. Para este particular tipo de curva, el coste de aplicar PIV a Δ dado por el teorema 4 puede expresarse en función de la distancia de Γ hasta la raíz más cercana de f, en vez de la distancia de Δ al origen. Además, si aplicamos PICS a Γ, en caso de error, tenemos por el teorema 5 una cota superior de la distancia de Γ al origen. Sin embargo, mostraremos que esta cota es muy laxa, de modo que no es útil para usar PICS como test de inclusión para hallar raíces. Necesitamos una cota más ajustada porque en la partición en subregiones de la región de búsqueda, propia de cualquier método divide-y-vencerás, si el borde Γ de una partición pasa por una raíz, su transformación Δ es singular. Con una cota más ajustada podemos modificar la partición de modo que su borde no pase por, o cerca de, una raíz, y así evitar curvas singulares. En consecuencia, definimos otro procedimiento (de inserción con control de proximidad de raíz, PCR, descrito en la figura 3.1 y teorema 6) que, cuando retorna con error, nos da una cota de cierta función (el número de condición $\kappa_f(\Gamma)$) que depende de las raíces y la curva. En la siguiente sección se verá como usar esta cota en el método iterativo de partición de la señal de búsqueda.

Para calcular el índice de las curvas Δ usando PIV, el teorema 4 requiere que Δ sea Lipschitziana. Para cumplir este requisito técnico en curvas de la forma $\Delta = f(\Gamma)$ para un polinomio f, imponemos que Γ esté uniformemente parametrizada. Esto no supone una restricción en la práctica, ya que la curva Γ que encierra la región que nos interesa normalmente se construye conectando segmentos rectos (o arcos de circunferencia) uniformemente parametrizados. Para mostrar que si Γ es uniformemente parametrizada entonces $f(\Gamma)$ es Lipschitziana, usamos que Γ es acotada (porque es la imagen de un conjunto compacto $\Gamma : [a, b] \to \mathbb{C}$). Consecuentemente, como una función diferenciable en un conjunto acotado es Lipschitziana [Kolmogorov and Fomın, 1975], el polinomio f es Lipschitziano (en Γ) con constante $L = \sup_{x \in \Gamma} |f'(\Gamma(x))|$. Por tanto, $\Delta = f(\Gamma)$ es la composición de una curva uniformemente parametrizada Γ y una función Lipschitziana f, y por tanto Δ es una curva Lipschitziana con la misma constante L. Se verifica así la hipótesis del teorema 4, y PIV calcula el índice de $\Delta = f(\Gamma)$ (es decir, el número de raíces dentro de Γ).

El factor clave en el coste del cálculo del índice, por el teorema 4, es la distancia al origen desde Δ. Esta distancia y las raíces de f están relacionadas como se describe en la siguiente proposición.

Sea $f(z) = a_n z^n + \cdots + a_1 z + a_0$ un polinomio de grado n y su descomposición en raíces $f(z) = a_n(z-z_1)(z-z_2)\dots(z-z_n)$. Recordemos que la distancia $\mathrm{d}(A,B)$ entre dos conjuntos A y B es el mínimo de las distancias entre cada par de puntos, cada uno perteneciente a su respectivo conjunto. En particular, si Z es el conjunto de las raíces de f, $Z = \{z_1, z_2, \dots, z_n\}$, su distancia $\mathrm{d}(Z,\Gamma)$ a Γ es la distancia desde la raíz más cercana a Γ, y $\mathrm{d}(O,\Delta)$ es la distancia desde el origen O a Δ.

Proposición. *Si $\Gamma : [a,b] \to \mathbb{C}$ es una curva uniformemente parametrizada, f un polinomio de grado n, con constante de Lipschitz L, y $\Delta = f(\Gamma)$, entonces se tiene:*

$$|a_n|\,\mathrm{d}(Z,\Gamma)^n \leq \mathrm{d}(O,\Delta) \leq L\,\mathrm{d}(Z,\Gamma)$$

Demostración. Para la primera desigualdad, usando la descomposición en raíces de f, tenemos que para cada $t \in [a,b]$:

$$f(\Gamma(t)) = a_n(\Gamma(t)-z_1)(\Gamma(t)-z_2)\cdots(\Gamma(t)-z_n)$$

Tomando módulo,

$$|\Delta(t)| = |f(\Gamma(t))| = |a_n||\Gamma(t)-z_1||\Gamma(t)-z_2|\cdots|\Gamma(t)-z_n|$$

Recordemos que $\operatorname{ínf}(gh) \geq \operatorname{ínf}(g)\operatorname{ínf}(h)$, si g y h son funciones no negativas. Si tomamos el ínfimo sobre $t \in [a,b]$ en la anterior igualdad, tenemos

$$\begin{aligned}\inf_{t\in[a,b]} |\Delta(t)| &= |a_n| \operatorname{ínf}(|\Gamma(t)-z_1||\Gamma(t)-z_2|\dots|\Gamma(t)-z_n|) \geq \\ &\geq |a_n| \operatorname{ínf}|\Gamma(t)-z_1| \operatorname{ínf}|\Gamma(t)-z_2| \dots \operatorname{ínf}|\Gamma(t)-z_n| \geq \\ &\geq |a_n| \left(\min_{i=1\dots n} \inf_{t\in[a,b]} |\Gamma(t)-z_i|\right)^n.\end{aligned}$$

La última desigualdad se debe a que el producto de n factores es mayor que la n-ésima potencia del factor mínimo. Por tanto $\inf_{t\in[a,b]} |\Delta(t)| \geq |a_n|\,(\min \operatorname{ínf}|\Gamma(t)-z_i|)^n$. Como $\mathrm{d}(O,\Delta) = \inf_{t\in[a,b]} |\Delta(t)|$ y $\mathrm{d}(Z,\Gamma) = \min_{i=1\dots n} \inf_{t\in[a,b]} |\Gamma(t)-z_i|$ por definición, se tiene que $\mathrm{d}(O,\Delta) \geq |a_n|\,\mathrm{d}(Z,\Gamma)^n$.

Para la segunda desigualdad, para cada $t \in [a,b]$ y raíz z_i, $|\Delta(t)| = |f(\Gamma(t))| = |f(\Gamma(t)) - f(z_i)| \leq L|\Gamma(t)-z_i|$, por lipschitzianidad. Tomando el ínfimo sobre

$t \in [a,b]$ como antes, se tiene que, para cada z_i, $\inf_{t\in[a,b]} |\Delta(t)| \leq L \inf_{t\in[a,b]} |\Gamma(t) - z_i|$. Por tanto $\inf_{t\in[a,b]} |\Delta(t)| \leq L \min_{i=1\ldots n} \inf_{t\in[a,b]} |\Gamma(t) - z_i|$ y llegamos a la conclusión.

□

Recordemos que si aplicamos el procedimiento PIV para calcular el número de raíces dentro de Γ, su coste será $O\left(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\varepsilon}\right)\right)$ por la cota del teorema 4, donde $\varepsilon = \mathrm{d}(O,\Delta)$. Para expresarlo en términos que dependan de Γ, por la desigualdad $\frac{1}{|a_n|\,\mathrm{d}(Z,\Gamma)^n} \geq \frac{1}{\mathrm{d}(O,\Delta)}$ deducida de la anterior proposición, el cálculo tiene un coste de orden menor o igual a

$$O\left(\frac{1}{(|a_n|\,\mathrm{d}(Z,\Gamma)^n)^2}\log\left(\frac{1}{|a_n|\,\mathrm{d}(Z,\Gamma)^n}\right)\right)$$

es decir $O\left(\frac{1}{\mathrm{d}(Z,\Gamma)^{2n}}\log\left(\frac{1}{\mathrm{d}(Z,\Gamma)}\right)\right)$. Debe notarse que esta dependencia del coste respecto a la distancia a la raíz más cercana coincide con la que tienen otros algoritmos que calculan el número de raíces en una región [Renegar, 1987, Pan, 1997].

Como la distancia $\mathrm{d}(Z,\Gamma)$ a las raíces es desconocida, nos enfrentamos a un coste indeterminado usando PIV, como se discutió en el capítulo anterior. Esta fue la motivación para introducir PICS (figura 2.17), que nos asegura un coste acotado. Además, una segunda ventaja de PICS es que si se alcanza la cota del coste, retorna con error devolviendo un punto de la curva cerca del origen, $|\Delta(t)| \leq \frac{LQ}{\mathrm{sen}\left(\frac{\pi}{8}\right)}$, como establece el teorema 5. Esto nos da una cota a $\mathrm{d}(O,\Delta)$, que es $\frac{LQ}{\mathrm{sen}\left(\frac{\pi}{8}\right)}$. En el caso particular de curvas de la forma $\Delta = f(\Gamma)$ podemos deducir una cota superior de $\mathrm{d}(Z,\Gamma)$: por la proposición 3.1, $|a_n|\,\mathrm{d}(Z,\Gamma)^n \leq \mathrm{d}(O,\Delta) \leq |\Delta(t)| \leq \frac{LQ}{\mathrm{sen}\left(\frac{\pi}{8}\right)}$, y por tanto $\mathrm{d}(Z,\Gamma) \leq \sqrt[n]{\frac{LQ}{|a_n|\,\mathrm{sen}(\pi/8)}}$, en caso de una salida con error de PICS.

Esta cota dada para PICS es muy laxa, ya que decrece mucho más lentamente que el parámetro Q: su valor puede ser cercano a 1 para valores moderados del grado n del polinomio, incluso con un Q muy pequeño. Una fórmula de acotación en la que intervenga la raíz n-ésima es de dudosa utilidad. En caso de retornar con error necesitamos un cota de $\mathrm{d}(Z,\Gamma)$ más ajustada, para así poder localizar una

raíz bien cercana al punto que produce el error. El procedimiento PICS, al menos como esta descrito en el teorema 5, no proporciona una buena cota.

Procedimiento de inserción con control de proximidad de raíz:
Para encontrar el número de raíces de un polinomio f dentro de una curva $\Gamma : [a, b] \to \mathbb{C}$

Parámetros de entrada: La curva Γ uniformemente parametrizada, el polinomio f de grado n, una secuencia $S = (s_0, \ldots, s_n)$, muestreo de $[a, b]$, y un parámetro real $Q > 0$.

Salida: Una secuencia $(t_0, \ldots, t_m)$ que es válida para calcular $\mathrm{Ind}(f(\Gamma))$ si se sale normalmente. Un valor $t \in [a, b]$ tal que hay una raíz z de f con $\mathrm{d}(z, \Gamma(t)) \leq \dfrac{4nQ}{2-\sqrt{2}}$, si se sale con error.

Método:

Mientras haya un s_i en S con $p(s_i)$ o con $q_2(s_i)$ hacer:
{ Insertar $\dfrac{s_i + s_{i+1}}{2}$ entre s_i y s_{i+1};
Si $r(s_i, Q)$ entonces retornar $t = s_{i+1}$; [Salida con error].
}
Retornar la secuencia resultante; [Salida normal]

Figura 3.1: Procedimiento de inserción con control de proximidad de raíz.

Debemos pues definir otro procedimiento para el cual exista una cota que dependa linealmente de Q. En el Procedimiento con Control de proximidad de las Raíces (PCR, figura 3.1) utilizamos los predicados p, q_2 y r significando $p(s_i)$ que "los valores s_i y s_{i+1} en la secuencia S tienen sus imágenes $f(\Gamma(s_i))$ y $f(\Gamma(s_{i+1}))$ no conectadas", $q_2(s_i)$ es "los valores s_i y s_{i+1} en la secuencia S verifican:

$$|f(\Gamma(s_i))| + |f(\Gamma(s_{i+1}))| \leq 2\,|f'(\Gamma(s_i))|\,(s_{i+1} - s_i) + |f(\Gamma(s_{i+1})) - f(\Gamma(s_i))|\text{"},$$

y $r(s_i, Q)$ significa "los valores s_i y s_{i+1} en la secuencia S verifican $(s_{i+1} - s_i) \leq Q$". Como antes, s_{i+1} en el caso extremo de $i = n$ debe entenderse como s_0. Las aserciones p y r son iguales que en PICS, pero q_2 es distinto de q. La aserción q_2 será necesaria más adelante en la proposición 1 para demostrar que no hay giros perdidos y en la proposición 3 para demostrar que se verifica cierta cota en caso de retorno con error.

El retorno de PCR verifica una afirmación que involucra $\kappa_f(\Gamma)$, el *número de*

condición de la curva Γ *con respecto a* f, definido como la suma de los inversos de las distancias a Γ de cada raíz de f:

$$\kappa_f(\Gamma) = \sum_{i=1}^{n} \frac{1}{\mathrm{d}(z_i, \Gamma)}.$$

La cota en caso de retorno con error hace que el número de condición $\kappa_f(\Gamma)$ en PCR juegue un papel similar al que tiene el inverso de $\mathrm{d}(O, \Delta)$, la singularidad de Δ, en PICS. Por el teorema 5, un retorno con error de PICS implica un valor bajo de $\mathrm{d}(O, \Delta)$, es decir, un valor alto de $\dfrac{1}{\mathrm{d}(O, \Delta)}$. Mostraremos que un retorno con error de PCR implica un valor alto de $\kappa_f(\Gamma) = \sum_{i=1}^{n} \dfrac{1}{\mathrm{d}(z_i, \Gamma)}$.

Con ayuda de la siguiente proposición demostraremos (en el teorema 6) que si PCR retorna normalmente, la secuencia retornada nos da el número de raíces dentro de Γ.

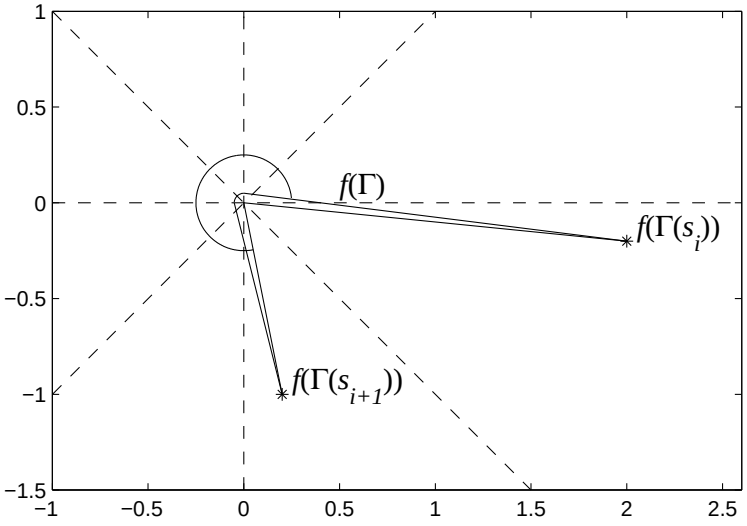


Figura 3.2: La longitud de arco de la curva $f(\Gamma([s_i, s_{i+1}]))$ es mayor que $|f(\Gamma(s_i))| + |f(\Gamma(s_{i+1}))|$.

Proposición 1. *En una curva* $f(\Gamma)$*, si hay un giro perdido entre* s_i *y* s_{i+1}*, entonces se verifica* $q_2(s_i)$ *en la secuencia* $S = (\ldots, s_i, s_{i+1}, \ldots)$.

Demostración. Geométricamente, si la curva $f(\Gamma([s_i, s_{i+1}]))$ traza un giro perdido,

su longitud es mayor que la suma de los módulos $|f(\Gamma(s_i))|$ y $|f(\Gamma(s_{i+1}))|$ (figura 3.2).

La longitud de arco de $f(\Gamma)$ en el intervalo paramétrico $[s_i, s_{i+1}]$ es $\int_{s_i}^{s_{i+1}} |f'(\Gamma(x))|\, dx$ porque Γ está uniformemente parametrizada [Kolmogorov and Fomin, 1975]. El polinomio de Taylor de $f'(\Gamma(x))$ en torno a s_i es

$$f'(\Gamma(x)) = f'(\Gamma(s_i)) + f''(\Gamma(s_i))(x - s_i) + \sum_{k=3}^{n} \frac{f^{(k)}(\Gamma(s_i))}{(k-1)!}(x - s_i)^{k-1}$$

por tanto

$$\begin{aligned}
\int_{s_i}^{s_{i+1}} |f'(\Gamma(x))|\, dx &= \int_{s_i}^{s_{i+1}} \left| f'(\Gamma(s_i)) + \sum_{k=2}^{n} \frac{f^{(k)}(\Gamma(s_i))}{(k-1)!}(x - s_i)^{k-1} \right| dx \leq \\
&\leq \int_{s_i}^{s_{i+1}} |f'(\Gamma(s_i))|\, dx + \sum_{k=2}^{n} \int_{s_i}^{s_{i+1}} \left| \frac{f^{(k)}(\Gamma(s_i))}{(k-1)!}(x - s_i)^{k-1} \right| dx = \\
&= |f'(\Gamma(s_i))|\,(s_{i+1} - s_i) + \sum_{k=2}^{n} \frac{\left|f^{(k)}(\Gamma(s_i))\right|}{(k-1)!} \int_{s_i}^{s_{i+1}} (x - s_i)^{k-1}\, dx = \\
&= |f'(\Gamma(s_i))|\,(s_{i+1} - s_i) + \sum_{k=2}^{n} \frac{\left|f^{(k)}(\Gamma(s_i))\right|}{(k-1)!} \frac{(s_{i+1} - s_i)^k}{k}
\end{aligned}$$

Además, el polinomio de Taylor de $f(\Gamma(x))$ alrededor s_i es

$$f(\Gamma(x)) = f(\Gamma(s_i)) + f'(\Gamma(s_i))(x - s_i) + \sum_{k=2}^{n} \frac{f^{(k)}(\Gamma(s_i))}{k!}(x - s_i)^k$$

Para $x = s_{i+1}$, tomando valores absolutos y despejando el sumatorio tenemos:

$$\begin{aligned}
\sum_{k=2}^{n} \frac{\left|f^{(k)}(\Gamma(s_i))\right|}{k!}(s_{i+1} - s_i)^k &= |f(\Gamma(s_{i+1})) - f(\Gamma(s_i)) - f'(\Gamma(s_i))(s_{i+1} - s_i)| \leq \\
&\leq |f(\Gamma(s_{i+1})) - f(\Gamma(s_i))| + |f'(\Gamma(s_i))|\,(s_{i+1} - s_i)
\end{aligned}$$

Por tanto

$$\begin{aligned}&\int_{s_i}^{s_{i+1}} |f'(\Gamma(x))|dx \leq \\ &\leq |f'(\Gamma(s_i))|\,(s_{i+1}-s_i) + |f(\Gamma(s_{i+1})) - f(\Gamma(s_i))| + |f'(\Gamma(s_i))|\,(s_{i+1}-s_i) = \\ &= 2\,|f'(\Gamma(s_i))|\,(s_{i+1}-s_i) + |f(\Gamma(s_{i+1})) - f(\Gamma(s_i))|\,.\end{aligned}$$

Así, si hay un giro perdido entonces

$$\begin{aligned}|f(\Gamma(s_i))| + |f(\Gamma(s_{i+1}))| \leq \int_{s_i}^{s_{i+1}} |f'(\Gamma(x))|\; dx \leq \\ \leq 2\,|f'(\Gamma(s_i))|\,(s_{i+1}-s_i) + |f(\Gamma(s_{i+1})) - f(\Gamma(s_i))|\,,\end{aligned}$$

que es precisamente $q_2(s_i)$.

□

Ahora clasificamos las iteraciones realizadas por el bucle "mientras". Una iteración es *de tipo p* si se realiza porque se verifica el predicado $p(s_i)$, y es *de tipo q* si se realiza porque se verifica la propiedad "no $p(s_i)$, y $q_2(s_i)$". Así toda iteración es de tipo p o de tipo q, pero no de ambos a la vez. Del mismo modo se pueden clasificar las salidas con error como de tipo p, o q, según el tipo de iteración en la que ocurre el retorno.

Mostraremos en el teorema 6 que PCR retorna con una secuencia válida para calcular el número de raíces dentro de Γ, o con una cota inferior del número de condición (equivalentemente, una cota superior de la distancia a una raíz, el análogo del teorema 5 para PICS). Previamente damos cotas inferiores de este número de condición en caso de excepciones de tipo p y q en las proposiciones 2 y 3, respectivamente, después de varios lemas auxiliares. Consideraremos los ángulos medidos en el intervalo $[0, 2\pi)$.

Lema 8. *Si α es el ángulo orientado entre tres números complejos (x, z, y) con vértice en z (es decir, $\alpha = \arg(y-z) - \arg(x-z)$) entonces*

$$\text{mín}(\mathrm{d}(z,x), \mathrm{d}(z,y)) \leq \frac{\mathrm{d}(x,y)}{2|\operatorname{sen}(\alpha/2)|}$$

Demostración. El ángulo orientado puede tomar cualquier valor entre 0 y 2π (ver la figura 3.3). Vamos a discutir solo el caso de ángulo α menor o igual que π, porque

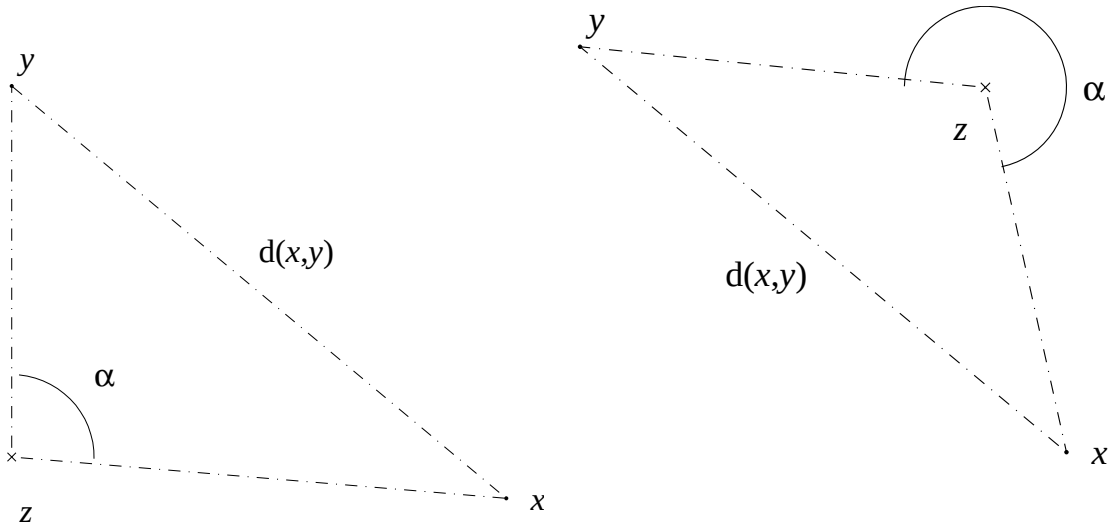


Figura 3.3: El ángulo orientado en z puede ser tanto menor como mayor que π.

si el ángulo orientado α entre (x, z, y) es $\alpha > \pi$, entonces el ángulo orientado entre (y, z, x), es decir $2\pi - \alpha$, es menor que π. Por otro lado $|\operatorname{sen}((2\pi - \alpha)/2)| = |\operatorname{sen}(\pi - \alpha/2)| = |\operatorname{sen}(-\alpha/2)| = |\operatorname{sen}(\alpha/2)|$ porque la función $|\operatorname{sen}|$ es de periodo π e impar. Por tanto la afirmación de la proposición es la misma para los ángulos (x, z, y) y (y, z, x), así que podemos discutir solo el caso de ángulo menor o igual que π.

Supongamos que $\mathrm{d}(z, x) \geq \mathrm{d}(z, y)$, y sea D el punto del segmento $\overline{zx}$ a la misma distancia de z que y (figura 3.4).

Se verifica $\mathrm{d}(D, y) \leq \mathrm{d}(x, y)$ porque el triángulo isósceles de vértices z, y y D tiene el lado opuesto a z mínimo entre aquellos triángulos que tienen un ángulo α y cuyo menor lado adyacente a este tiene longitud $\mathrm{d}(z, y)$. Además, considerando el triángulo rectángulo que surge de la bisección de α, tenemos que $\operatorname{sen}(\alpha/2) = \frac{\frac{\mathrm{d}(D,y)}{2}}{\mathrm{d}(z,y)}$, es decir

$$\mathrm{d}(z, y) = \frac{\mathrm{d}(D, y)}{2\operatorname{sen}(\alpha/2)} \leq \frac{\mathrm{d}(x, y)}{2\operatorname{sen}(\alpha/2)}$$

El valor de $\operatorname{sen}(\alpha/2)$ es el mismo que $|\operatorname{sen}(\alpha/2)|$ porque $\alpha/2 < \pi$.

En el caso contrario $\mathrm{d}(z, x) < \mathrm{d}(z, y)$, tenemos de modo similar que $\mathrm{d}(z, x) \leq \frac{\mathrm{d}(x, y)}{2\operatorname{sen}(\alpha/2)}$. Tanto en el primero como en el segundo caso tenemos la cota, por

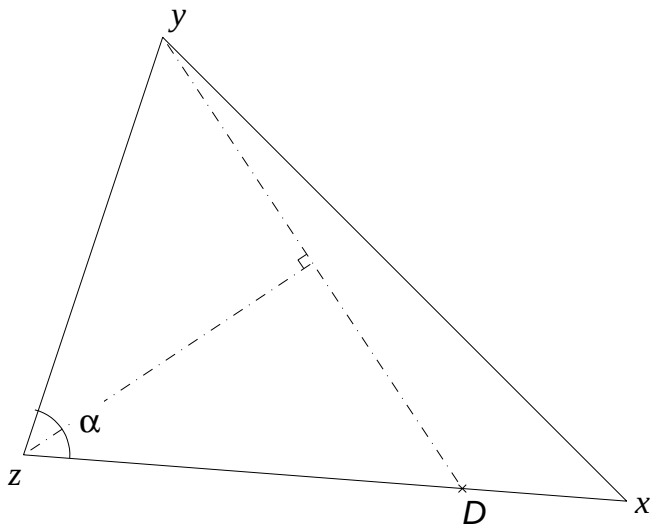


Figura 3.4: El punto D está a la misma distancia de z que y.

tanto el mínimo es menor o igual, como queríamos mostrar.

□

Proposición 2. *Supongamos que se aplica PCR una curva* Γ *y un polinomio* f *de raíces* $z_1, z_2, \ldots, z_n$. *En una p-iteración en la que el valor insertado sea* s_i, *si* $|s_{i+1} - s_i| < \delta$, *entonces* $\kappa_f(\Gamma) \geq \dfrac{\operatorname{sen}(\pi/8)}{\delta}$.

Demostración. El valor insertado s_i es el punto medio de s_{i-1} y s_{i+1}, por tanto $s_{i+1} - s_{i-1} = s_{i+1} - s_i + s_i - s_{i-1} < \delta + \delta$. Por la parametrización uniforme de Γ, $|s_{i+1} - s_{i-1}| = |\Gamma(s_{i+1}) - \Gamma(s_{i-1})|$ y por tanto $|\Gamma(s_{i+1}) - \Gamma(s_{i-1})| < 2\delta$.

Además, como el valor s_i se inserta en una p-iteración, los puntos $f(\Gamma(s_{i-1}))$ y $f(\Gamma(s_{i+1}))$ (cuyos parámetros s_{i-1} y s_{i+1} están situados consecutivamente en la secuencia S antes de la inserción) están en sectores no conexos. Geométricamente esto implica que:

$$\frac{\pi}{4} < \arg f(\Gamma(s_{i+1})) - \arg f(\Gamma(s_{i-1})) < \frac{7\pi}{4}$$

Por otro lado, por la descomposición en raíces del polinomio f

$$f(z) = a_n(z - z_1)(z - z_2)\ldots(z - z_n)$$

se verifica que $\arg f(z) = \arg(a_n) + \arg(z - z_1) + \arg(z - z_2) + \cdots + \arg(z - z_n)$, donde la suma de ángulos se entiende modulo 2π.

Dando los valores concretos $z = \Gamma(s_{i+1})$ y $z = \Gamma(s_{i-1})$ tenemos:

$$\arg f(\Gamma(s_{i+1})) = \arg(a_n) + \arg(\Gamma(s_{i+1}) - z_1) + \cdots + \arg(\Gamma(s_{i+1}) - z_n)$$

y

$$\arg f(\Gamma(s_{i-1})) = \arg(a_n) + \arg(\Gamma(s_{i-1}) - z_1) + \cdots + \arg(\Gamma(s_{i-1}) - z_n)$$

Llamando α_j al ángulo orientado entre $\Gamma(s_{i-1})$ y $\Gamma(s_{i+1})$ con vértice en z_j, es decir, $\alpha_j = \arg(\Gamma(s_{i+1}) - z_j) - \arg(\Gamma(s_{i-1}) - z_j)$, la diferencia de las expresiones anteriores da:

$$\arg f(\Gamma(s_{i+1})) - \arg f(\Gamma(s_{i-1})) = \alpha_1 + \alpha_2 + \cdots + \alpha_n$$

Por tanto tenemos que

$$\frac{\pi}{4} < \sum_{j=1}^{n} \alpha_j < \frac{7\pi}{4}.$$

Aparte de esto, llamamos $m_j = \text{mín}(|z_j - \Gamma(s_{i-1})|, |z_j - \Gamma(s_{i+1})|)$ para cada j entre 1 y n. Por la proposición 8 con $x = \Gamma(s_{i-1})$, $y = \Gamma(s_{i+1})$ y $z = z_j$, tenemos que

$$m_j = \text{mín}(\mathrm{d}(z_j, \Gamma(s_{i-1})), \mathrm{d}(z_j, \Gamma(s_{i+1}))) \leq \frac{|\Gamma(s_{i-1}) - \Gamma(s_{i+1})|}{2|\operatorname{sen}(\alpha_j/2)|}$$

Recordemos que $|\Gamma(s_{i+1}) - \Gamma(s_{i-1})| < 2\delta$, luego $m_j < \dfrac{\delta}{|\operatorname{sen}(\alpha_j/2)|}$. Tomando inversos:

$$\frac{1}{m_j} > \frac{1}{\delta}\left|\operatorname{sen}\left(\frac{\alpha_j}{2}\right)\right|$$

Por definición de distancia, $\mathrm{d}(z_j, \Gamma) \leq m_j$, así que

$$\kappa_f(\Gamma) = \sum_{j=1}^{n} \frac{1}{\mathrm{d}(z_j, \Gamma)} \geq \sum_{j=1}^{n} \frac{1}{m_j} > \frac{1}{\delta} \sum_{j=1}^{n} \left|\operatorname{sen}\left(\frac{\alpha_j}{2}\right)\right|$$

Ahora notemos que, por la anterior cadena de desigualdades $\dfrac{\pi}{4} < \displaystyle\sum_{j=1}^{n} \alpha_j < \frac{7\pi}{4}$, tenemos $\dfrac{\pi}{8} < \displaystyle\sum_{j=1}^{n} \frac{\alpha_j}{2} < \frac{7\pi}{8}$, y entonces $\displaystyle\sum_{j=1}^{n} \frac{\alpha_j}{2}$ está en el intervalo $[0, \pi]$ donde la

función seno es subaditiva [Schechter, 1996] (esto significa que $\text{sen}(\alpha) + \text{sen}(\beta) \geq \text{sen}(\alpha + \beta)$). por tanto:

$$\frac{1}{\delta}\sum_{j=1}^{n}\left|\text{sen}\left(\frac{\alpha_j}{2}\right)\right| \geq \frac{1}{\delta}\left|\sum_{j=1}^{n}\text{sen}\left(\frac{\alpha_j}{2}\right)\right| \geq \frac{1}{\delta}\left|\text{sen}\left(\sum_{j=1}^{n}\frac{\alpha_j}{2}\right)\right|$$

Como $\sum_{j=1}^{n}\frac{\alpha_j}{2}$ está entre $\frac{\pi}{8}$ y $\frac{7\pi}{8}$, y en este intervalo la función seno alcanza el mínimo en un punto extremo:

$$\kappa_f(\Gamma) > \frac{1}{\delta}\left|\text{sen}\left(\sum_{j=1}^{n}\frac{\alpha_j}{2}\right)\right| \geq \frac{1}{\delta}\,\text{mín}\left(\left|\text{sen}\left(\frac{\pi}{8}\right)\right|, \left|\text{sen}\left(\frac{7\pi}{8}\right)\right|\right)$$

Estos dos valores coinciden, luego $\kappa_f(\Gamma) > \frac{1}{\delta}\,\text{sen}(\pi/8)$, como queríamos demostrar. □

Lema 9. *Para cada punto $x \in \Gamma$ se tiene:*

$$\left|\frac{f'(x)}{f(x)}\right| \leq \kappa_f(\Gamma)$$

Demostración. Un cálculo [Henrici, 1988] nos da

$$\frac{f'(x)}{f(x)} = \sum_{i=1}^{n}\frac{1}{(x - z_i)}$$

incluso en el caso de raíces múltiples. Luego $\left|\frac{f'(x)}{f(x)}\right| = \left|\sum_{i=1}^{n}\frac{1}{(x-z_i)}\right| \leq \sum_{i=1}^{n}\frac{1}{|x - z_i|}$

Además, como $|x - z_i| \geq \text{d}(z_i, \Gamma)$ para cada z_i, entonces $\frac{1}{|x-z_i|} \leq \frac{1}{\text{d}(z_i,\Gamma)}$ y $\sum_{i=1}^{n}\frac{1}{|x-z_i|} \leq \sum_{i=1}^{n}\frac{1}{\text{d}(z_i,\Gamma)} = \kappa_f(\Gamma)$. Encadenando las desigualdades se concluye. □

Lema 10. *Si x, y son números complejos cuya diferencia de argumentos es α, entonces*

$$|x - y| \leq |x| + |y| - 2\,\text{mín}(|x|, |y|)\left(1 - \text{sen}\left(\frac{\alpha}{2}\right)\right)$$

Demostración. Supongamos que $\text{mín}(|x|, |y|) = |y|$, es decir, $|y| \leq |x|$. El caso contrario es similar. Consideremos el paralelogramo formado por el origen, x, $x+y$

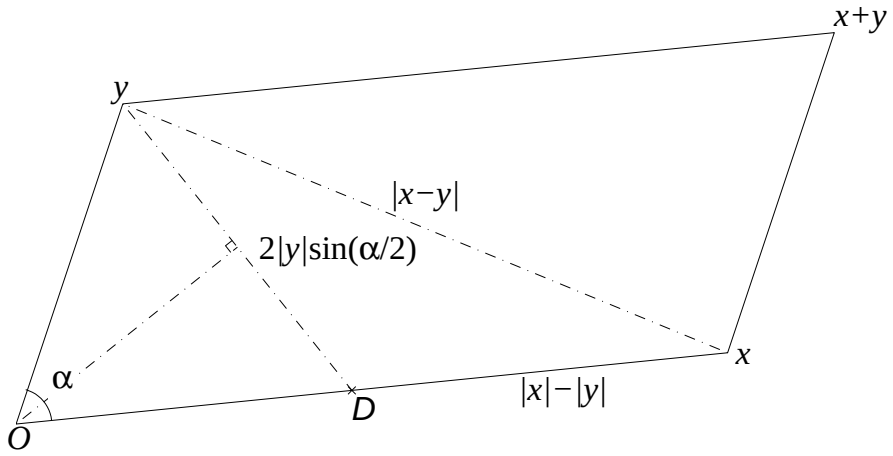


Figura 3.5: El punto D está a la misma distancia de O que y.

y y (figura 3.5). Resolviendo el triángulo formado por el origen, y, y el punto D en el segmento $\overline{Ox}$ a la misma distancia de O que y, se ve que su lado $\overline{yD}$ mide $2|y|\operatorname{sen}\left(\frac{\alpha}{2}\right)$ unidades. Los otros lados del triángulo D, x, y miden $|x-y|$ y $|x|-|y|$, luego por la desigualdad triangular:

$$|x-y| \leq |x|-|y|+2|y|\operatorname{sen}\left(\frac{\alpha}{2}\right) = |x|+|y|+2|y|\left(\operatorname{sen}\left(\frac{\alpha}{2}\right)-1\right)$$

por tanto $|x-y| \leq |x|+|y|-2|y|\left(1-\operatorname{sen}\left(\frac{\alpha}{2}\right)\right)$ como queríamos demostrar.

□

Proposición 3. *Supongamos que PCR se aplica a la curva Γ y al polinomio f de raíces $z_1, z_2, \ldots, z_n$. En una q-iteración en la que el valor insertado sea s_i, si $|s_{i+1}-s_i| < \delta$, entonces $\kappa_f(\Gamma) \geq \dfrac{2-\sqrt{2}}{4\delta}$.*

Demostración. Siendo una q-iteración, "$q_2(s_{i-1})$ y no $p(s_{i-1})$" es cierto en la secuencia $S = (\ldots, s_{i-1}, s_{i+1}, \ldots)$ antes de la inserción de s_i, luego

$$|f(\Gamma(s_{i-1}))| + |f(\Gamma(s_{i+1}))| \leq 2\,|f'(\Gamma(s_{i-1}))|\,(s_{i+1}-s_{i-1}) + |f(\Gamma(s_{i+1})) - f(\Gamma(s_{i-1}))|$$

Por el lema 10, siendo α la diferencia de argumentos de $f(\Gamma(s_{i-1}))$ y $f(\Gamma(s_{i+1}))$,

se verifica:

$$\begin{aligned}|f(\Gamma(s_{i+1})) - f(\Gamma(s_{i-1}))| \leq \\ \leq |f(\Gamma(s_{i-1}))| + |f(\Gamma(s_{i+1}))| - \\ -2\,\text{mín}(|f(\Gamma(s_{i-1}))|, |f(\Gamma(s_{i+1}))|)\left(1 - \text{sen}\left(\frac{\alpha}{2}\right)\right).\end{aligned}$$

Encadenando con la desigualdad $q_2(s_i)$ tenemos:

$$\begin{aligned}|f(\Gamma(s_{i-1}))| + |f(\Gamma(s_{i+1}))| \leq 2\,|f'(\Gamma(s_{i-1}))|\,(s_{i+1} - s_{i-1}) + \\ +|f(\Gamma(s_{i-1}))| + |f(\Gamma(s_{i+1}))| - 2\,\text{mín}(|f(\Gamma(s_{i-1}))|, |f(\Gamma(s_{i+1}))|)\left(1 - \text{sen}\left(\frac{\alpha}{2}\right)\right)\end{aligned}$$

es decir

$$2\,\text{mín}(|f(\Gamma(s_{i-1}))|, |f(\Gamma(s_{i+1}))|)\left(1 - \text{sen}\left(\frac{\alpha}{2}\right)\right) \leq 2\,|f'(\Gamma(s_{i-1}))|\,(s_{i+1} - s_{i-1}),$$

o

$$\frac{|f'(\Gamma(s_{i-1}))|}{\text{mín}(|f(\Gamma(s_{i-1}))|, |f(\Gamma(s_{i+1}))|)} \geq \frac{1 - \text{sen}\left(\frac{\alpha}{2}\right)}{s_{i+1} - s_{i-1}}$$

Por otra parte, por el lema 9

$$\kappa_f(\Gamma) \geq \left|\frac{f'(\Gamma(s_{i-1}))}{f(\Gamma(s_{i-1}))}\right| \geq \left|\frac{f'(\Gamma(s_{i-1}))}{\text{mín}(|f(\Gamma(s_{i-1}))|, |f(\Gamma(s_{i+1}))|)}\right|$$

luego $\kappa_f(\Gamma) \geq \dfrac{1 - \text{sen}(\alpha/2)}{s_{i+1} - s_{i-1}}$. Además, por "no $p(s_{i-1})$", $f(\Gamma(s_{i-1}))$ y $f(\Gamma(s_{i+1}))$ están en sectores adyacentes, lo que implica que su diferencia de argumentos α es menor que $\pi/2$ y entonces $\text{sen}(\alpha/2) < \text{sen}(\pi/4) = \sqrt{2}/2$ y $1 - \text{sen}(\alpha/2) > 1 - \sqrt{2}/2$. También siendo s_i el punto medio de s_{i-1} y s_{i+1}, $s_{i+1} - s_{i-1} = s_{i+1} - s_i + s_i - s_{i-1} < \delta + \delta$. Luego $\kappa_f(\Gamma) \geq \dfrac{2 - \sqrt{2}}{4\delta}$.

□

La precisión y el coste de PCR están descritos en el siguiente teorema:

Teorema 6. *Si $\Gamma : [a, b] \to \mathbb{C}$ está uniformemente parametrizada, S es una secuencia muestreo de $[a, b]$, Q un real positivo, y f un polinomio de grado n, el procedimiento de inserción con control de proximidad de raíz,* PCR*, verifica:*

a) Retorna en menos de $\left\lfloor \frac{b-a}{Q} + 1 \right\rfloor$ *iteraciones.*

b) Si retorna normalmente la secuencia retornada nos da el número de raíces de f *dentro de* Γ.

c) Si retorna con error, entonces $\kappa_f(\Gamma) \geq \frac{2-\sqrt{2}}{4Q}$.

Demostración. Para a), notemos que el predicado r causa una salida si hay una inserción en S de un valor del parámetro que está a menos de Q de otro valor en S. El número máximo de valores en un intervalo $[a, b]$ a una distancia mayor o igual que Q es precisamente $\left\lfloor \frac{b-a}{Q} + 1 \right\rfloor$. Luego este es el máximo número de iteraciones posibles.

Para b), en una salida normal tenemos una secuencia S que verifica "no $p(s_i)$ y no $q_2(s_i)$" para cada s_i. Consideremos el polígono formado por sus puntos imagen, que son conexos (no p). No tiene giros perdidos (por la proposición 1, no q_2 implica que no hay giro perdido). Luego el número de cruces del polígono por el semieje positivo de abscisas nos da el índice de $f(\Gamma)$, que es el número de raíces dentro de Γ.

Para c), en una salida con error el predicado $|s_{i+1} - s_i| < Q$ es cierto. Si la salida es de tipo p, por la proposición 2 $\kappa_f(\Gamma) \geq \frac{\operatorname{sen}(\pi/8)}{Q}$, y si es de tipo q, por la proposición 3 $\kappa_f(\Gamma) \geq \frac{2-\sqrt{2}}{4Q}$. En cualquier caso, como $(2-\sqrt{2})/4 < \operatorname{sen}(\pi/8)$, obtenemos la cota del teorema.

□

Si hay un retorno con error, el número de condición $\kappa_f(\Gamma)$ nos da una cota de la distancia desde Γ a la raíz de f más cercana, del siguiente modo:

Proposición 4. *En caso de retorno con error de PCR, hay una raíz* z_i *con* $\mathrm{d}(z_i, \Gamma) \leq \frac{4nQ}{2-\sqrt{2}}$.

Demostración. Como $\kappa_f(\Gamma) = \sum_{i=1}^{n} \frac{1}{\mathrm{d}(z_i, \Gamma)} \geq \frac{2-\sqrt{2}}{4Q}$ por el teorema 6 c) , al menos un sumando es mayor que la n-ésima parte, que es $\frac{1}{\mathrm{d}(z_i, \Gamma)} \geq \frac{2-\sqrt{2}}{4nQ}$ para algún i. Tomando inversos tenemos el resultado deseado.

□

En resumen, el procedimiento de inserción con control de proximidad de raíz evita un número excesivo de iteraciones, controlado por el parámetro de entrada Q. PCR calcula correctamente el número de raíces dentro de Γ si no hay raíces cercanas. Pero si hay raíces en el contorno Γ o cerca de él, el procedimiento puede retornar con error, indicando este hecho, o puede retornar normalmente con una secuencia válida para calcular el número de raíces, siempre en menos de $\left\lfloor \frac{b-a}{Q} - 1 \right\rfloor$ iteraciones.

Estas características son necesarias para utilizar PCR como componente en un método geométrico para hallar raíces. Así se evitan los inconvenientes, descritos en la introducción, de otras aproximaciones al problema de desarrollar un método geométrico. Estos inconvenientes son las curvas singulares (que pueden surgir en la partición de la zona de búsqueda en subregiones), y la necesidad de información global (como la constante de Lipschitz) para mostrar la corrección del cálculo.

Hemos abordado estos problemas con un número de condición κ_f asociado a cada curva. PCR detecta puntos de una curva que están próximos a una raíz, retornando con error en tal caso. Además la prueba de que se calcula en efecto el índice no usa información global.

Hemos mostrado que PCR retorna en menos de $\left\lfloor \frac{b-a}{Q} + 1 \right\rfloor$ iteraciones, ya sea normalmente o con error. Seguramente sea posible demostrar que, si el parámetro Q es lo suficientemente pequeño, PCR retorna normalmente en menos de $O(\kappa_f(\Gamma)^2 \log(\kappa_f(\Gamma)))$ iteraciones, un resultado similar al teorema 4. Debe notarse que esta dependencia de la distancia a las raíces es consistente con otros algoritmos que calculan el números de raíces en una región [Renegar, 1987, Pan, 1997]. En cualquier caso, este uso del número de condición κ_f para análisis teórico no es necesario para nuestro propósito de hallar raíces por métodos geométricos.

3.2. Descomposición recursiva de la región de búsqueda

En esta sección detallamos el método de descomposición, la segunda componente de nuestro método geométrico, según la pauta común que se describió en

la introducción. La primera componente de estos métodos es un test de inclusión, para decidir si hay una o más raíces en una región. La segunda es un procedimiento recursivo para subdividir la región y localizar las raíces con la precisión necesaria. Con el resultado del teorema 6, el procedimiento PCR calcula el número de raíces dentro de Γ mediante el índice de $\Delta = f(\Gamma)$. Este procedimiento se usará como test de inclusión, y debemos asegurar que retorna sin error. Es decir, PCR ha de invocarse sobre curvas que no pasen cerca de raíces, como indica el teorema anterior. Esto se interrelaciona con el proceso de división en subregiones, pues el borde de estas no puede pasar cerca de alguna raíz. Para exponer adecuadamente esta interrelación, se ha optado por describir tres construcciones del procedimiento recursivo, tres intentos progresivamente complejos, el último de los cuales es el definitivo (Tabla 3.1).

El tamaño de una región P se mide por su diámetro rectangular $\mathrm{dm}_R(P)$, que se define con precisión en la sección 3.2.1. Una *aproximación de una raíz con una precisión de* $A > 0$ es una región de diámetro rectangular menor que A que contiene a la raíz.

El Procedimiento Recursivo de División (PRec, figura 3.6) divide la región inicial en subregiones. Si el test de inclusión afirma que alguna de las subregiones contiene una o más raíces, esta es dividida a su vez, y así sucesivamente hasta que el diámetro cae por debajo de la precisión A. Las regiones finalmente obtenidas se insertan en la secuencia Π, que al final de la ejecución de PRec contiene aproximaciones a las raíces dentro de P. Cada región P_i incluida en Π está etiquetada con el número de raíces que contiene, n_i. En caso de que $n_i > 1$ se tiene una raíz múltiple, o un *cluster* (un conjunto de raíces situadas a distancia menor que A entre sí). Denotamos con P_I la región objetivo, dentro de la cual se buscan las raíces, y con P a una región genérica del plano.

La figura 3.6 es un bosquejo inicial de PRec, y falta especificar los subprocedimientos iTest(P) y Bisec(P). El primero, iTest, es el test de inclusión. Calcula el índice de la imagen por f del borde de P. Internamente, este cálculo es realizado aplicando PCR a esta imagen, y con cierto valor del parámetro Q, como se ha comentado en el apartado anterior. Así, si $\Gamma : [a, b] \to \mathbb{C}$ es el borde de P, denotamos con $\mathrm{PCR}(\Gamma, \theta)$ la aplicación del procedimiento de la figura 3.1, con el parámetro Q tomando el valor θ, para calcular el índice de $f(\Gamma)$. ¿Cuál es el valor θ que debe pasarse a PCR? El doble reto que nos fijamos al construir PRec es, por

Procedimiento Recursivo de División: Para encontrar las raíces de un polinomio f dentro de una región convexa P_I.

Parámetros de entrada: Una región convexa P_I del plano complejo, un polinomio f, y un parámetro de precisión $A > 0$.

Salida: La secuencia $\Pi = (P_1, P_2, \ldots, P_k)$ (conteniendo k aproximaciones con una precisión A de todas las raíces de f en P_I) y la secuencia $N = (n_1, n_2, \ldots, n_k)$ (de números $n_i \geq 1$ tal que cada P_i contiene n_i raíces de f contando su multiplicidad).

Procedimiento: PRec(P, A) {
  Si iTest$(P) = 0$, retornar Π y N vacías. [Salida 1]
  Si $\mathrm{dm}_R(P) < A$ entonces
    retornar $\Pi = (P)$ y $N =$ iTest(P). [Salida 2]
  Si no
    Bisec(P), que da dos subregiones P_0 y P_1.
    Para $i = 0$ y $i = 1$, hacer
      $(\Pi_i, N_i) =$ PRec(P_i, A).
  Retornar la concatenación de (Π_0, N_0) y (Π_1, N_1). [Salida 3]
}

Figura 3.6: Procedimiento Recursivo de División (PRec) para hallar raíces de polinomios. Falta por especificar dos subprocedimientos: iTest(P) y la descomposición Bisec(P).

un lado, asegurar que el test de inclusión solo se invoca en curvas que no retornen con error en PCR y, por otro lado, tener un coste computacional asequible. La determinación de estas curvas es el resultado de un proceso iterativo de prueba y error con un número acotado de iteraciones. Una contribución importante de este trabajo es usar los resultados del teorema 6 para encontrar estas curvas. Por el teorema 6*c)*, cuando no hay raíces a menos de $(4 + 2\sqrt{2})n\theta$ de Γ, PCR(Γ, θ) no retorna con error[1]. Por tanto usar un valor θ lo bastante pequeño es una condición suficiente para evitar la influencia de cualquier raíz que pudiera causar un error en PCR (figure 3.7). Sin embargo mientras menor sea el valor de θ mayor es el coste computacional, por el teorema 6*a*). Precisaremos más adelante (en la proposición 8) qué valor específico se escoge para el parámetro θ. Como este valor es diferente

[1]Aunque el retorno con error de PCR nos da una aproximación de una raíz, hemos preferido ignorar tal modo de hallar raíces, para no introducir casos especiales en PRec y mantenerlo tan regular como sea posible.

para cada curva Γ, lo denotamos con Z_Γ.

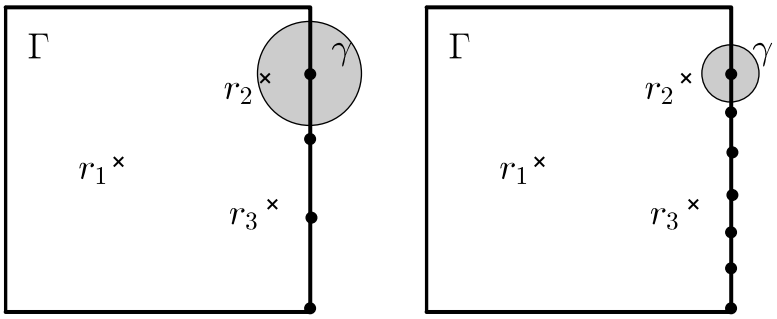


Figura 3.7: En la primera figura, PCR usa un valor grande de θ. La raíz r_2 está lo bastante cerca del punto insertado γ como para causar un error. r_3 también está lo bastante cerca, pero se da el caso de que no produce error. En la segunda figura, PCR usa un valor pequeño de θ, con lo que evita el error producido por r_2. Sin embargo esta solución trae aparejada una computación más costosa: mientras menor sea θ, más muestras de Γ hay que evaluar.

El segundo subprocedimiento, Bisec(P), crea dos subregiones P_0 y P_1 a partir de P, y ha de cumplir los tres requisitos siguientes: a) debe ser una partición de P (es decir, $P = P_0 \cup P_1$ y $P_0 \cap P_1 = \emptyset$), para que cada raíz contenida en P esté en un P_i y solo en uno, b) el diámetro debe ser decreciente, $\mathrm{dm}_R(P_i) < \mathrm{dm}_R(P)$, y c) cada borde Γ_i de P_i debe admitir un valor Z_{Γ_i} tan grande como sea posible de modo que PCR(Γ_i, Z_{Γ_i}) calcula, sin error, el índice de Γ_i.

Desde luego el requisito c) incluye evitar curvas que pasen por una raíz, que tienen un coste infinito. ¿Hay algún método de partición que satisfago los requisitos anteriores? La búsqueda de tal método Bisec vertebra esta sección. En la primera subsección 3.2.1 detallamos como se aplican los cortes rectos a las curvas parametrizadas que constituyen los bordes. En la siguiente subsección 3.2.2 hacemos un primer intento de definir Bisec. Cortamos a la mitad la región, primero en horizontal y luego en vertical, y así sucesivamente. Sin embargo este primer intento tiene un inconveniente: los cortes pueden pasar muy cerca de una raíz, incluso por encima, produciendo curvas en las que PCR tiene alto coste computacional, incluso retorna con error. En la siguiente subsección 3.2.3 hacemos un segundo intento, usando el Cortes Iterados con Desplazamiento. El método CID hace el corte (horizontal o vertical) no en el medio exacto de la región, sino con

el desplazamiento adecuado para que las curvas resultantes eviten la proximidad de raíces, posibilitando que se pueda usar un parámetro θ lo bastante grande para tener bajo coste computacional, y aún así calcular sin error su índice.

Lamentablemente, cuando se desplazan los cortes alternativamente horizontales y verticales, se pueden producir regiones cuyo diámetro no decrece suficientemente tras varias subdivisiones. Es decir el requisito *c)* anterior se cumple pero el *b)* no. Para satisfacer los tres requisitos se introduce el corte a lo largo del eje menor, en el tercer y definitivo intento en la subsección 3.2.4, en vez de alternar cortes horizontales y verticales. En cualquier caso, las subregiones se producen por un corte recto, por lo que el requisito *a)* se cumple en los tres intentos. La tabla 3.1 muestra las tres versiones progresivas de la función de partición Bisec. Hemos decidido presentar sucesivamente los tres intentos con el propósito de desplegar gradualmente la solución al problema.

Requisito / Intento	*a)* Partición	*b)* Diámetro decreciente	*c)* PCR de bajo coste sin error
1) Corte central Alternando H y V	Sí	Sí	No
2) CID Alternando H y V	Sí	No	Sí
3) CID Por el eje menor	Sí	Sí	Sí

Tabla 3.1: Requisitos verificados por intentos sucesivos del método Bisec.

Una vez logrado el tercer y último intento de definición de Bisec(P), en la sección 3.3 se demuestra que en efecto se producen subregiones satisfaciendo los tres requisitos. Finalmente la sección 3.4 prueba que PRec acaba en un número finito de llamadas recursivas, y que la afirmación sobre la salida de PRec es correcta. También determinamos su coste computacional.

Antes de seguir, un comentario sobre el tipo de regiones P que se van a conside-

rar. Por simplicidad, la región inicial, así como las que surgen de las subdivisiones, deben ser arcoconexas. Con este supuesto, su borde es una curva simple cerrada Γ (curva de Jordan [Kolmogorov and Fomın, 1975]). Una región no arcoconexa es, por ejemplo, el interior de dos círculos disjuntos. Regiones no arcoconexas, cuyo borde está formado por varias curvas cerradas, se pueden tratar de un modo similar a las arcoconexas, porque el número de raíces dentro de una región no arcoconexa es la suma de los índices de estas curvas cerradas. Sin embargo el supuesto de arcoconexidad simplifica los razonamientos.

Bisec(P) va a producir las subregiones mediante cortes a lo largo de líneas rectas. Para impedir que tales cortes produzcan una subregión no arcoconexa, imponemos que la región inicial P_I sea convexa (es decir, que si un segmento recto tiene sus extremos en P_I, todo el segmento está dentro de P_I). Por tanto las subregiones serán también convexas (y también arcoconexas) porque están producidas por cortes rectos. Desde un punto de vista práctico, al hallar las raíces en una región no convexa, PRec debe aplicarse a su envolvente convexa, o a una descomposición en partes convexas.

3.2.1. Particionado por cortes rectos

Antes de desarrollar los tres sucesivos intentos, vamos a describir cómo se calcula el borde de las subregiones a partir del borde de la región original. Consideramos el caso de un corte horizontal, pero un corte vertical es similar.

En el estudio de curvas parametrizadas es frecuente considerar la traslación del parámetro, y la concatenación de dos curvas (puede verse por ejemplo [Do Carmo, 1976]). Ahora recordamos estos conceptos, que vamos a extender a muestreos, definiendo la traslación de un muestreo y la concatenación de dos muestreos.

Dada una curva $\Gamma : [a, b] \to \mathbb{C}$, su *traslación por* $c \in \mathbb{R}$, denotada por $\Gamma + c$, es la curva $\Gamma + c : [a + c, b + c] \to \mathbb{C}$ definida como $(\Gamma + c)(u) = \Gamma(u - c)$ para cada $u \in [a + c, b + c]$. Las curvas Γ y $\Gamma + c$ tienen la misma imagen, es decir, son dos parametrizaciones del mismo conjunto de puntos (figura 3.8).

Si S es un muestreo de Γ (es decir, una secuencia $S = (a = s_0, s_1, \ldots, s_m = b)$ de parámetros crecientes $s_i \in [a, b]$), entonces $S + c$, definido como $S + c = (s_0 + c, s_1 + c, \ldots, s_m + c)$, es un muestreo de $\Gamma + c$.

Si dos segmentos de curva Γ_1, Γ_2 tienen intervalos paramétricos $[a_1, b_1]$, $[a_2, b_2]$

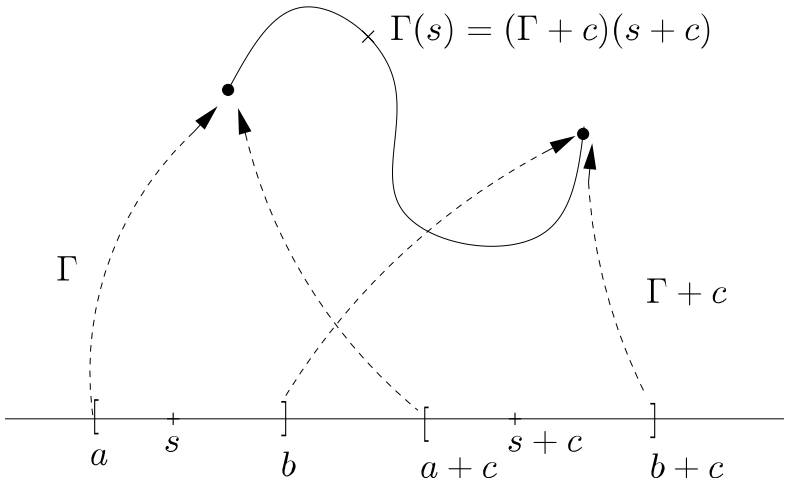


Figura 3.8: Las curvas Γ y $\Gamma + c$.

respectivamente, con $\Gamma_1(b_1) = \Gamma_2(a_2)$ (el final de una curva coincide con el inicio de la otra), entonces se define su *concatenación*, denotada por $\Gamma_1;\Gamma_2$, como la curva $\Gamma_1;\Gamma_2 : [a_1, b_1 + (b_2 - a_2)] \to \mathbb{C}$ con la siguente expresión paramétrica :

$$\Gamma_1;\Gamma_2(u) = \begin{cases} \Gamma_1(u) & \text{si } u \in [a_1, b_1] \\ \Gamma_2(u - (b_1 - a_2)) & \text{si } u \in [b_1, b_1 + (b_2 - a_2)] \end{cases}$$

Puede verse como Γ_1 seguida de una traslación de Γ_2 por la cantidad adecuada $c = b_1 - a_2$ como para que los intervalos paramétricos de Γ_1 y $\Gamma_2 + c$ sean consecutivos (figura 3.9). Notemos que en esta figura el desplazamiento c, por el que Γ_2 tiene que trasladarse para encajar con Γ_1, es negativo, mientras que en la figura 3.8 el desplazamiento c es positivo. Ambos son ejemplos igualmente válidos de traslación.

Si S_{Γ_1} y S_{Γ_2} son secuencias de muestreo de Γ_1 y Γ_2 respectivamente (es decir $S_{\Gamma_1} = (a_1 = s_0, s_1, \ldots, s_{m_1} = b_1)$ y $S_{\Gamma_2} = (a_2 = s'_0, s'_1, \ldots, s'_{m_2} = b_2)$), entonces la concatenación (como secuencias) S_{Γ_1} y $S_{\Gamma_2} + (b_1 - a_2)$ es una secuencia de muestreo de $\Gamma_1;\Gamma_2$. La concatenación de secuencias $(a, \ldots, b)$ y $(c, \ldots, d)$ es $(a, \ldots, b, c, \ldots, d)$. Es un detalle menor el que, como siempre vamos a concatenar secuencias tales que la primera acaba en el mismo valor que la segunda empieza (es decir $b = c$), podríamos haber considerado $(a, \ldots, b, \ldots, d)$ como la secuencia concatenada. Esto no implica ninguna diferencia cuando comprobemos las proposiciones p, q, q_2 o r, o insertemos valores de interpolación en secuencias de paráme-

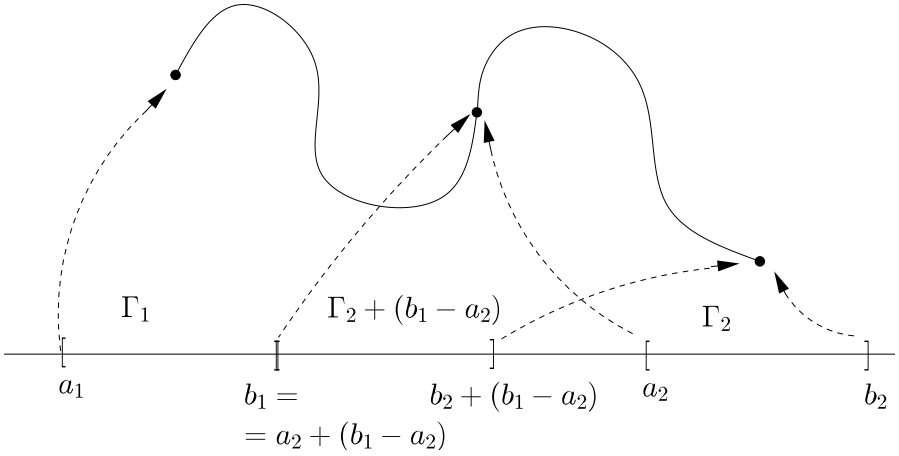


Figura 3.9: La concatenación $\Gamma_1; \Gamma_2$ es Γ_1 seguida por $\Gamma_2 + (b_1 - a_2)$, tiene de intervalo paramétrico $[a_1, b_1 + (b_2 - a_2)]$.

tros.

Ahora definimos las lineas, horizontales o verticales, a lo largo de las cuales haremos los cortes. Consideramos cada región plana como un conjunto cerrado, es decir, incluyendo su borde. Para cada región plana P, su *línea de soporte horizontal superior* l_T es la recta horizontal más alta que tiene algún punto en común con P. Del mismo modo, la *línea de soporte horizontal inferior* l_B de P es la línea recta horizontal más baja que tenga algún punto en común con P. El *diámetro vertical* $\mathrm{dm}_V(P)$ es la distancia entre estas líneas $\mathrm{d}(l_T, l_B)$. Es decir $\mathrm{dm}_V(P) = \min_{x \in l_T,\, y \in l_B} \mathrm{d}(x, y)$.

De modo similar, la *línea de soporte vertical izquierda* l_L es la línea recta vertical más a la izquierda que tiene algún punto en común con P, y la *línea de soporte vertical derecha* l_R es la línea recta vertical más a la derecha que tiene algún punto en común con P. El *diámetro horizontal* $\mathrm{dm}_H(P)$ es la distancia entre estas líneas $\mathrm{d}(l_L, l_R)$. Es decir $\mathrm{dm}_H(P) = \min_{x \in l_L,\, y \in l_R} \mathrm{d}(x, y)$. La figura 3.10 ilustra estas definiciones, y también el *diámetro* clásico[2] de P definido como $\mathrm{dm}(P) = \max_{x,y \in P} \mathrm{d}(x, y)$, y el *diámetro rectangular* $\mathrm{dm}_R(P) = \sqrt{\mathrm{dm}_H(P)^2 + \mathrm{dm}_V(P)^2}$.

Para una región P de borde Γ con intervalo paramétrico $[a, b]$, denotamos las dos subregiones que surgen de P por un corte horizontal a lo largo de $m_H(P)$

[2]Para regiones definidas por segmentos rectos, como los polígonos que surgen en PCR, los diámetros horizontal y vertical son más fáciles de calcular que el diámetro clásico.

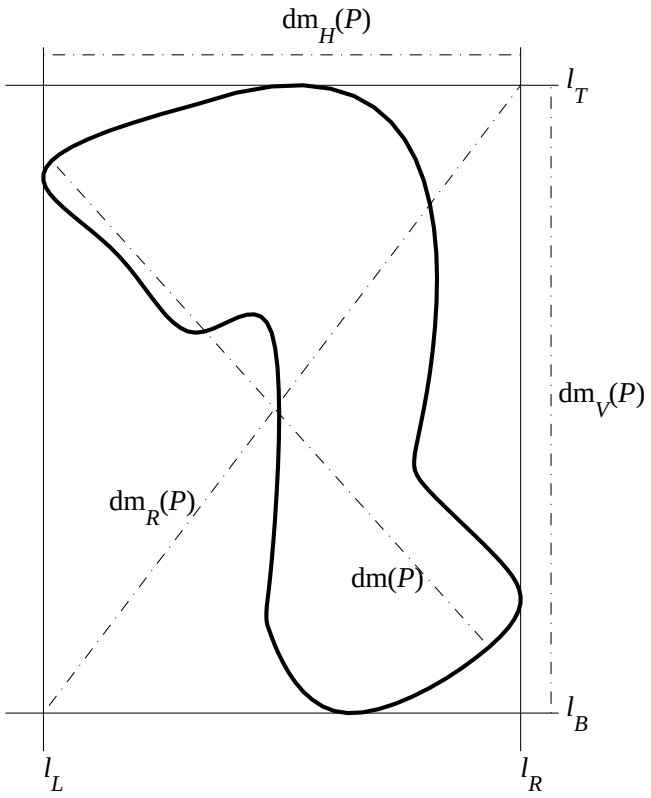


Figura 3.10: Líneas de soporte y diámetros para una región plana P.

como $T(P)$ y $B(P)$, y sus bordes respectivos como Γ_T y Γ_B. Detallaremos la parametrización de estos bordes usando la traslación y la concatenación de las parametrizaciones de Γ y $m_H(P)$. Sea $\Upsilon : [0, d] \to \mathbb{C}$ el segmento de recta perteneciente a $m_H(P)$ que está contenido en P. Esta parametrización Υ esta hecha a partir de los puntos de corte c_1 y c_2 de $m_H(P)$ con Γ, como $\Upsilon(u) = (1-u)c_1 + uc_2$, siendo c_1 el punto con menor parte real (figura 3.11). La curva Γ_T está hecha concatenando un segmento de curva de Γ (su mitad superior) y Υ. Del mismo modo, la curva Γ_B está hecha concatenando la curva Υ recorrida en la dirección opuesta (denotado $\Upsilon^- : [0, d] \to \mathbb{C}$) con el otro segmento de curva de Γ (su mitad inferior). La expresión de Υ^- a partir de Υ es $\Upsilon^-(u) = \Upsilon(d - u)$. Si el punto extremo $\Gamma(a) = \Gamma(b)$ está por debajo de $m_H(P)$, la mitad superior de Γ es el segmento de curva $\Gamma_{[s,t]}$ comprendido entre los dos valores del parámetro s, t con $a \leq s < t \leq b$ y $c_1 = \Gamma(t), c_2 = \Gamma(s)$ (figura 3.12). La mitad inferior es la concatenación $\Gamma_{[t,b]}; \Gamma_{[a,s]}$. Por tanto $\Gamma_T = \Upsilon; \Gamma_{[s,t]}$ y $\Gamma_B = \Upsilon^-; \Gamma_{[t,b]}; \Gamma_{[a,s]}$ (figure 3.13). En el caso de que

el punto extremo $\Gamma(a) = \Gamma(b)$ esté por encima o en $m_H(P)$, los bordes de las subregiones se obtienen de modo parecido.

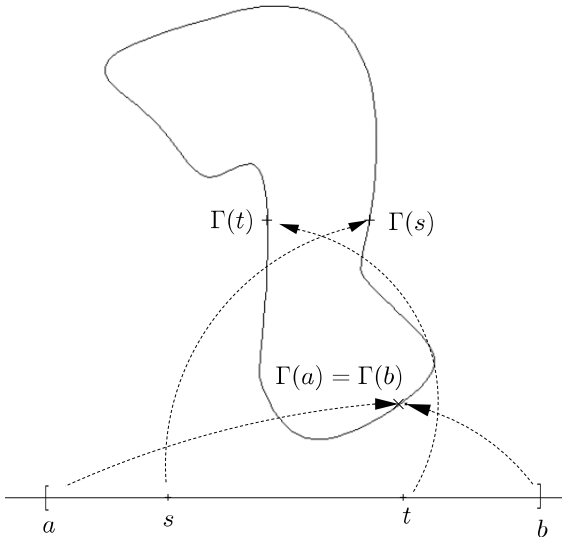


Figura 3.11: Las flechas muestran la parametrización Γ. Se marcan los puntos de corte $c_1 = \Gamma(t)$ y $c_2 = \Gamma(s)$ de una partición horizontal.

El método para encontrar los parámetros s y t (correspondientes a los puntos de corte c_1 y c_2 de $m_H(P)$ con Γ) depende de la expresión de Γ. Explicaremos más adelante cómo se haría en el caso de que Γ esté construida concatenando segmentos de recta uniformemente parametrizados. Esta es una manera sencilla de expresar curvas, adecuado para describir diversas regiones, que usaremos para implementar PRec como se detalla en [Cortés-Fácila et al., 2014]. Los resultados expuestos en esta memoria sobre partición en subregiones son igualmente válidos para otras formas de expresar las curvas (por ejemplo usando parametrizaciones no uniformes, o segmentos no rectos), siempre y cuando haya un método para encontrar los parámetros de los puntos de corte.

Supongamos que Γ es la concatenación de k segmentos rectos de parametrizaciones $\Sigma_j : [a_j, b_j] \longrightarrow \mathbb{C}$, para $j = 0, \ldots, k-1$, en cuyo caso $\Sigma_j(b_j) = \Sigma_{j+1}(a_{j+1})$. Γ es cerrada si además $\Sigma_{k-1}(b_{k-1}) = \Sigma_0(a_0)$. De esta manera $\Gamma = \Sigma_0; \Sigma_1; \cdots; \Sigma_{k-1}$, y su intervalo paramétrico es $[a, b] = [a_0, \sum_{j=0}^{k-1}(b_j - a_j)]$. Llamamos $e_j = \Sigma_j(a_j)$ a los vértices de la curva poligonal Γ (figura 3.14). También suponemos que la secuencia de muestreo $S_\Gamma = (a = s_0, \ldots, s_m = b)$ de Γ contiene los parámetros de los vértices

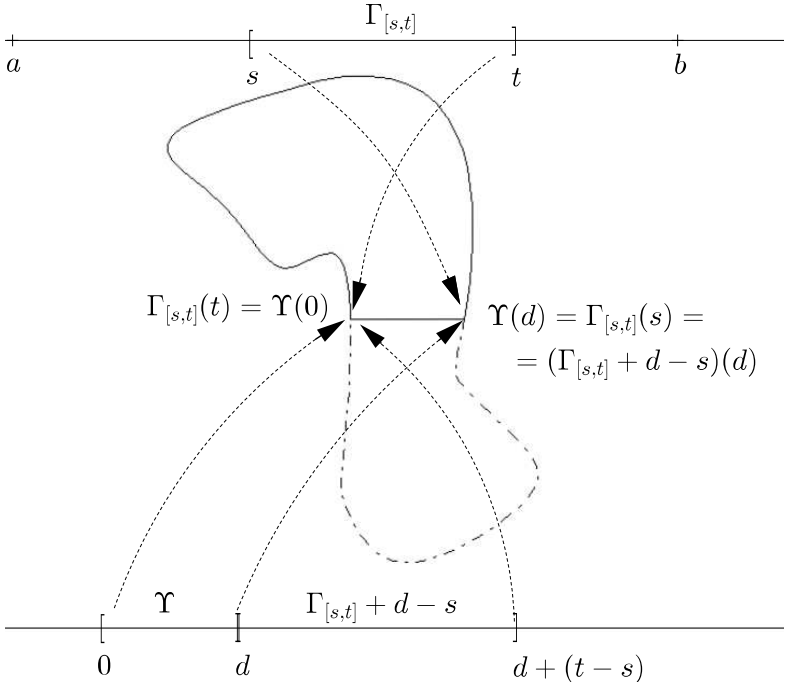


Figura 3.12: Γ_T es la concatenación de Υ y $\Gamma_{[s,t]}$. La traslación de la segunda curva es $\Gamma_{[s,t]} + (d - s)$.

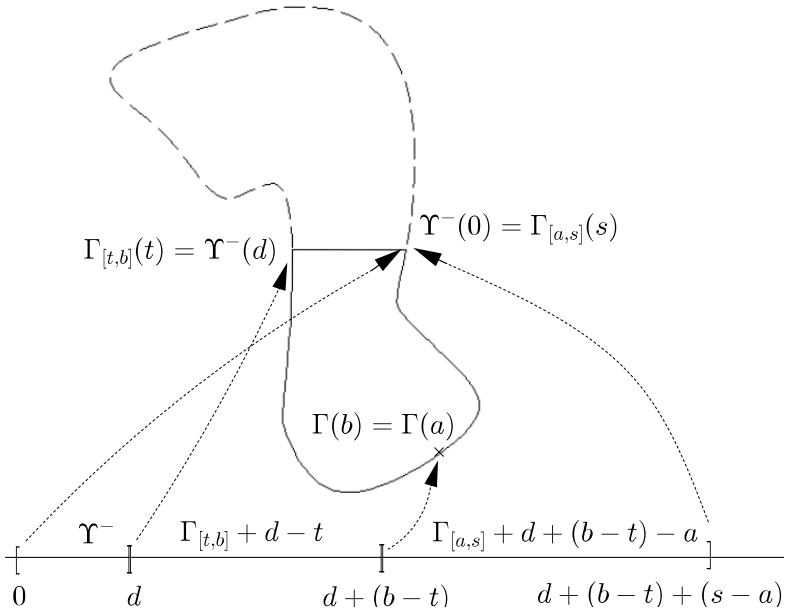


Figura 3.13: Γ_B es $\Upsilon^-; \Gamma_{[b,t]}; \Gamma_{[a,s]}$. Las traslaciones hechas para esta concatenación son $\Gamma_{[b,t]} + d - t$ y $\Gamma_{[a,s]} + d + (b - t) - a$.

e_j. Puede comprobarse que en la parametrización $\Sigma_0; \Sigma_1; \cdots; \Sigma_{k-1} : [a, b] \longrightarrow \mathbb{C}$ de Γ, el parámetro del vértice e_j es $\sum_{i=0}^{j}(b_i - a_i)$, aunque no usaremos ese detalle en la discusión.

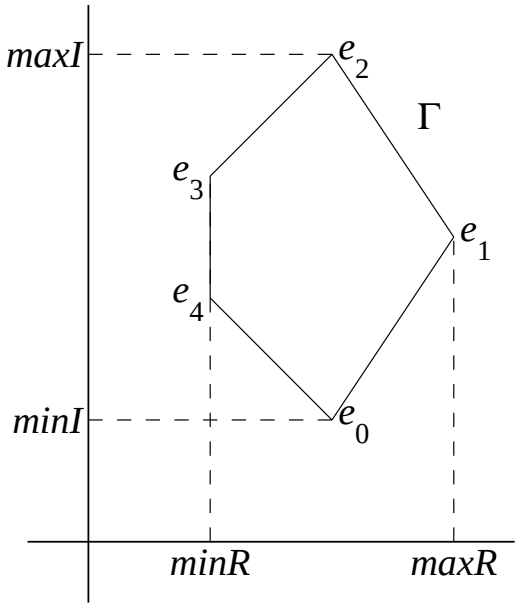


Figura 3.14: La curva Γ es la concatenación de los segmentos rectos $\overline{e_j, e_{j+1}}$ uniformemente parametrizados. El muestreo S de Γ incluye el parámetro de e_j, con j variando desde 0 hasta $k-1$, siendo k el número de vértices, y $e_k = e_0$.

Con Γ y su secuencia de muestreo S_Γ construida de esta manera, poligonalmente, calculamos primero los diámetros. Llamamos $minR$ y $maxR$ a el mínimo y el máximo, respectivamente, de la parte real de Γ. De modo parecido, $minI$ y $maxI$ son el mínimo y el máximo de la parte imaginaria. El diámetro horizontal mide $maxR - minR$, y el vertical $maxI - minI$. Los extremos $minR$, $maxR$, $minI$ y $maxI$ se alcanzarán en puntos que son un extremo e_j de los segmentos Σ_j que componen Γ, y los pará,etros de estos extremos están incluidos en la secuencia S_Γ (figura 3.14), por tanto podemos encontrar estos extremos recorriendo la secuencia.

La división en subregiones Bisec(P) calcula la línea media m_H o m_V. Consideramos el caso de una división horizontal, siendo similar la vertical. Usando la media $mI = \dfrac{minI + maxI}{2}$, recorremos S_Γ separando los puntos con parte imaginaria mayor o igual qie mI (con los que componemos el borde de $T(P)$) de aquellos con

parte imaginaria menor o igual que mI (para componer $B(P)$). Como mucho dos puntos del borde Γ tienen parte imaginaria exactamente mI, porque la línea m_H corta a Γ en dos puntos, por la convexidad de P. Es decir $m_H \cap \Gamma = \{c_1, c_2\}$. Cada c_k, $k = 1, 2$ se calcula a partir de un par de puntos consecutivos de S_Γ, uno de los cuales está encima de mI y el otro bajo ella, $\Gamma(s_i)$ y $\Gamma(s_{i+1})$ en la figura 3.15. Si estos puntos de corte c_k no están en S_Γ, los insertamos en las secuencias correspondientes a los bordes de $T(P)$ y $B(P)$. La interpolación lineal para hallar c_k, si $\Gamma(s_i) = (x_i, y_i)$ y $\Gamma(s_{i+1}) = (x_{i+1}, y_{i+1})$, es $c_k = \left(mI, \dfrac{mI - x_{i+1}}{x_i - x_{i+1}}(y_i - y_{i+1}) + y_{i+1} \right)$ (en el caso de una división horizontal). Por la parametrización uniforme del segmento Σ_j que contiene c_1 o c_2, sus parámetros s o t se pueden hallar también por interpolación lineal. De este modo construimos el segmento Υ uniendo c_1 y c_2, y los bordes de las subregiones $\Gamma_T = \Upsilon; \Gamma_{[s,t]}$ y $\Gamma_B = \Upsilon^-; \Gamma_{[t,b]}; \Gamma_{[a,s]}$. Resumiendo, este es el método para hacer cortes rectos a P a lo largo de líneas horizontales y verticales, si P es poligonal.

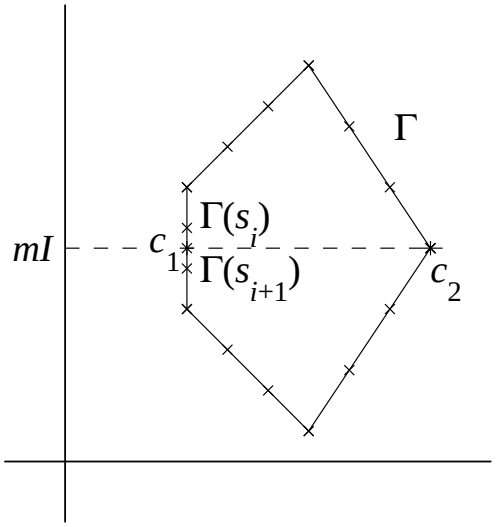


Figura 3.15: La curva Γ tiene dos puntos a altura mI. Las coordenadas de c_1 y su parámetro s se hallan por interpolación lineal de $\Gamma(s_i)$ y $\Gamma(s_{i+1})$. De modo similar con c_2. El parámetro de c_2 en este ejemplo casualmente pertenece a S_Γ.

3.2.2. Primer intento: alternando cortes horizontales y verticales

Siguiendo con la tabla 3.1, vamos a mostrar que los cortes a lo largo de la línea media alternando horizontal y vertical producen subregiones que cumplen el requisito *b)*, es decir, que tienen diámetros decrecientes. Dada una región genérica P, recordemos que el diámetro vertical $\mathrm{dm}_V(P)$ es la altura de P y el diámetro horizontal $\mathrm{dm}_H(P)$ es su ancho (mostrados en la figura 3.10, y también que el diámetro rectangular es $\mathrm{dm}_R(P) = \sqrt{\mathrm{dm}_H(P)^2 + \mathrm{dm}_V(P)^2}$ y el diámetro clásico $\mathrm{dm}(P) = \max_{x,y\in P} \mathrm{d}(x,y)$). Se relacionan por las siguientes cotas, que necesitaremos para demostrar que el primer intento de Bisec produce diámetros decrecientes.

Lema 11. *Si dos regiones planas P_1, P_2 verifican $P_1 \subset P_2$, entonces* $\mathrm{dm}(P_1) \leq \mathrm{dm}(P_2)$, $\mathrm{dm}_H(P_1) \leq \mathrm{dm}_H(P_2)$, *y* $\mathrm{dm}_V(P_1) \leq \mathrm{dm}_V(P_2)$.

Demostración. Para el diámetro clásico es evidente porque el máximo en la definición de $\mathrm{dm}(P_2)$ se toma en un conjunto mayor que en la de $\mathrm{dm}(P_1)$. Para el diámetro horizontal, notemos que las dos líneas de soporte verticales de P_1 están entre las dos líneas de soporte verticales de P_2, quizás coincidiendo con alguna de ellas. Por lo tanto su distancia es menor. Lo mismo sucede con el diámetro vertical. □

Proposición 5. *Para una región plana P, se verifica:*

$$\max(\mathrm{dm}_H(P), \mathrm{dm}_V(P)) \leq \mathrm{dm}(P) \leq \mathrm{dm}_R(P)$$

Demostración. Para la primera desigualdad, notemos que las líneas de soporte de P tienen al menos un punto en común con P. Siendo x_0 uno de estos puntos de la línea de soporte horizontal superior, es decir $x_0 \in P \cap l_T$, y y_0 para la inferior, $y_0 \in P \cap l_B$, utilizando las definiciones de $\mathrm{dm}_V(P)$ y $\mathrm{dm}(P)$ tenemos que $\mathrm{dm}_V(P) = \min_{x\in l_T,\, y\in l_B} \mathrm{d}(x,y) \leq \mathrm{d}(x_0,y_0) \leq \max_{x,\, y\in P} \mathrm{d}(x,y) = \mathrm{dm}(P)$. Asimismo $\mathrm{dm}_H(P) \leq \mathrm{dm}(P)$, luego $\max(\mathrm{dm}_H(P), \mathrm{dm}_V(P)) \leq \mathrm{dm}(P)$.

Para la segunda desigualdad, definimos la *HV-envolvente* de P, $\mathrm{Env}_{HV}(P)$, como el rectángulo delimitado por las líneas de soporte horizontales y verticales. Como $P \subset \mathrm{Env}_{HV}(P)$, por el lema 11 tenemos que $\mathrm{dm}(P) \leq \mathrm{dm}(\mathrm{Env}_{HV}(P))$. Además, el diámetro del rectángulo $\mathrm{Env}_{HV}(P)$, de base $\mathrm{dm}_H(P)$ y altura $\mathrm{dm}_V(P)$,

es la distancia entre vértices opuestos, luego $\mathrm{dm}(\mathrm{Env}_{HV}(P)) = \sqrt{\mathrm{dm}_H(P)^2 + \mathrm{dm}_V(P)^2} = \mathrm{dm}_R(P)$. Encadenando con la desigualdad anterior, se concluye.

□

Para dividir una figura en partes menores, definimos los operadores T, B, L y R que actúan en una región plana P. Si $m_H(P)$ es la línea recta equidistante de las líneas de soporte horizontales, $T(P)$ es la intersección de P con el semiplano superior definido por $m_H(P)$, y $B(P)$ es la intersección de P con el semiplano inferior. Del mismo modo, si $m_V(P)$ es la línea recta equidistante de las líneas verticales de apoyo, $L(P)$ es la intersección de P con el semiplano izquierdo del plano definido por $m_V(P)$, y $R(P)$ es la intersección de P con el semiplano derecho. Los operadores T y B se dice que son *de tipo horizontal*, mientras que L y R son *de tipo vertical*. La figura 3.16 muestra varias composiciones de estos operadores aplicados a una región no convexa.

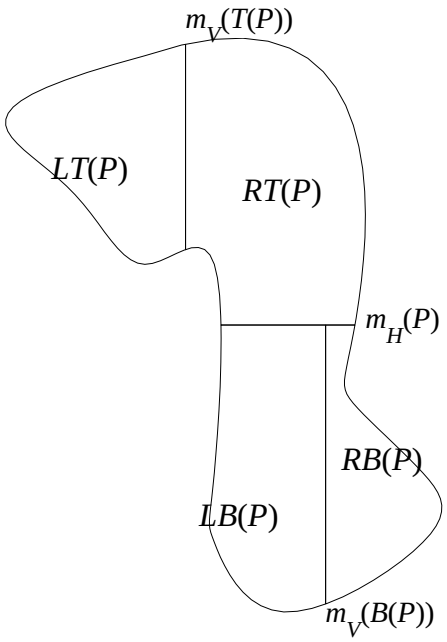


Figura 3.16: Aplicaciones de los operadores T, B, L, R to P.

Vamos a demostrar que aplicando los operadores T, B, L y R, alternando horizontales y verticales, las regiones que se producen tienen diámetros decrecientes.

Lema 12. *Los operadores hacen decrecer el diámetro, verificando:*

$$\mathrm{dm}_H(T(P)) \leq \mathrm{dm}_H(P), \quad \mathrm{dm}_H(L(P)) \leq \frac{\mathrm{dm}_H(P)}{2},$$
$$\mathrm{dm}_V(T(P)) \leq \frac{\mathrm{dm}_V(P)}{2} \quad y \ \mathrm{dm}_V(L(P)) \leq \mathrm{dm}_V(P)$$

Las mismas fórmulas son válidas cambiando T por B y L por R.

Demostración. Para el diámetro horizontal, la primera desigualdad es consecuencia del lema 11. La segunda proviene de que las líneas de soporte vertical de $L(P)$ están entre l_L y $m_V(P)$, luego el diámetro horizontal es menor o igual que la distancia $\mathrm{d}(l_L, m_V(P))$, que es la mitad de $\mathrm{dm}_H(P)$. Para el diámetro vertical, el razonamiento es similar, y también para los operadores B y R.

□

Vamos a hallar la tasa de decrecimiento de diámetro cuando aplicamos alternativamente divisiones horizontales y verticales a la región inicial. Un operador es *de tipo* $\mathcal{O}_m$, para $m \geq 1$, si es la composición de m operadores de tipo horizontal alternándose con m de tipo vertical, aplicando primero uno de los horizontales. Es decir, O es de tipo $\mathcal{O}_m$ si $O = V_m H_m V_{m-1} H_{m-1} \cdots V_1 H_1$, siendo $H_i \in \{T, B\}$ y $V_i \in \{L, R\}$. Hay 2^{2m} operadores de tipo $\mathcal{O}_m$. (Véase la figura 3.17).

Lema 13. *Para $m \geq 1$, si O_m es un operador de tipo $\mathcal{O}_m$*

$$\mathrm{dm}_R(O_m(P)) \leq \frac{\mathrm{dm}_R(P)}{2^m}$$

Demostración. Por inducción en m. Denotamos como antes con H un operador que puede ser T o B y con V otro que puede ser L o R. Para $m = 1$, tenemos que $O = VH$. Encadenando algunas desigualdades del lema 12 varias veces:

$$\mathrm{dm}_R(VH(P)) = \sqrt{\mathrm{dm}_H(VH(P))^2 + \mathrm{dm}_V(VH(P))^2} \leq$$
$$\leq \sqrt{\frac{\mathrm{dm}_H(H(P))^2}{2^2} + \mathrm{dm}_V(H(P))^2} \leq \frac{\sqrt{\mathrm{dm}_H(P)^2 + \mathrm{dm}_V(P)^2}}{2} = \frac{\mathrm{dm}_R(P)}{2}$$

Para $m > 1$, notemos que si O_m es de tipo $\mathcal{O}_m$, entonces $O_m = VHO_{m-1}$ con O_{m-1} de tipo $\mathcal{O}_{m-1}$. Aplicando otra vez la proposición 5 y el lema 12 varias veces,

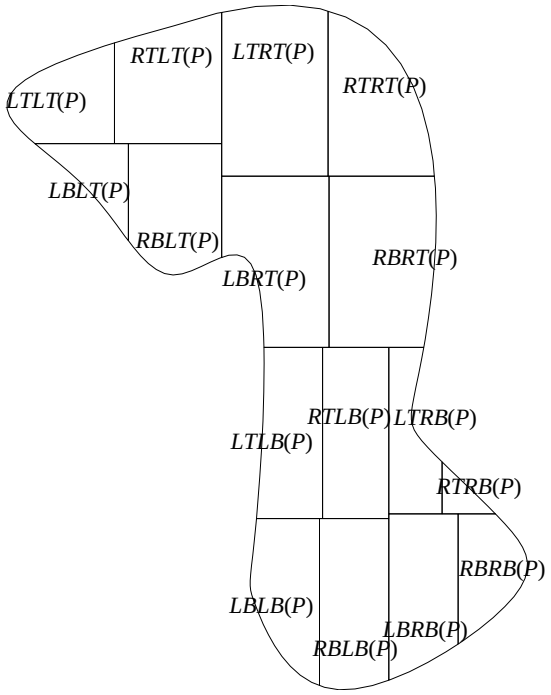


Figura 3.17: Las regiones que surgen de aplicar todos los operadores en $\mathcal{O}_2$ a una región.

se tiene:

$$\begin{aligned}
&\mathrm{dm}_R(O_m(P)) = \\
&\quad = \mathrm{dm}_R(VHO_{m-1}(P)) \leq \sqrt{\mathrm{dm}_H(VHO_{m-1}(P))^2 + \mathrm{dm}_V(VHO_{m-1}(P))^2} \leq \\
&\qquad \leq \sqrt{\frac{\mathrm{dm}_H(HO_{m-1}(P))^2}{2^2} + \mathrm{dm}_V(HO_{m-1}(P))^2} \leq \\
&\leq \sqrt{\frac{\mathrm{dm}_H(O_{m-1}(P))^2}{2^2} + \frac{\mathrm{dm}_V(O_{m-1}(P))^2}{2^2}} \leq \frac{\sqrt{\mathrm{dm}_H(O_{m-1}(P))^2 + \mathrm{dm}_V(O_{m-1}(P))^2}}{2}
\end{aligned}$$

Esto, aplicando la hipótesis de inducción, es menor o igual que:

$$\frac{\sqrt{\left(\frac{\mathrm{dm}_H(P)}{2^{m-1}}\right)^2 + \left(\frac{\mathrm{dm}_V(P)^2}{2^{m-1}}\right)^2}}{2} = \frac{\sqrt{\mathrm{dm}_H(P)^2 + \mathrm{dm}_V(P)^2}}{2^m} = \frac{\mathrm{dm}_R(P)}{2^m}$$

□

En particular, $\mathrm{dm}_R(O_{m+1}(P)) \leq \dfrac{\mathrm{dm}_R(O_m(P))}{2}$. Con respecto al diámetro clásico, en general no es cierto que $\mathrm{dm}(O_m(P)) \leq \dfrac{\mathrm{dm}(P)}{2^m}$. Por ejemplo si P es el círculo de radio 1, $\mathrm{dm}(P) = 2$ pero $RT(P)$ es un sector circular con $\mathrm{dm}(RT(P)) = \sqrt{2} > \dfrac{\mathrm{dm}(P)}{2}$. Sin embargo, tenemos lo siguiente:

Corolario. *Si O_m es un operador de tipo $\mathcal{O}_m$*

$$\mathrm{dm}(O_m(P)) \leq \frac{\mathrm{dm}_R(P)}{2^m}$$

Demostración. Aplicando la proposición 5, $\mathrm{dm}(O_m(P)) \leq \mathrm{dm}_R(O_m(P))$, y encadenando con el lema anterior.

□

Por el lema anterior, tenemos que la división de la región por cortes horizontales y verticales alternativos reduce en efecto el diámetro rectangular de las subregiones obtenidas. Por el corolario, también reduce el diámetro clásico.

Como primer intento de Bisec (correspondiente a la primera fila de la tabla 3.1), los cortes se hacen de modo alternante: cuando se aplica a una región producida por un corte horizontal, Bisec la divide con un corte vertical, y viceversa. En la región inicial aplicamos un corte horizontal. Con esta definición, en la primera llamada a $\mathrm{PRec}(P_I, A)$ (figura 3.6), el retorno de $\mathrm{Bisec}(P_I)$ es $P_0 = T(P_I)$ y $P_1 = B(P_I)$, y en las llamdas subsiguientes ($\mathrm{PRec}(P_0, A)$ y $\mathrm{PRec}(P_1, A)$) $\mathrm{Bisec}(P_0)$ retorna $P_{00} = RT(P_I)$, $P_{01} = LT(P_I)$ y $\mathrm{Bisec}(P_1)$ $P_{10} = RB(P_I)$, $P_{11} = LB(P_I)$.

Por el lema 13 para $m = 1$, las cuatro regiones obtenidas P_{ij}, $0 \leq i, j \leq 1$ satisfacen $\mathrm{dm}_R(P_{ij}) < \dfrac{\mathrm{dm}_R(P_I)}{2}$. Además, el lema 13 para $m > 1$ también describe el diámetro de las regiones en llamadas posteriores, que es $\mathrm{dm}_R(P_{i_1 j_1 i_2 j_2 \cdots i_m j_m}) < \dfrac{\mathrm{dm}_R(P_I)}{2^m}$ después de $2m$ llamadas recursivas (los índices $0 \leq i_k, j_k \leq 1$ codifican las subregiones). Así, $\mathrm{Bisec}(P)$ (en este primer intento) cumple los requisitos *a)* y *b)* de la página 102.

Sin embargo esto es solo un primer intento porque no se verifica el requisito *c*): las subregiones obtenidas pueden retornar con error en PCR. Lo siguiente es un ejemplo de esto en una situación genérica. Consideremos la región P cuyo borde es una circunferencia Γ centrada ligeramente a la derecha del origen, que contiene las raíces del polinomio $f(z) = z^3 - 1$. Su imagen $\Delta = f(\Gamma)$ rodea tres veces el

origen (ver la figura 3.18). El origen está marcado con ∘, y es la imagen de las tres raíces. Siguiendo el procedimiento PRec (figura 3.6), Bisec(P) descompone P en dos sectores circulares, mediante un corte horizontal. Una de estas subregiones hijas tiene un borde, digamos Γ_c de intervalo paramétrico $[a_c, b_c]$, que cruza una raíz, como se muestra en la figura 3.18 *c*). La imagen de Γ_s cruza el origen, véase $f(\Gamma_s)$ en la figura 3.18 *d*), y por tanto PCR aplicado a Γ_s (es decir PCR(Γ_s, θ), para cualquier valor del parámetro θ) retorna con error porque no puede calcular su índice. Una situación similar puede surgir con cortes verticales.

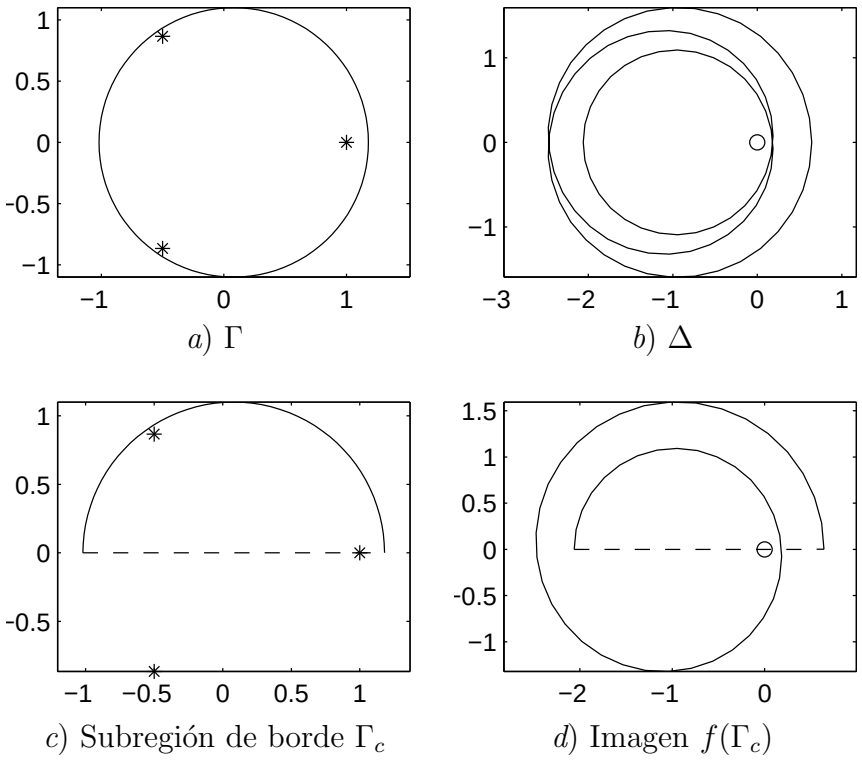


Figura 3.18: Región que contiene las raíces de $z^3 - 1$. El borde de la subregión Γ_c cruza una raíz.

Aunque la configuración de la figura 3.18 *c*), con una raíz exactamente en el borde Γ_c, puede ser altamente improbable, quizás hay una raíz lo bastante cerca para causar un error. Por tanto es necesario evitar estos bordes hijo fallidos (es decir, pasando por una raíz) para que PRec pueda progresar. También es necesario evitar bordes hijo que pasen cerca de una raíz porque el coste computacional de

PCR es inversamente proporcional a a distancia a una raíz.

3.2.3. Segundo intento: cortes iterados con desplazamiento

Definimos un nuevo método de partición que construye subregiones en las que PCR retorna con éxito y con bajo coste (correspondiente a la segunda fila de la tabla 3.1). Dividiremos la región padre P por un corte recto. Sin embargo el corte no será, en general, a lo largo de $m_H(P)$, sino a lo largo de una paralela a esta línea. Lo mismo se hará con $m_V(P)$.

El problema de hallar un método de corte efectivo y no costoso se discute a la luz del teorema 6. Hemos encontrado una solución iterativa, con desplazamientos progresivos a partir de intentos fallidos. La figura 3.19 muesta varios desplazamientos producidos por el procedimiento. Llamamos a este método Cortes Iterados con Desplazamiento (CID).

Por el teorema 6*c*), si $\mathrm{PCR}(\Gamma, \theta)$ retorna con error, entonces hay una raíz z_1 de f a $(4+\sqrt{2})n\theta$ o menos de Γ, es decir $\mathrm{d}(z_1, \Gamma) \leq (4+\sqrt{2})n\theta$. Decimos que una raíz z_1 está *cerca de* Γ si $\mathrm{d}(z_1, \Gamma) \leq (4+\sqrt{2})n\theta$ (la cercanía depende del valor de θ). Así, en caso de error, hay una raíz (o varias) cerca de Γ. También decimos que esa raíz (o raíces) ha *causado* el error. Quizás esto es un abuso de lenguaje, porque el valor de la raíz (o raíces) permanece desconocido, pero facilita la exposición.

Definimos primero CID. Toma como parámetros la región P y un índice de iteración. La primera iteración de CID realiza un corte Υ a lo largo de $m_H(P)$, como se describe en la subsección 3.2.1, produciendo las curvas $\Gamma_T = \Upsilon; \Gamma_{[s,t]}$ y $\Gamma_B = \Upsilon^-; \Gamma_{[t,b]}; \Gamma_{[a,s]}$. Ahora aplicamos PCR a las curvas hijas: $\mathrm{PCR}(\Gamma_T, \theta)$ y $\mathrm{PCR}(\Gamma_B, \theta)$, donde $\theta = Z_{\Gamma_H} = \dfrac{\mathrm{mín}(\mathrm{dm}_H(H(P)), \mathrm{dm}_V(H(P)))}{4(4+2\sqrt{2})(n+2)n}$, siendo H o bien T o bien B, como corresponda. Si cualquiera de las dos llamadas retorna con error, el bucle descarta este corte y usa otro dado a lo largo de una linea situada más arriba. Quizás esta segunda línea produce otra vez un error. En tal caso la siguiente iteración del bucle realiza una división siguiendo una tercera línea horizontal, esta vez por debajo de $m_H(P)$.

Hemos descrito el caso de un CID horizontal, siendo similar el vertical. Para ser preciso, CID comienza en el paso 0 con el corte central (horizontal o vertical), que se considera desplazado una distancia de $h_0 = 0$. Después sigue, en el paso i, un corte desplazado h_i de la línea central si el paso $i-1$ ha producido una

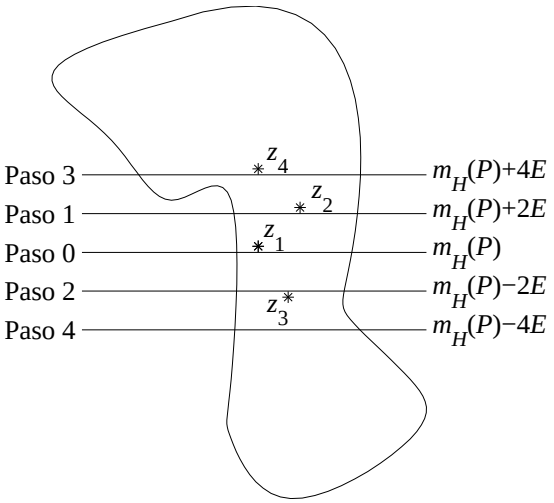


Figura 3.19: Cortes Iterados con Desplazamiento (CID): siendo $E = (4+2\sqrt{2})n\theta$, las líneas $m_H(P)$, $m_H(P)+2E$, $m_H(P)-2E$, $m_H(P)+4E$ causan error sucesivamente por proximidad con las raíces z_1, z_2, z_3 y z_4 respectivamente. Finalmente $m_H(P)-4E$ no tiene ninguna raíz a distancia menor que E.

subregión que causa un error. Si i es impar entonces $h_i = (i+1)E$, si no $h_i = -iE$ (siendo being $E = (4+2\sqrt{2})n\theta$). Denotamos con CID(P,i) las dos subregiones producidas a partir de P con un corte Υ desplazado una distancia h_i del corte central, especificando corte horizontal o vertical si es necesario. Llamamos *bucle de desplazamiento* a la secuencia de estas llamadas CID(P,i) con valores crecientes de i.

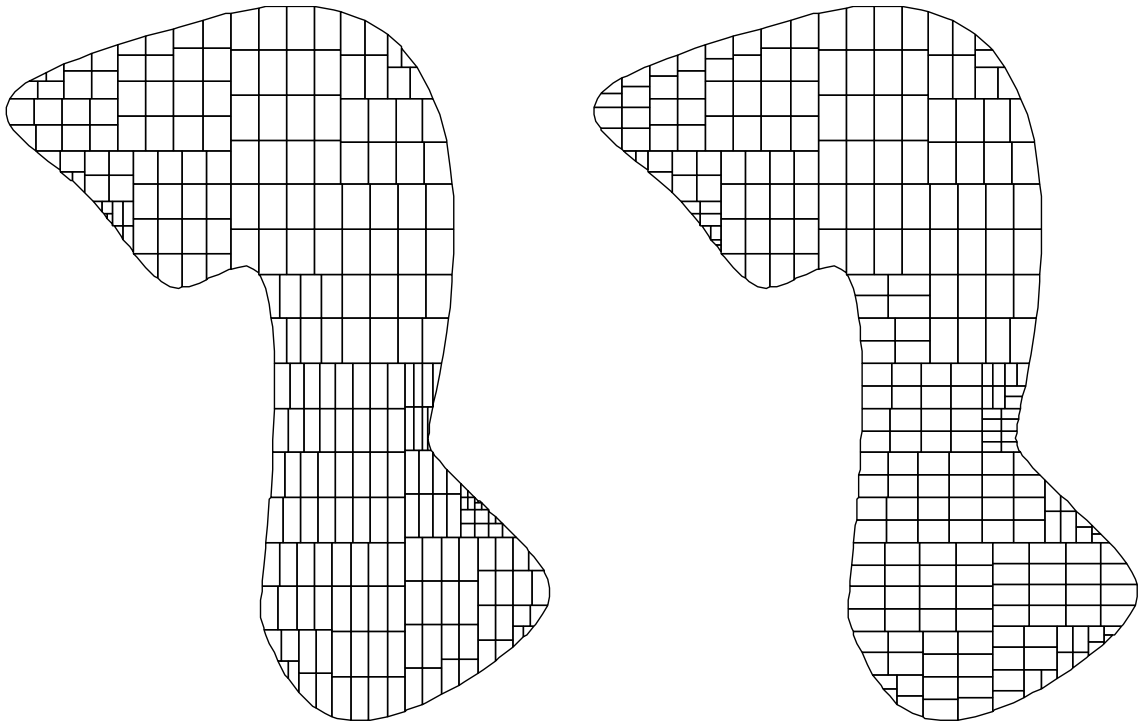
Con respecto al valor θ con el que CID llama a PCR, nos basamos en la proposición 6 más adelante. Si θ toma el valor $Z_\Gamma = \dfrac{\text{mín}(\text{dm}_H(P), \text{dm}_V(P))}{4(4+2\sqrt{2})(n+2)n}$, entonces E (y por lo tanto h_i) es lo bastante pequeño como para que el último corte hecho por CID divida en efecto a P. Como este es el último corte hecho por CID, PCR retorna con éxito en las subregiones que produce.

El segundo intento de la tabla 3.1 usa CID para satisfacer el requisito *c*). Como el tercer intento también usa CID, satisface asimismo el requisito *c*). El segundo intento, sin embargo, no garantiza el decrecimiento del diámetro. Los razonamientos expuestos en la discusión del primer intento para probar que el diámetro decrece (lema 13) ya no son válidos, porque CID no corta las regiones por el punto medio. Esto motiva el tercer intento de división.

3.2.4. Tercer y definitivo intento: cortes a lo largo del eje menor

Decimos que P *tiene eje menor horizontal* si $\mathrm{dm}_V(P) \geq \mathrm{dm}_H(P)$, y que *tiene eje menor vertical* si $\mathrm{dm}_V(P) < \mathrm{dm}_H(P)$.

La tercera versión de Bisec, para dividir una región P en PRec sin error, es la siguiente: para una región dada P, si $\mathrm{dm}_V(P) \geq \mathrm{dm}_H(P)$, entonces Bisec($P$) es el resultado de CID(P, i) con corte horizontal, y si $\mathrm{dm}_V(P) < \mathrm{dm}_H(P)$, entonces Bisec($P$) es el resultado de CID(P, i) con corte vertical. Estos son los cortes a lo largo del eje menor de P. Definimos Bisec de esta manera, en vez de alternar cortes horizontales y verticales, porque así obtenemos un decrecimiento en diámetro. Lo demostraremos en la proposición 8*b)*. La figura 3.20 compara las subregiones que surgen por los dos métodos.



a) *b)*

Figura 3.20: En *a*), después de cortes alternos horizontales y verticales, algunas regiones resultan muy elongadas. En *b*), con cortes a lo largo del eje menor, las regiones se diferencian solo ligeramente de la forma cuadrada.

Hay otro fenómeno que puede ocurrir si alternamos cortes horizontales y verticales como en el segundo intento: si por casualidad los cortes horizontales se desplazan hacia arriba sistemáticamente, y los verticales hacia la derecha, por ejemplo, alguna de las regiones que surgen tras varios de estos cortes son muy elongadas. La elongación se mide por la relación de aspecto, que definiremos con precisión antes del lema 19. Es necesario mantener la relación de aspecto acotada porque veremos (teorema 7) que la máxima profundidad de recursión que alcanza PRec (y por tanto su coste) en una región depende de su relación de aspecto. El tercer intento de la tabla 3.1 no solo decrece el diámetro, sino que también mantiene la elongación de las subregiones por debajo de cierto umbral.

3.2.5. Definición final de PRec

Definimos ahora PRec con el método de partición incorporando cortes desplazados, y a lo largo del eje menor, es decir, el tercer intento de Bisec. El procedimiento PRec se muestra de nuevo en el pseudocódigo de la figura 3.21. Es una ampliación de la figura 3.6 con una llamada al procedimiento PCR para calcular el número de raíces, y con el bucle de desplazamiento de llamadas a CID en caso de errores en PCR. Para comentarlo adecuadamente, citaremos las llamadas anidadas de PRec diciendo que la primera llamada tiene *profundidad de recursión* 0, y que las llamadas a PRec hechas dentro de una llamada de profundidad de recursión v tienen *profundidad de recursión* $v+1$. Los siguientes cuatro puntos definen la estrategia de PRec para la gestión de errores: primero, cuando $\mathrm{PRec}(P, A)$ a profundidad v recibe un error de $\mathrm{PCR}(\Gamma, Z_\Gamma)$ (línea 2), identifica Γ como singular y retorna con error a la instancia llamadora, a profundidad $v-1$. PRec retorna a la línea 6, y la siguiente línea gestiona el error que pudiera haber ocurrido. Siendo γ el punto retornado en error por $\mathrm{PRec}(P_0, A)$ (o $\mathrm{PRec}(P_1, A)$), este error es un *error de corte* si $\mathrm{d}(\gamma, \Upsilon) \leq (4+\sqrt{2})n\theta$, y es un *error diferido* en caso contrario. Segundo: si el error es de tipo corte (línea 8), esta instancia de PRec a profundidad $v-1$ considera P_0 como singular (o P_1, según la subregión que cause error), y genera un nuevo corte en P. Tercero: si el error es de tipo diferido, la instancia PRec considera su región P como singular y eleva el error a su llamadora a profundidad $v-2$, que a su vez lo clasifica como de corte o diferido, y procede correspondientemente. Cuarto: la llamada inicial $\mathrm{PRec}(P_I, A)$ clasifica todos los errores como de corte, suponiendo

Procedimiento Recursivo de División, con gestión de errores: Para hallar las raíces de un polinomio f dentro de una región convexa P_I.

Parámetros de entrada: Una región convexa P_I del plano complejo, un polinomio f de grado n, y un parámetro de precisión $A > 0$.

Salida: Las secuencias $\Pi = (P_1, P_2, \ldots, P_k)$ y $N = (n_1, n_2, \ldots, n_k)$ cuando se retorna normalmente. Un punto γ del borde Γ de P que está a $(4+\sqrt{2})nZ_\Gamma$ o o menos de una raíz, cuando se retorna con error, donde $Z_\Gamma = \dfrac{\text{mín}(\text{dm}_H(P), \text{dm}_V(P))}{4(4+2\sqrt{2})(n+2)n}$.

Método: PRec(P, A) {

1: Calcular PCR(Γ, Z_Γ).

2: Si PCR retorna con error el valor de parámetro t, entonces
retornar $\gamma = \Gamma(t)$. [Retorno con error]

3: Si el número de cruces de PCR(Γ, Z_Γ) es 0, entonces
retornar Π y N vacías. [Salida normal 1]

4: Si $\text{dm}_R(P) \leq A$ entonces
retornar $\Pi = (P)$ y $N =$ (cruces de PCR(Γ, Z_Γ)). [Salida normal 2]
$i = 0$. [Contador de desplazamientos]
Repetir { [Bucle de division]

5: $(P_0, P_1) =$CID(P, i)
lo que produce el segmento de corte Υ.

6: $(\Pi_0, N_0) =$ PRec(P_0, A); $(\Pi_1, N_1) =$ PRec(P_1, A).

7: Si cualquiera de las dos llamadas anteriores a PRec retorna con error el punto γ, entonces

8: Si $\text{d}(\gamma, \Upsilon) \leq (4+\sqrt{2})nZ_\Gamma$ entonces
incrementar i. [error de corte]
si no retornar γ. [error diferido]

9: } Hasta que ambas llamadas a PRec retornen normalmente.

10: Retornar la concatenación de (Π_0, N_0) y (Π_1, N_1). [Salida normal 3]
}

Figura 3.21: Procedimiento Recursivo de División con gestión de errores.

que no hay raíces cerca de Γ_I.

Si aplicamos PCR(Γ_I, Z_{Γ_I}) a la región inicial y obtenemos un retorno normal, podemos asegurar que PRec retornará sin error. En caso contrario, PRec retornará con error precisamente un punto del borde cerca de una raíz, como dice la afirmación sobre la salida en la figura 3.21. Hemos introducido el cuarto punto anterior para evitar esta situación. Así PRec es un método para hallar con seguridad las raíces en P_I, con la precondición de que su borde Γ_I esté bien separado de las raíces.

La figura 3.22 muestra una traza del método. Empezando con PRec(Γ_a, A), la región rectangular de borde Γ_a es dividida en el paso 1 por el corte Υ en las subregiones P_0 y P_1, de bordes Γ_0 y Γ_1. En los pasos 2 y 3 (las llamadas PRec(Γ_0, A) y PRec(Γ_1, A), respectivamente), PCR calcula sin error el número de raíces dentro de ellas. Γ_0 no tiene ninguna raíz y Γ_1 tiene una, r_1. Aunque r_1 está cerca del borde de estas regiones, en este ejemplo no causa error PCR. Notemos que el teorema 6c) afirma que si hay un error, entonces hay una raíz cercana, pero el recíproco no es cierto en general.

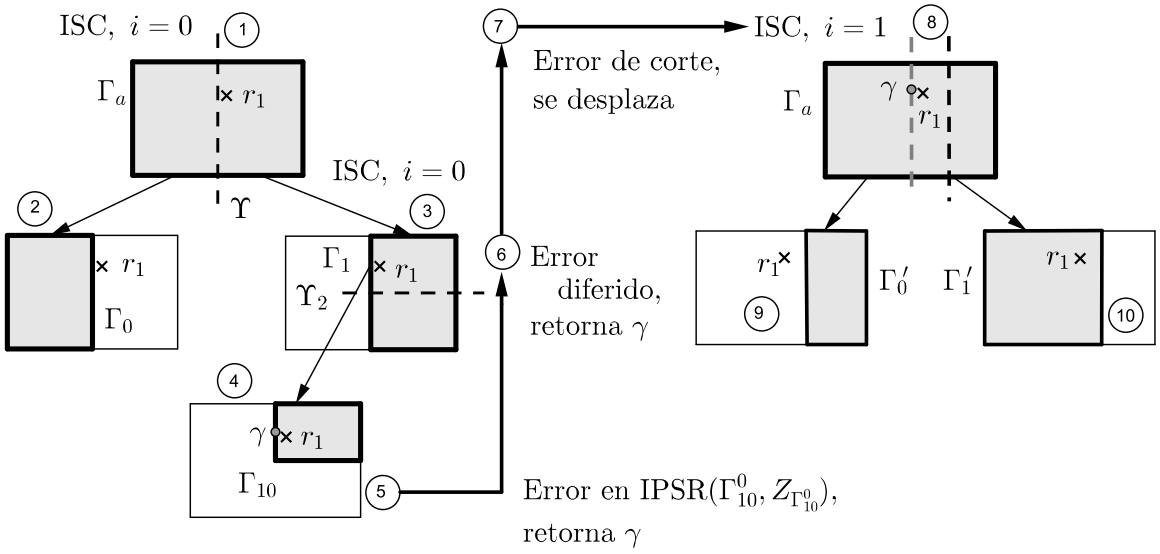


Figura 3.22: El etiquetado numérico paso 1, paso 2, ..., paso 10 muestra la sucesión temporal de acciones descrita en el texto.

En el paso 3, Γ_1 se divide por Υ_2 en Γ_{10} y Γ_{11} (esta última subregión no se muestra en la figura 3.22). Dentro de PRec(Γ_{10}, A) (paso 4), PCR($\Gamma_{10}, Z_{\Gamma_{10}}$) retorna con error el parámetro u del punto $\gamma = \Gamma_{10}(u)$. PRec(Γ_{10}, A) retorna (paso

5) con error este punto a su llamadora (que es PRec(Γ_1, A)), que lo clasifica como diferido (paso 6) porque γ no pertenece a Υ_2. El error es por tanto elevado a PRec(Γ_a, A), que lo clasifica como error de corte (paso 7) porque γ pertenece a Υ. Como resultado, PRec (Γ_a, A) desplaza el corte Υ (paso 8), creando las nuevas subregiones de bordes Γ'_0 y Γ'_1, que son posteriormente procesadas por PRec (pasos 9 y 10).

PRec está ya definitivamente expuesto. Falta probar que CID no llega a hacer un desplazamiento tan grande que produzca una subregión vacía, y que el corte a lo largo del eje menor en efecto d el diámetro de las subregiones. Esto se hará en la siguiente sección, por que es una material extenso.

3.3. Propiedades del procedimiento recursivo

Vamos a demostrar, en la proposición 8, subsección 3.3.3, que el tercer intento de la tabla 3.1, que usa CID a lo largo del eje menor, cumple los requisitos *b)* y *c)*. Previamente daremos las proposiciones 6 en la subsección 3.3.1 y 7 en 3.3.2. La proposición 6 afirma que, para algún valor de i, en la i-ésima iteración del bucle de desplazamiento (lineas 5-9 de la figura 3.21) para particionar una región P, las subregiones P_0, P_1 producidas por CID(P, i) tienen bordes $\Gamma_{P_0}, \Gamma_{P_1}$ tales que PCR(Γ_{P_0}, θ) y PCR(Γ_{P_1}, θ) retornan exitosamente (si θ está por debajo de cierta cota). Esto implica la salida del bucle de desplazamiento, y por tanto que los desplazamientos de los cortes dados por CID llegan como mucho a una distancia $K\theta$ de la línea central de P, para un factor K que depende del polinomio. En particular si θ está por debajo de la cota Θ_H(o Θ_V para cortes verticales), los desplazamientos de corte no alcanzan las líneas de soporte. De este modo obtenemos siempre regiones no vacías.

La proposición 7 afirma que tras m cortes en ciertas hipótesis, el diámetro de la región resultante es menor que la precisión A. Necesitamos la larga subsección 3.3.2 para explicitar esas hipótesis. La proposición 8 dice que los cortes hechos por CID a lo largo del eje menor satisfacen esas hipótesis. De este modo obtenemos diámetros decrecientes.

3.3.1. Efectividad de CID

En el bucle de desplazamiento de la figura 3.19, cada raíz z_i es la causa de una nueva iteración. Esto nos permite acotar el número de iteraciones usando el número de raíces dentro de P, y también acotar la máxima distancia alcanzable por los desplazamientos. Siendo n_P el número de raíces dentro de P, y n el grado del polinomio, definimos $\Theta_H = \dfrac{\mathrm{dm}_V(P)}{(8+4\sqrt{2})(n_P+1)n}$ y $\Theta_V = \dfrac{\mathrm{dm}_H(P)}{(8+4\sqrt{2})(n_P+1)n}$. Los valores h_i son las distancias previamente definidas en el segundo intento ($h_i = (i+1)E$ si i is impar, si no $h_i = -iE$, con $E = (4+2\sqrt{2})n\theta$).

Proposición 6. *Sea P una región que contiene n_P raíces, y $\theta < \Theta_H$. Si no hay ninguna raíz a $(4+2\sqrt{2})n\theta$ o menos del borde de P, entonces hay una iteración i del bucle de desplazamiento tal que la llamada* $\mathrm{CID}(P,i)$ *(si es horizontal) retorna dos regiones no vacías Γ_T, Γ_B, verificando que* $\mathrm{PCR}(\Gamma_T,\theta)$ *y* $\mathrm{PCR}(\Gamma_B,\theta)$ *retornan exitosamente y*

$$|h_i| < (4+2\sqrt{2})(n_P+1)n\theta.$$

Lo mismo es válido para $\mathrm{PCR}(P,i)$ *con cortes verticales (y subregiones Γ_L,Γ_R), si $\theta < \Theta_V$.*

Demostración. Consideremos la primera iteración $\mathrm{CID}(P,0)$, el paso 0 de un corte horizontal (figure 3.19), es decir, las dos subregiones $T(P)$ y $B(P)$ de bordes respectivos Γ_T y Γ_B, que surgen de P de borde Γ, por un corte horizontal a lo largo de $m_H(P)$, y siendo $\Upsilon : [0,d] \to \mathbb{C}$ el corte. Si $\mathrm{PCR}(\Gamma_T,\theta)$ retorna con error, como $\Gamma_T = \Upsilon;\Gamma_{[s,t]}$ y por hipótesis no hay raíces cerca de Γ (ni por tanto cerca de su segmento $\Gamma_{[s,t]}$) entonces por el teorema 6*c* hay un punto en Υ a $E = (4+2\sqrt{2})n\theta$ o menos de una raíz.

En ese caso, con el segmento Υ de $m_H(P)$ cerca de una raíz, digamos z_1, en la siguiente iteración el procedimiento $\mathrm{CID}(P,1)$ hace un corte a $2E$ por encima de $m_H(P)$. El nuevo corte está a distancia mayor que E de z_1, luego z_1 no puede provocar error. Quizás hay otra raíz z_2 cerca de esta primera línea desplazada causando un nuevo error en PCR. En tal caso CID (en el paso 2) hace un corte horizontal por una segunda línea desplazada, por debajo de $m_H(P)$, a una distancia $-2E$. Es decir, el paso 1 en CID se dispara por z_1, el paso 2 por z_2, y así sucesivamente. Por tanto CID hará como mucho hasta el paso n_P (siendo n_P el número de raíces dentro de P). Esto es porque las líneas insertadas en los cortes $0,1,2,...n_P-1$ todas

tienen al menos una raíz cercana y por tanto no queda ninguna que cause error en el corte del paso n_P. como mucho un número $\left\lceil \frac{n_P}{2} \right\rceil$ de las n_P líneas tentativas están por encima de $m_H(P)$ y el resto $\left\lfloor \frac{n_P}{2} \right\rfloor$ están por debajo. Como los cortes están separados $2E$, la última línea de corte horizontal está a una distancia h_i de $m_H(P)$ menor o igual, en valor absoluto que $\left\lceil \frac{n_P}{2} \right\rceil 2E$, ya sea por arriba o por debajo (porque $\left\lfloor \frac{n_P}{2} \right\rfloor < \left\lceil \frac{n_P}{2} \right\rceil$). Esto es igual a $(n_P+1)E$ si n_P es impar ($n_P E$ si n_P es par). En ambos casos, se cumple que $|h_i| \leq (n_P+1)E = (4+2\sqrt{2})(n_P+1)n\theta$, como queríamos demostrar.

Falta probar que la última línea desplazada en efecto corta P. Lo haremos para todas cada línea desplazada, incluyendo la última. Estas líneas están a una distancia h_i de la línea central, luego cortar en efecto a P significa $|h_i| < \frac{\mathrm{dm}_V(P)}{2}$ en CID horizontal ($|h_i| < \frac{\mathrm{dm}_H(P)}{2}$ en CID vertical). Por hipótesis, tenemos que $\theta < \Theta_H = \frac{\mathrm{dm}_V(P)}{2(4+2\sqrt{2})(n_P+1)n}$, luego

$$|h_i| \leq (4+2\sqrt{2})(n_P+1)n\theta < \frac{(4+2\sqrt{2})(n_P+1)n\,\mathrm{dm}_V(P)}{2(4+2\sqrt{2})(n_P+1)n} = \frac{\mathrm{dm}_V(P)}{2}.$$

Un razonamiento similar es válido para subdivisiones verticales, luego $h < \frac{\mathrm{dm}_H(P)}{2}$ también, y concluimos.

□

3.3.2. Diámetros decrecientes

Recordemos que el lema 13 nos da una reducción en diámetro para divisiones hechas por la línea media y alternando horizontal y vertical. Como el tercer intento de Bisec no opera de esta manera. necesitamos desarrollar otro resultado general acerca del decrecimiento de los diámetros con cortes desplazados, y a lo largo del eje menor (más adelante en la proposición 7).

Como en el lema 13, consideramos operadores de corte (o simplemente cortes) T, B, L y R, genéricamente denotados C. La subregión que surge tras aplicar varios operadores de corte a una región P, es el resultado de una cadena $C_m \cdots C_2 C_1(P)$ de cortes. Probamos en ese lema anterior que tras una cadena de $2m$ cortes alter-

nando horizontal y vertical (llamada operador de tipo $\mathcal{O}_m$), la región que surge tiene diámetro menor que $\frac{\mathrm{dm}_R(P)}{2^m}$. Sin embargo, ahora los cortes no se alternan, sino que se hacen a lo largo del eje menor, y la relación entre el número de cortes y el diámetro de la región obtenida no es tan simple. Para ilustrar el problema, consideremos una región inicial con una forma muy elongada a lo ancho (figura 3.23). La cadena de cortes a lo largo del eje menor empieza con una primera subsecuencia de cortes verticales. El tercer corte produce una subregión aproximadamente tan alta como ancha. Después de estos cortes verticales, viene una segunda subsecuencia de cortes a lo largo del eje menor, cuyo tipo resulta ser alternante entre horizontal y vertical.

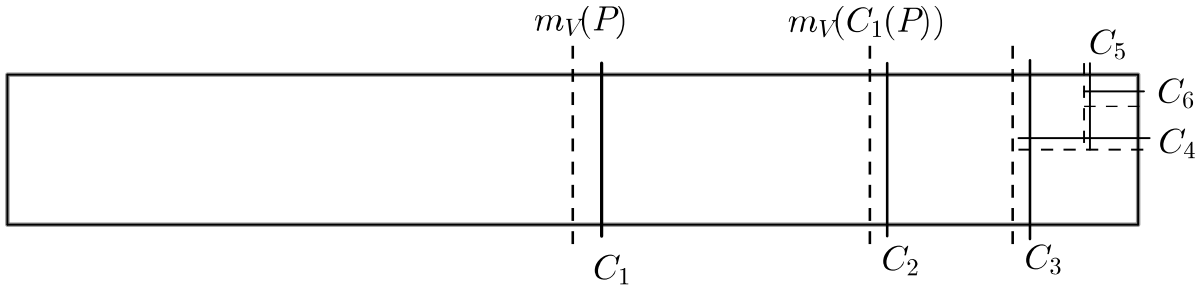


Figura 3.23: Seis cortes a lo largo del eje menor hechos a una región elongada P. Empiezan con C_1, C_2 y C_3, cada uno con cierto desplazamiento desde la línea central, discontinua. Tras estos, C_4, C_5 y C_6 alternan entre tipo horizontal y vertical.

En general, una cadena de cortes a lo largo del eje menor no tiene una primera subsecuencia de cortes del mismo tipo y, después,una segunda y última subsecuencia de cortes de tipo alternante. Además, en virtud del desplazamiento, la subregión que surge puede tener alta elongación, como en la figura 3.23 después de C_6, requiriendo a su vez una subsecuencia de cortes a lo largo del eje menor todos del mismo tipo, vertical en este caso.

Cualquier corte dado a una región P produce subregiones de menor diámetro rectangular. ¿Cuántos cortes m se necesitan para reducir el diámetro rectangular de la región resultante por debajo de la precisión A? El interés por m viene del hecho de que, en el procedimiento PRec, las regiones producidas por una cadena de m cortes son precisamente aquellas que aparecen a profundidad de recursión m. Por tanto, el número de cortes necesario es igual a la máxima profundidad de

recursión alcanzada por PRec. Sabiendo este número m demostraremos la finitud de PRec en el teorema 7.

Anteriormente en el lema 13 relacionamos el número de cortes (alternando horizontal y vertical) y el decrecimiento en diámetro rectangular: $2m$ cortes producen una subregión con un diámetro rectangular menor que $\frac{1}{2^m}$ por el diámetro de P. Ahora tenemos un resultado diferente. Dada una cadena de m cortes a lo largo del eje menor, $C_m \cdots C_2 C_1(P)$, supongamos que tiene un número t_H de operadores de tipo horizontal y un número t_V de operadores de tipo vertical, de modo que $m = t_H + t_V$. Definimos $t_0 = \text{mín}(t_H, t_V)$, es decir, el mínimo de la cantidad de operadores de cada tipo . Mostraremos que el decrecimiento en diámetro rectangular después de aplicar la cadena $C_m \cdots C_2 C_1(P)$ está relacionado con el mínimo t_0 de esta forma: $\text{dm}_R(C_m C_{m-1} \cdots C_1(P)) \leq K^{t_0} \text{dm}_R(P)$, con un factor K menor que 1 (lema 17). Probaremos este lema en la subsección 3.3.2.1. A partir de él deduciremos que, dada A, tras una cadena con $t_0 \geq \frac{\lg_2 \frac{\text{dm}_R(P)}{A}}{\lg_2(1/K)}$ (equivalentemente $K^{t_0} \text{dm}_R(P) \leq A$), el diámetro rectangular de la región resultante es menor o igual que A. Esto nos da el mínimo número t_0 de operadores de cada tipo necesarios para reducir el diámetro rectangular de la región resultante por debajo de A, pero como se ha comentado queremos saber el número de cortes m necesarios para este fin, que es igual a la profundidad de recursión.

¿Cómo tiene que ser de larga una cadena $C_m C_{m-1} \cdots C_1(P)$ de m cortes para tener al menos t_0 operadores de cada tipo? La respuesta nos da la profundidad de recursión m que PRec debe alcanzar para tener una reducción en diámetro rectangular al menos por un factor de K^{t_0}. Mostraremos en la subsección 3.3.2.2 que una cadena de m cortes tiene al menos $t_0 \geq \frac{m - L_0}{L_1}$ operadores de cada tipo, para ciertos L_0, L_1 dependiendo de P. Por tanto una cadena de m cortes reduce el diámetro rectangular al menos por un factor de $K^{\frac{m-L_0}{L_1}}$. Este es el lema 21, que expresa la reducción en diámetro en función de m, como el lema 17 la expresaba en función de t_0. Finalmente, en la proposición 7, formulamos la conclusión de esta sección sobre diámetros decrecientes: dada A, tras una cadena con $m \geq F \cdot L_1 + L_0$ cortes para cierto F, el diámetro rectangular de la región resultante es menor o igual que A.

3.3.2.1. Cadenas de cortes desplazados

Se define el operador T_λ de la siguiente manera (ver la figura 3.24). La línea horizontal a una distancia λ por encima de $m_H(P)$ se denota $m_{H+\lambda}(P)$. La región $T_\lambda(P)$ es la intersección de P con el semiplano superior definido por $m_{H+\lambda}(P)$. Si λ es un valor negativo, la línea $m_{H+\lambda}(P)$ debe entenderse por debajo de $m_H(P)$. Análogamente el operador $B_\lambda(P)$ es la intersección de P con el semiplano inferior definido por la misma línea horizontal, $m_{H+\lambda}(P)$. De esta manera $T_\lambda(P) \cup B_\lambda(P) = P$, y $T_\lambda(P) \cap B_\lambda(P)$ es su borde común, un segmento de $m_{H+\lambda}(P)$. De modo similar $R_\lambda(P)$ es la intersección de P con el semiplano derecho definido por $m_{V+\lambda}(P)$ (la línea vertical a una distancia λ a la derecha de $m_V(P)$), y $L_\lambda(P)$ es la intersección con el semiplano izquierdo definido por la misma línea $m_{V+\lambda}(P)$.

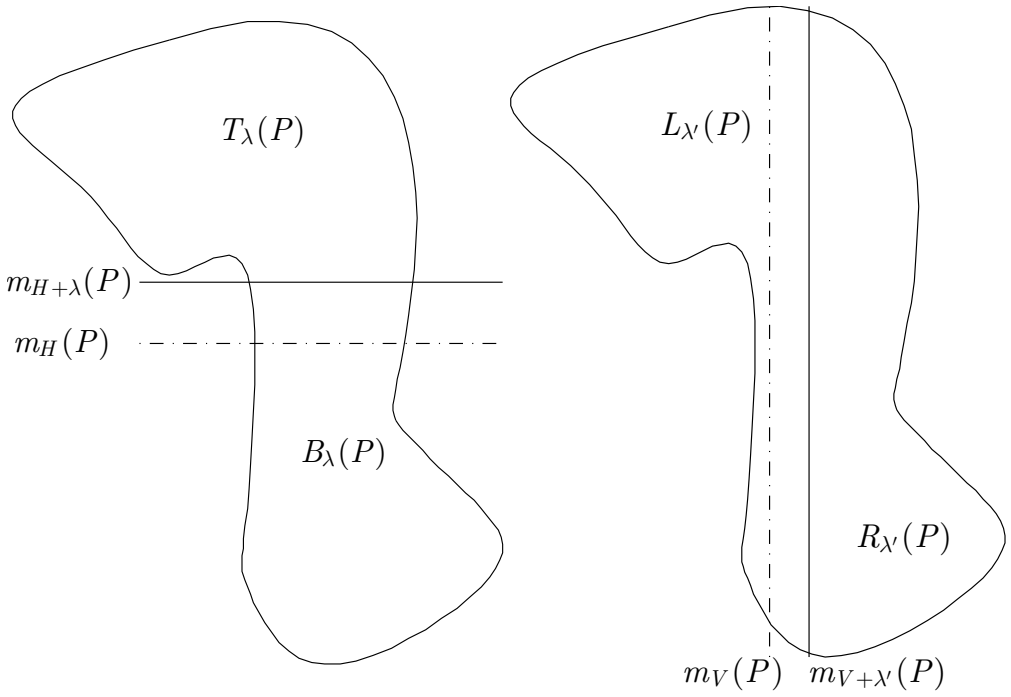


Figura 3.24: La acción de los operadores T_λ, B_λ, $L_{\lambda'}$ y $R_{\lambda'}$.

Lema 14. *Si $\lambda \geq 0$, entonces $B(P) \subset B_\lambda(P)$ y $R(P) \subset R_\lambda(P)$. Si $\lambda \leq 0$, entonces $B(P) \supset B_\lambda(P)$ y $R(P) \supset R_\lambda(P)$. Para los operadores T_λ y L_λ, en la mismas hipótesis, las inclusiones se invierten.*

Demostración. Es inmediato teniendo en cuenta la posición de $m_{H+\lambda}(P)$ y $m_H(P)$ (o de $m_{V+\lambda}(P)$ y m_V(P)).

□

Decimos que los operadores T_λ y B_λ son *de tipo horizontal* y L_λ y R_λ *de tipo vertical*. Denotamos con H_λ a un operador horizontal genérico, es decir, o T_λ o B_λ. Asimismo V_λ denota o L_λ o R_λ. Notemos que para λ (o $-\lambda$) suficientemente grande en valor absoluto, el resultado de un operador puede ser el conjunto vacío, cuyo diámetro es convencionalmente cero.

Lema 15. *Para un operador horizontal* H_λ, $\mathrm{dm}_H(H_\lambda(P)) \leq \mathrm{dm}_H(P)$ *y* $\dfrac{\mathrm{dm}_V(P)}{2} - |\lambda| \leq \mathrm{dm}_V(H_\lambda(P)) \leq \dfrac{\mathrm{dm}_V(P)}{2} + |\lambda|$.

Para un operador vertical V_λ, $\mathrm{dm}_V(V_\lambda(P)) \leq \mathrm{dm}_V(P)$ *y* $\dfrac{\mathrm{dm}_H(P)}{2} - |\lambda| \leq \mathrm{dm}_H(V_\lambda(P)) \leq \dfrac{\mathrm{dm}_H(P)}{2} + |\lambda|$.

Demostración. Discutimos los operadores horizontales, siendo similares los verticales. La primera desigualdad es consecuencia del lema 11 y el lema 14. Para las dos desigualdades encadenadas, notemos que si $|\lambda| > \dfrac{\mathrm{dm}_V(P)}{2}$, entonces $H_\lambda(P)$ es el conjunto vacío y las desigualdades son trivialmente ciertas. En el caso contrario, $|\lambda| \leq \dfrac{\mathrm{dm}_V(P)}{2}$, la línea que define a $H_\lambda(P)$, $m_{H+\lambda}(P)$, estará situada por encima de $m_H(P)$, si o bien $\lambda \geq 0$ y $H_\lambda = T_\lambda$, o bien $\lambda \leq 0$ y $H_\lambda = B_\lambda$, y por debajo en otros casos. En cualquier caso está a una distancia $|\lambda|$ de $m_H(P)$. Esta línea horizontal $m_H(P)$ está a $\dfrac{\mathrm{dm}_V(P)}{2}$ de l_T si H_λ es T_λ (alternativamente l_B si H_λ es T_λ). Como $m_{H+\lambda}(P)$ y l_T (o l_B) son las líneas de soporte de $\mathrm{dm}_V(H_\lambda(P))$, entonces se verifican las desigualdades.

□

Analizaremos el impacto del desplazamiento en la reducción de diámetro introduciendo la noción de tolerancia. Para un operador horizontal aplicado a una región P, es decir $H_\lambda(P)$, definimos su *tolerancia* como $\dfrac{2|\lambda|}{\mathrm{dm}_V(P)}$. Depende de λ y P, de modo que operadores con tolerancia cero hacen el corte por la línea media $m_H(P)$, con tolerancia $1/2$ cortan a medio camino entre la línea media y una de las dos líneas de soporte horizontal, y con tolerancia 1 o mayor el corte es degenerado

(es decir, una de las subregiones que surgen tiene área nula o es vacía). Del mismo modo, decimos que un operador vertical $V_\lambda(P)$ tiene una *tolerancia* de $\dfrac{2|\lambda|}{\mathrm{dm}_H(P)}$. En la figura 3.24, ambos cortes tienen tolerancia 1/5, aunque $\lambda > \lambda'$. Hemos escogido este término del campo de la ingeniería mecánica, donde se refiere al rango permisible de variación en una medida de un objeto sin que afecte a su función. Consideramos que un corte por la línea media está en el punto de referencia, y que un corte con algún desplazamiento se aleja de esta referencia.

Lema 16. *Si $H_\lambda(P)$ tiene una tolerancia de μ, entonces*

$$\frac{1-\mu}{2}\,\mathrm{dm}_V(P) \le \mathrm{dm}_V(H_\lambda(P)) \le \frac{1+\mu}{2}\,\mathrm{dm}_V(P).$$

Similarmente, si $V_\lambda(P)$ tiene una tolerancia de μ, entonces

$$\frac{1-\mu}{2}\,\mathrm{dm}_H(P) \le \mathrm{dm}_H(V_\lambda(P)) \le \frac{1+\mu}{2}\,\mathrm{dm}_H(P).$$

Demostración. Por definición de tolerancia $|\lambda| = \mu\dfrac{\mathrm{dm}_V(P)}{2}$. Substituimos en las desigualdades del lema 15 para un operador horizontal:

$$\mathrm{dm}_V(H_\lambda(P)) \ge \frac{\mathrm{dm}_V(P)}{2} - |\lambda| = \frac{\mathrm{dm}_V(P)}{2} - \mu\frac{\mathrm{dm}_V(P)}{2} = (1-\mu)\frac{\mathrm{dm}_V(P)}{2}.$$

y

$$\mathrm{dm}_V(H_\lambda(P)) \le \frac{\mathrm{dm}_V(P)}{2} + |\lambda| = \frac{\mathrm{dm}_V(P)}{2} + \mu\frac{\mathrm{dm}_V(P)}{2} = (1+\mu)\frac{\mathrm{dm}_V(P)}{2}.$$

El caso de un operador vertical es similar. □

Si un corte horizontal o vertical tiene tolerancia mayor o igual que 1, el lema anterior es trivialmente cierto, e insustancial, porque una de las subregiones que sale del corte tiene diámetro cero, y la otra tiene el mismo diámetro que P. Los cortes producidos por CID serán todos resultado de operadores de tolerancia menor que 1 (de hecho, menor que $^1/_2$), como veremos en la proposición 8.

Si C_i, para $i = 1, \ldots m$, denota un operador ya sea H_λ o $V_{\lambda'}$, la composición $C_m C_{m-1} \cdots C_2 C_1$ es una cadena de operadores. El concepto de tolerancia se extiende de operadores a cadenas de operadores de la siguiente manera. Para un

entero $m \geq 1$, un real $\mu \geq 0$ y una región P, decimos que una cadena de m operadores aplicada a P (es decir, $C_m \cdots C_1(P)$) tiene *tolerancia hasta* μ si, para cada $i = 1, \ldots m$, el i-ésimo operador C_i tiene tolerancia menor o igual que μ. Notemos que C_i se aplica a la región $C_{i-1} \cdots C_1(P)$, y por tanto la tolerancia de C_i depende del diámetro (horizontal o vertical según el tipo de C_i) de $C_{i-1} \cdots C_1(P)$, y no del diámetro de P.

El siguiente lema dice que el diámetro rectangular de cualquier región que surja de una cadena de tolerancia hasta μ (con $\mu < 1$) aplicada a P es menor que el de P. El factor de reducción tiene exponente $t_0 = \text{mín}(t_H, t_V)$ (siendo t_H y t_V el número de operadores de tipo horizontal y vertical, respectivamente, en la cadena).

Lema 17. *Para $m \geq 1$, supongamos que la cadena $C_m C_{m-1} \cdots C_1$ tiene un número t_H de operadores de tipo horizontal y t_V de tipo vertical, de modo que $t_H + t_V = m$. Si $C_m \cdots C_1(P)$ tiene tolerancia hasta μ con $\mu < 1$, entonces*

$$\text{dm}_R(C_m C_{m-1} \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{t_0} \text{dm}_R(P)$$

siendo $t_0 = \text{mín}(t_H, t_V)$.

Demostración. Mostraremos primero que se verifica tanto $\text{dm}_H(C_m C_{m-1} \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{t_V} \text{dm}_H(P)$ como $\text{dm}_V(C_m C_{m-1} \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{t_H} \text{dm}_V(P)$, y luego la afirmación sobre el diámetro rectangular.

Con respecto al diámetro horizontal $\text{dm}_H(C_m C_{m-1} \cdots C_1(P))$, consideremos en general $C_i C_{i-1} \cdots C_1(P)$ para $i = m, m-1, \ldots, 1$. Si C_i es horizontal, por la desigualdad $\text{dm}_H(H_\lambda(P)) \leq \text{dm}_H(P)$ del lema 15,

$$\text{dm}_H(C_i C_{i-1} \cdots C_1(P)) \leq \text{dm}_H(C_{i-1} \cdots C_1(P)).$$

Si C_i es vertical, por la desigualdad $\text{dm}_H(V_\lambda(P)) \leq \frac{1+\mu}{2} \text{dm}_H(P)$ del lema 16,

$$\text{dm}_H(C_i C_{i-1} \cdots C_1(P)) \leq \frac{1+\mu}{2} \text{dm}_H(C_{i-1} \cdots C_1(P)).$$

Empezando por $\text{dm}_H(C_m C_{m-1} \cdots C_1(P))$, aplicaremos m veces alguna de las dos desigualdades anteriores, la primera si C_i es horizontal y la segunda si C_i es vertical, de modo iterativo. Así, por la primera desigualdad, si C_m es horizontal,

o si no por la segunda, si C_m es vertical, tenemos

$$\mathrm{dm}_H(C_m \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{\beta_m} \mathrm{dm}_H(C_{m-1} \cdots C_1(P))$$

siendo $\beta_m = 0$ en el primer caso y $\beta_m = 1$ en el segundo. Después aplicamos por segunda vez una de las desigualdades, ahora al factor $\mathrm{dm}_H(C_{m-1} \cdots C_1(P))$ que aparece en el segundo miembro anterior. Aplicamos la primera desigualdad si C_{m-1} es horizontal, la segunda en el otro caso, dando

$$\left(\frac{1+\mu}{2}\right)^{\beta_m} \mathrm{dm}_H(C_{m-1} \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{\beta_m} \left(\frac{1+\mu}{2}\right)^{\beta_{m-1}} \mathrm{dm}_H(C_{m-2} \cdots C_1(P))$$

siendo $\beta_{m-1} = 0$ si C_{m-1} es horizontal y $\beta_{m-1} = 1$ si es vertical. Aplicamos por tercera vez alguna de las dos desigualdades al segundo miembro anterior, dando

$$\left(\frac{1+\mu}{2}\right)^{\beta_m} \left(\frac{1+\mu}{2}\right)^{\beta_{m-1}} \mathrm{dm}_H(C_{m-2} \cdots C_1(P)) \leq$$
$$\leq \left(\frac{1+\mu}{2}\right)^{\beta_m} \left(\frac{1+\mu}{2}\right)^{\beta_{m-1}} \left(\frac{1+\mu}{2}\right)^{\beta_{m-2}} \mathrm{dm}_H(C_{m-3} \cdots C_1(P))$$

donde β_{m-2} es 0 si C_{m-2} es horizontal y 1 en caso contrario. Haciendo esto para $i = m, m-1, \ldots, 1$, tenemos una cadena de desigualdades

$$\mathrm{dm}_H(C_m \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{\beta_m} \mathrm{dm}_H(C_{m-1} \cdots C_1(P)) \leq$$
$$\leq \left(\frac{1+\mu}{2}\right)^{\beta_m} \left(\frac{1+\mu}{2}\right)^{\beta_{m-1}} \mathrm{dm}_H(C_{m-2} \cdots C_1(P)) \leq \cdots$$
$$\cdots \leq \left(\frac{1+\mu}{2}\right)^{\beta_m} \left(\frac{1+\mu}{2}\right)^{\beta_{m-1}} \cdots \left(\frac{1+\mu}{2}\right)^{\beta_1} \mathrm{dm}_H(P)$$

donde $\beta_i = 0$ si C_i es horizontal y $\beta_i = 1$ si C_i es vertical. Luego la suma de los exponentes $\sum_{i=1}^{m} \beta_i$ es t_V, el número de operadores verticales en $C_m, C_{m-2}, \cdots, C_1$. Por tanto $\mathrm{dm}_H(C_m \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{t_V} \mathrm{dm}_H(P)$.

De modo similar $\mathrm{dm}_V(C_m \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{t_H} \mathrm{dm}_V(P)$, usando las de-

sigualdades análogas de los lemas 15 y 16 para el diámetro vertical. Por tanto

$$\mathrm{dm}_R(C_m \cdots C_1(P)) = \sqrt{\mathrm{dm}_H(C_m \cdots C_1(P))^2 + \mathrm{dm}_V(C_m \cdots C_1(P))^2} \leq$$

$$\leq \sqrt{\left(\frac{1+\mu}{2}\right)^{2t_V} \mathrm{dm}_H(P)^2 + \left(\frac{1+\mu}{2}\right)^{2t_H} \mathrm{dm}_V(P)^2} =$$

$$= \left(\frac{1+\mu}{2}\right)^{t_0} \sqrt{\left(\frac{1+\mu}{2}\right)^{2(t_V-t_0)} \mathrm{dm}_H(P)^2 + \left(\frac{1+\mu}{2}\right)^{2(t_H-t_0)} \mathrm{dm}_V(P)^2}$$

Uno de los exponentes, $2(t_V - t_0)$ o $2(t_H - t_0)$, es igual a 0. Supongamos que es el primero (equivalentemente, que $t_0 = t_V$), siendo el otro caso similar. Además $\frac{1+\mu}{2} \leq 1$, lo que implica $\left(\frac{1+\mu}{2}\right)^{2(t_H-t_0)} \leq 1$. Por tanto

$$\left(\frac{1+\mu}{2}\right)^{t_0} \sqrt{\left(\frac{1+\mu}{2}\right)^{2(t_V-t_0)} \mathrm{dm}_H(P)^2 + \left(\frac{1+\mu}{2}\right)^{2(t_H-t_0)} \mathrm{dm}_V(P)^2} =$$

$$= \left(\frac{1+\mu}{2}\right)^{t_0} \sqrt{\mathrm{dm}_H(P)^2 + \left(\frac{1+\mu}{2}\right)^{2(t_H-t_0)} \mathrm{dm}_V(P)^2} \leq$$

$$\leq \left(\frac{1+\mu}{2}\right)^{t_0} \sqrt{\mathrm{dm}_H(P)^2 + \mathrm{dm}_V(P)^2} = \left(\frac{1+\mu}{2}\right)^{t_0} \mathrm{dm}_R(P).$$

□

Damos ahora un último lema 18 acerca del cambio en diámetro producido por un operador de corte aplicado a P. Informalmente, dice que como P es convexa, el decrecimiento en diámetro es moderado, es decir solo hasta un factor de $\frac{1-\mu}{2}$. Sabemos que las regiones producidas por PRec son convexas como se comenta justo antes de la subsección 3.2.1. Por tanto el lema 18 se puede aplicar a cada región de la forma $C_m \cdots C_2 C_1(P)$. El lema 18 se usa en la prueba del lema 20.

Lema 18. *Supongamos que P es convexa y $\mu < 1$. Si $H_\lambda(P)$ tiene una tolerancia de μ, entonces*

$$\frac{1-\mu}{2} \mathrm{dm}_H(P) \leq \mathrm{dm}_H(H_\lambda(P)).$$

Si $V_\lambda(P)$ tiene una tolerancia de μ, entonces

$$\frac{1-\mu}{2}\,\mathrm{dm}_V(P) \leq \mathrm{dm}_V(V_\lambda(P)).$$

Demostración. La intersección de P con la línea de soporte superior l_T es, por convexidad, un segmento I_T quizás reducido a un punto. Similarmente I_B, I_L y I_R son las intersecciones de P con las líneas l_B, l_L y l_R, respectivamente (figura 3.25). Llamemos E_C a la envolvente convexa [Knuth, 1992] de los segmentos I_T, I_R, I_B, I_L. Como P es convexa, contiene esta envolvente convexa E_C.

Probamos primero la desigualdad para el operador horizontal T_λ. Consideremos el triángulo D_T que une cualquier punto de I_T con la base del rectángulo de líneas de soporte que rodea a P, mostrado en la figura 3.25.

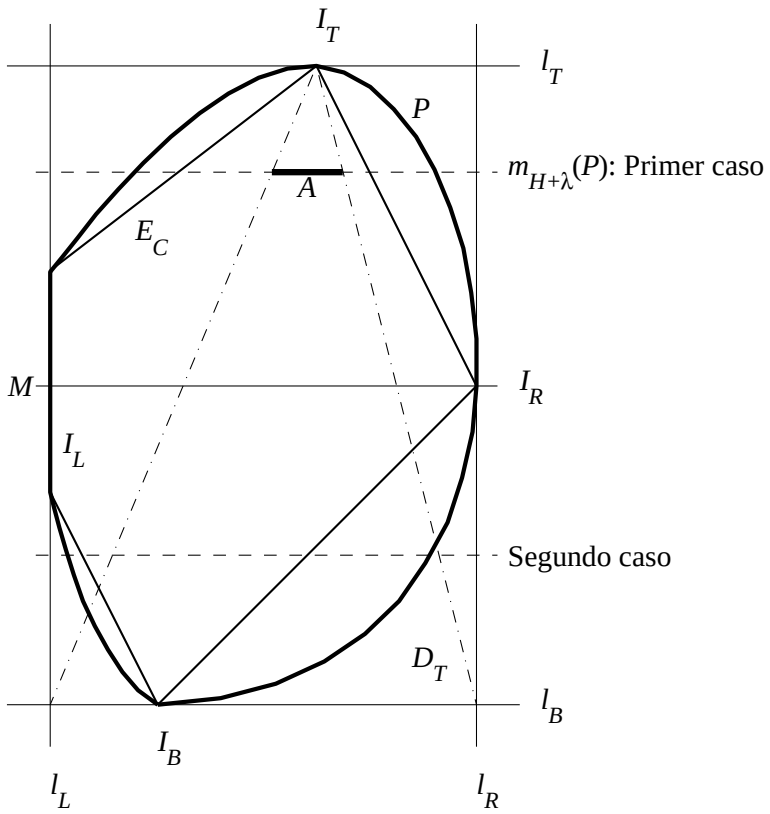


Figura 3.25: Una región convexa P (trazo grueso), la envolvente convexa E_C de sus intersecciones con las líneas de soporte (delgado), y el triángulo D_T que une cualquier punto de I_T con la línea opuesta (punteado).

Consideremos los dos puntos que están a máxima altura en I_L y I_R respectivamente, y tomemos el menor. Por ejemplo en la figura 3.25 este es el único puno que pertenece a I_R. Una línea horizontal M por este punto divide la altura de P en dos intervalos. Consideraremos dos casos: primero, la línea $m_{H+\lambda}(P)$ que produce $T_\lambda(P)$ está por arriba o coincide con M; segundo, $m_{H+\lambda}(P)$ está bajo M.

En el primer caso, $T_\lambda(P)$ contiene la parte de la envolvente convexa E_C por encima de $m_{H+\lambda}(P)$, que a su vez contiene la parte del triángulo D_T por encima de esta línea. Es decir $T_\lambda(P) \supset E_C \cap T_\lambda(P) \supset D_T \cap T_\lambda(P)$. Esta cadena de inclusiones, por el lema 11, implica $\mathrm{dm}_H(T_\lambda(P)) \geq \mathrm{dm}_H(E_C \cap T_\lambda(P)) \geq \mathrm{dm}_H(D_T \cap T_\lambda(P))$. Mostraremos que $\mathrm{dm}_H(D_T \cap T_\lambda(P)) \geq \dfrac{1-\mu}{2}\,\mathrm{dm}_H(P)$. El segmento $A = m_{H+\lambda}(P) \cap D_T$ nos da el diámetro horizontal de $D_T \cap T_\lambda(P)$. A está a altura $\dfrac{\mathrm{dm}_V(P)}{2} + \lambda$ en el triángulo D_T, que tiene base $\mathrm{dm}_H(P)$ y altura $\mathrm{dm}_V(P)$. Por tanto por una regla de tres tiene una longitud de $\left(\dfrac{\mathrm{dm}_V(P)}{2} - \lambda\right)\dfrac{\mathrm{dm}_H(P)}{\mathrm{dm}_V(P)}$. Además, como $\dfrac{2|\lambda|}{\mathrm{dm}_V(P)} = \mu$, luego $\dfrac{2\lambda}{\mathrm{dm}_V(P)} \leq \mu$ y entonces $\lambda \leq \dfrac{\mu\,\mathrm{dm}_V(P)}{2}$, así que $-\lambda \geq -\dfrac{\mu\,\mathrm{dm}_V(P)}{2}$. Por tanto

$$\mathrm{long}(A) = \left(\frac{\mathrm{dm}_V(P)}{2} - \lambda\right)\frac{\mathrm{dm}_H(P)}{\mathrm{dm}_V(P)} \geq \left(\frac{1}{2} - \frac{\mu}{2}\right)\mathrm{dm}_H(P)$$

y $\mathrm{dm}_H(D_T \cap T_\lambda(P)) = \mathrm{long}(A) \geq \dfrac{1-\mu}{2}\,\mathrm{dm}_H(P)$. Por la anterior cadena de desigualdades, $\mathrm{dm}_H(T_\lambda(P)) \geq \dfrac{1-\mu}{2}\,\mathrm{dm}_H(P)$.

En el segundo caso, como μ es positivo, $-\mu \leq \mu$ luego $\dfrac{1-\mu}{2} < \dfrac{1+\mu}{2}$, y como $\mu < 1$, $\dfrac{1+\mu}{2} < 1$. Por tanto encadenando estas desigualdades $\dfrac{1-\mu}{2} < 1$, así que $\dfrac{1-\mu}{2}\,\mathrm{dm}_H(P) \leq \mathrm{dm}_H(P)$. Esto también es cierto en el primer caso, pero en este segundo caso además $T_\lambda(P)$ contiene al menos un punto de I_L y uno de I_R (los puntos a máxima altura de I_L y de I_R están situados en $T_\lambda(P)$). Luego la anchura de $T_\lambda(P)$ coincide con la anchura de P, que es $\mathrm{dm}_H(P) = \mathrm{dm}_H(T_\lambda(P))$. Por tanto $\dfrac{1-\mu}{2}\,\mathrm{dm}_H(P) \leq \mathrm{dm}_H(T_\lambda(P))$ y concluimos.

Para los otros operadores L_λ, R_λ y B_λ, la prueba anterior puede reescribirse con los cambios adecuados.

□

La igualdad en el lema anterior se alcanza, por ejemplo, en el caso de que P

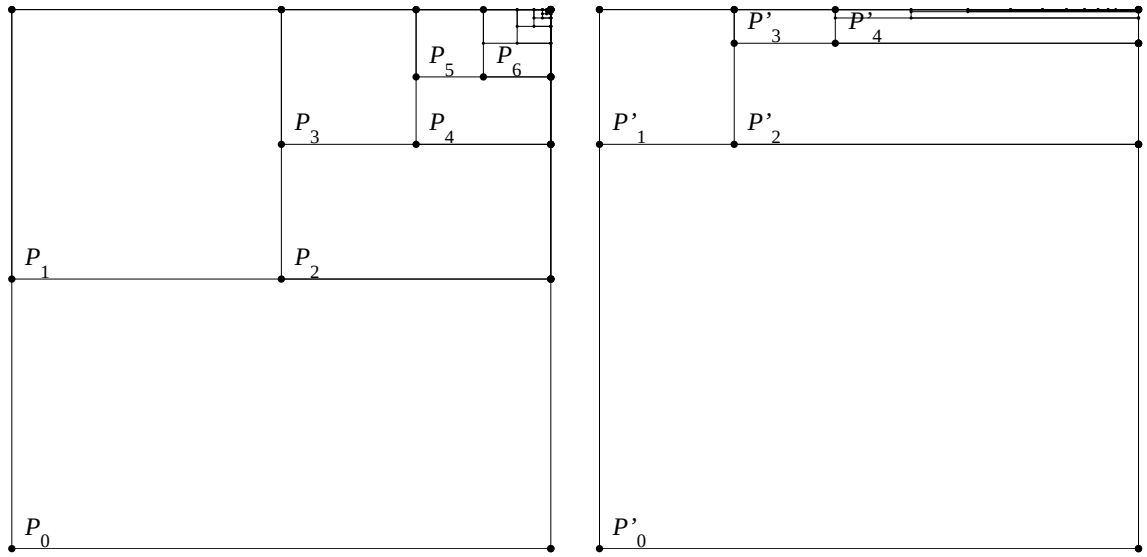
sea un triángulo rectángulo con catetos paralelos a los ejes.

3.3.2.2. Relación de aspecto

Ahora estudiamos la elongación de las regiones, que viene dada por la relación de aspecto. Como se comenta en la introducción de esta sección, estamos interesados en la reducción en diámetro en función del número de cortes M. El lema anterior 17 nos da el factor de reducción del diámetro, $\left(\frac{1+\mu}{2}\right)^{t_0}$, como función del mínimo número t_0 de operadores de cada tipo en la cadena $C_m C_{m-1} \cdots C_1(P)$. Para relacionar la longitud m de una cadena con el mínimo t_0, necesitamos discutir la relación de aspecto de P. Que la relación de aspecto está conectada con la relación entre t_0 y m se puede ver en la figura 3.23. La región está elongada horizontalmente y por tanto requiere varios cortes verticales, tres, hasta el primero horizontal. Es decir la cadena $C_4C_3C_2C_1(P)$ tiene $m = 4$ y $t_0 = 1$. Para una región con mayor elongación, el número m de cortes hasta el primer cambio de tipo es aún mayor, mientras que sigue siendo $t_0 = 1$.

La *relación de aspecto horizontal* de una región P es $\mathrm{asp}_H(P) = \dfrac{\mathrm{dm}_H(P)}{\mathrm{dm}_V(P)}$, y la *relación de aspecto vertical* es su recíproco $\mathrm{asp}_V(P) = \dfrac{\mathrm{dm}_V(P)}{\mathrm{dm}_H(P)}$. La *relación de aspecto* (por abreviar, aspecto) es $\mathrm{asp}(P) = \mathrm{máx}(\mathrm{asp}_H(P), \mathrm{asp}_V(P))$. La figura 3.26 *a*) muestra las subregiones que surgen tras la aplicación de cortes por la línea media, alternativamente T y R. La región inicial de la figura, P_0, es el cuadrado unidad (de aspecto 1), y su subregión $P_1 = T(P_0)$ tiene $\mathrm{asp}_H(P_1) = 2$ y $\mathrm{asp}_V(P_1) = {}^1\!/_2$. La subregión $P_2 = R(P_1)$ de P_1 tiene aspecto 1. Llamando $P_i = T(P_{i-1})$ si i es impar y $P_i = R(P_{i-1})$ si i es par, el aspecto de las regiones P_i alterna entre 1 y 2. En general, tras un corte horizontal y vertical (por la línea media), la región resultante tiene el mismo aspecto que la inicial.

Con cortes desplazados aparece un fenómeno: el aspecto puede incrementarse con el número de cortes. Es más, el incremento puede ser una función exponencial de este número. Esta situación surge si los desplazamientos de estos cortes están sesgados consistentemente en la misma dirección. La figura 3.26 *b)* muestra las subregiones obtenidas tras la aplicación alternativa de T_λ y $R_{\lambda'}$, correspondiendo λ y λ' a una tolerancia de $\mu = {}^1\!/_2$. La región inicial P'_0 es el cuadrado unidad, como en el ejemplo anterior, y $P'_i = T_\lambda(P'_{i-1})$ si i es impar y $P'_i = R_{\lambda'}(P'_{i-1})$ si i es



a) Alternando T y R *b*) Alternando T_λ y $R_{\lambda'}$

Figura 3.26: Un par de cortes, horizontal y vertical, por la línea media no altera el aspecto. Sin embargo, tras sucesivos pares de cortes horizontales y verticales, desplazados, y con cierta tendencia en sus desplazamientos, el aspecto resultante crece indefinidamente.

par. Los aspectos de $P'_1, P'_2, P'_3, \ldots$ son respectivamente $4, 3, 12, 9, 36, 27, \ldots$ Puede verse que $\operatorname{asp}(P'_{2i}) = 3^i$ y $\operatorname{asp}(P'_{2i+1}) = 4 \cdot 3^{i-1}$.

Queremos subregiones con bajo aspecto por razones de coste: por el teorema 6 *a*), el coste de PCR (Γ, θ) es directamente proporcional a la longitud de Γ, $b - a$. Específicamente, la cota del coste es $\left\lfloor \dfrac{b-a}{\theta} + 1 \right\rfloor$. Por la parametrización uniforme del borde Γ, $b - a = \operatorname{arclength}(\Gamma)$. Para una región general P, la longitud de su borde es mayor que $2\,\mathrm{dm}_H(P)$. Por tanto la cota del coste cumple $\left\lfloor \dfrac{b-a}{\theta} + 1 \right\rfloor \geq \dfrac{b-a}{\theta} \geq \dfrac{2\,\mathrm{dm}_H(P)}{\theta}$. Por la proposición 6, para evitar un retorno con error en PCR el parámetro θ debe ser menor que $\Theta_H = K\,\mathrm{dm}_V(P)$ para una constante K. Luego $\dfrac{2\,\mathrm{dm}_H(P)}{\theta} \geq \dfrac{2\,\mathrm{dm}_H(P)}{\Theta_H} = \dfrac{2\,\mathrm{dm}_H(P)}{K\,\mathrm{dm}_V(P)}$. Finalmente supongamos que P es más ancha que alta (el caso contrario es similar), por tanto $\operatorname{asp}(P) = \operatorname{asp}_H(P) = \dfrac{\mathrm{dm}_H(P)}{\mathrm{dm}_V(P)}$ y entonces $\mathrm{dm}_H(P) = \operatorname{asp}(P)\,\mathrm{dm}_V(P)$. Encadenando las desigualdades anteriores, la cota del coste es $\left\lfloor \dfrac{b-a}{\theta} + 1 \right\rfloor \geq \dfrac{2\operatorname{asp}(P)\,\mathrm{dm}_V(P)}{K\,\mathrm{dm}_V(P)} = \dfrac{2}{K}\operatorname{asp}(P)$.

Es decir, la cota del coste es mayor que el aspecto por un factor. Si el aspecto de las subregiones crece indefinidamente, también lo hace la cota del coste. Esta es la principal motivación para evitar subregiones elongadas, lo que se consigue cortando a lo largo del eje menor en vez de alternar cortes horizontales y verticales.

Probaremos dos resultados sobre relación de aspecto y número mínimo de operadores de cada tipo. El primer resultado dice que cualquier cadena de cortes por el eje menor aplicada a P, si es lo bastante larga y P tiene baja relación de aspecto, entonces contiene cortes de ambos tipos (lema 19). El segundo resultado dice que si, para un m dado, la cadena $C_m C_{m-1} \cdots C_1(P)$ de cortes por el eje menor tiene los dos cortes C_m y C_{m-1} de tipo diferente, entonces la región subyacente tiene baja relación de aspecto. Este es el lema 20, parafraseado aquí para simplificar la exposición. Por ejemplo en la figura 3.23, tras la acción de los cortes C_1, C_2 y C_3 a lo largo del mismo eje, el corte C_4 da lugar a dos regiones con baja relación de aspecto.

Para precisar estas afirmaciones consideramos la tolerancia de los cortes. Recordemos que un corte con tolerancia 1/2 se da a mitad de camino entre la línea central y una de las dos líneas de soporte. Para este valor particular de la tolerancia, el lema 19 afirma "Si la cadena $C_m C_{m-1} \cdots C_1(P)$ de cortes por el eje menor tiene tolerancia hasta $1/2$, y $m \geq 2\lg_2(\text{asp}(P))$, entonces esta cadena tiene dos cortes de distinto tipo". El lema 20 para este valor particular de la tolerancia afirma "Tras una cadena $C_m C_{m-1} \cdots C_1(P)$ de cortes por el eje menor, de tolerancia hasta $1/2$, con C_m y C_{m-1} de distinto tipo, la subregión que surge tiene relación de aspecto menor que 12". Los desplazamientos de la figura 3.23 tienen tolerancia menor que $1/2$, y por tanto la relación de aspecto de las subregiones obtenidas tras dos cortes de distinto tipo es menor que 12. Estos dos lemas unidos implican que, a pesar del desplazamiento, la relación de aspecto no puede crecer indefinidamente: para una región P de relación de aspecto arbitraria $\text{asp}(P)$, si la tolerancia de cada corte es menor o igual que $1/2$, una cadena $C_m C_{m-1} \cdots C_1(P)$ de cortes por el eje menor contendrá dos cortes $C_k C_{k-1}$ de distinto tipo si $m \geq 2\lg_2(\text{asp}(P))$ (por el lema 19). Por tanto la relación de aspecto de la región resultante tras la cadena $C_k C_{k-1} \cdots C_1(P)$ será menor que 12 (por el lema 20). Hemos ilustrado los lemas con una tolerancia menor o igual que $1/2$ no solo como mero ejemplo, sino también porque aplicaremos los resultados de esta sección a los cortes hechos por PRec, que tienen tolerancia menor o igual que $1/2$ como veremos en la proposición 8*a*).

Usando estos lemas mostraremos que una cadena de m cortes tiene un número mínimo $t_0 \leq \dfrac{m - L_0}{L_1}$ de operadores de cada tipo, para ciertos L_0, L_1 que dependen de la relación de aspecto y la tolerancia.

Lema 19. *Si P es una región convexa, y $C_k C_{k-1} \cdots C_2 C_1(P)$ es una cadena con al menos dos operadores, de tolerancia hasta μ, tales que C_i es un corte por el eje menor de $C_{i-1} \cdots C_1(P)$, y $k > 1 + \dfrac{\lg_2(\text{asp}(P))}{1 - \lg_2(1+\mu)}$, entonces los operadores C_i no pueden ser todos del mismo tipo, horizontal o vertical.*

Demostración. Demostraremos que los operadores no pueden ser todos horizontales, siendo similar la demostración de la otra imposibilidad. Si C_i es horizontal para $i = 1, 2 \ldots, k$, alcanzaremos una contradicción. Como C_k es horizontal entonces el eje menor de $C_{k-1} \cdots C_1(P)$ es horizontal, es decir $\text{dm}_H(C_{k-1} \cdots C_1(P)) \leq \text{dm}_V(C_{k-1} \cdots C_1(P))$, o

$$1 \leq \frac{\text{dm}_V(C_{k-1} \cdots C_1(P))}{\text{dm}_H(C_{k-1} \cdots C_1(P))}.$$

Además, por las desigualdades del 15 (en el denominador) y el lema 16 (en el numerador) para operadores horizontal C_i, empezando con $i = k-1$ y bajando hasta $i = 1$:

$$\frac{\text{dm}_V(C_{k-1} \cdots C_1(P))}{\text{dm}_H(C_{k-1} \cdots C_1(P))} \leq \frac{\frac{1+\mu}{2}\,\text{dm}_V(C_{k-2} \cdots C_1(P))}{\text{dm}_H(C_{k-2} \cdots C_1(P))} \leq$$
$$\leq \frac{\left(\frac{1+\mu}{2}\right)^2 \text{dm}_V(C_{k-3} \cdots C_1(P))}{\text{dm}_H(C_{k-3} \cdots C_1(P))} \leq \cdots \leq \frac{\left(\frac{1+\mu}{2}\right)^{k-1} \text{dm}_V(P)}{\text{dm}_H(P)}$$

Luego $1 \leq \left(\dfrac{1+\mu}{2}\right)^{k-1} \dfrac{\text{dm}_V(P)}{\text{dm}_H(P)}$. Notemos que $\text{asp}(P) = \dfrac{\text{dm}_V(P)}{\text{dm}_H(P)}$, porque C_1 es horizontal. Luego despejando k:

$$1 \leq \left(\frac{1+\mu}{2}\right)^{k-1} \text{asp}(P)$$
$$\left(\frac{2}{1+\mu}\right)^{k-1} \leq \text{asp}(P)$$
$$(k-1) \lg_2 \left(\frac{2}{1+\mu}\right) \leq \lg_2(\text{asp}(P))$$

$$(k-1) \leq \frac{\lg_2(\text{asp}(P))}{\lg_2 \frac{2}{1+\mu}}.$$

como $\dfrac{\lg_2(\text{asp}(P))}{\lg_2 \frac{2}{1+\mu}} = \dfrac{\lg_2(\text{asp}(P))}{1-\lg_2(1+\mu)}$, esto implica $(k-1) \leq \dfrac{\lg_2(\text{asp}(P))}{1-\lg_2(1+\mu)}$. Que $k \leq 1 + \dfrac{\lg_2(\text{asp}(P))}{1-\lg_2(1+\mu)}$ está en contradicción con la hipótesis $k > 1 + \dfrac{\lg_2(\text{asp}(P))}{1-\lg_2(1+\mu)}$. □

Intuitivamente el siguiente lema dice que tras dos cortes (por el eje menor) de distinto tipo, la región resultante no es muy elongada.

Lema 20. *Si P es una región convexa, $C_2C_1(P)$ es una cadena de operadores de tolerancia hasta μ, uno de tipo horizontal y el otro vertical, tales que C_2 es un corte por el eje menor de $C_1(P)$ y C_1 por el eje menor de P, entonces*

$$\text{asp}(C_2C_1(P)) \leq \frac{2(1+\mu)}{(1-\mu)^2}$$

Demostración. Supongamos que C_2 es un operador vertical $V_{\lambda'}$ y C_1 es horizontal, H_λ, siendo similar el caso contrario.

Por definición $\text{asp}(V_{\lambda'}H_\lambda(P))$ es el máximo de $\text{asp}_V(V_{\lambda'}H_\lambda(P)) = \dfrac{\text{dm}_V(V_{\lambda'}H_\lambda(P))}{\text{dm}_H(V_{\lambda'}H_\lambda(P))}$ y $\text{asp}_H(V_{\lambda'}H_\lambda(P)) = \dfrac{\text{dm}_H(V_{\lambda'}H_\lambda(P))}{\text{dm}_V(V_{\lambda'}H_\lambda(P))}$. Por las desigualdades del lema 15 y el lema 16, tenemos que

$$\frac{\text{dm}_V(V_{\lambda'}H_\lambda(P))}{\text{dm}_H(V_{\lambda'}H_\lambda(P))} \leq \frac{\text{dm}_V(H_\lambda(P))}{\frac{1-\mu}{2}\,\text{dm}_H(H_\lambda(P))} = \frac{2}{1-\mu}\frac{\text{dm}_V(H_\lambda(P))}{\text{dm}_H(H_\lambda(P))} \leq \frac{2}{1-\mu}$$

La última desigualdad viene del hecho de que el corte por el eje menor de $H_\lambda(P)$ es vertical, luego $\text{dm}_V(H_\lambda(P)) \leq \text{dm}_H(H_\lambda(P))$, es decir $\dfrac{\text{dm}_V(H_\lambda(P))}{\text{dm}_H(H_\lambda(P))} \leq 1$.

Por las desigualdades del lema 16 (numerador) y el lema 18 (denominador),

$$\frac{\text{dm}_H(V_{\lambda'}H_\lambda(P))}{\text{dm}_V(V_{\lambda'}H_\lambda(P))} \leq \frac{\frac{1+\mu}{2}\,\text{dm}_H(H_\lambda(P))}{\frac{1-\mu}{2}\,\text{dm}_V(H_\lambda(P))} \leq \frac{\frac{1+\mu}{2}\,\text{dm}_H(P)}{\left(\frac{1-\mu}{2}\right)^2\text{dm}_V(P)} =$$

$$= \frac{2(1+\mu)}{(1-\mu)^2}\frac{\text{dm}_H(P)}{\text{dm}_V(P)} \leq \frac{2(1+\mu)}{(1-\mu)^2}$$

La última desigualdad viene del hecho de que el corte por el eje menor de P es horizontal, luego $\mathrm{dm}_H(P) \leq \mathrm{dm}_V(P)$, es decir $\dfrac{\mathrm{dm}_H(P)}{\mathrm{dm}_V(P)} \leq 1$.

Como $\mathrm{asp}(V_{\lambda'}H_\lambda(P)) = \max(\mathrm{asp}_V(V_{\lambda'}H_\lambda(P)), \mathrm{asp}_H(V_{\lambda'}H_\lambda(P)))$, y como el máximo de dos valores está acotado por el máximo de las cotas de cada valor, tenemos que $\mathrm{asp}(V_{\lambda'}H_\lambda(P)) \leq \max\left(\dfrac{2}{1-\mu}, \dfrac{2(1+\mu)}{(1-\mu)^2}\right)$, y concluimos, porque $1 \leq \dfrac{1+\mu}{1-\mu}$ y por tanto multiplicando por $\dfrac{2}{1-\mu}$, $\dfrac{2}{1-\mu} \leq \dfrac{2(1+\mu)}{(1-\mu)^2}$.

□

Finalmente, aplicamos los lemas anteriores para tener una reducción en diámetro:

Lema 21. *Si P es una región convexa y $C_mC_{m-1}\cdots C_2C_1(P)$ es una cadena de operadores de tolerancia hasta μ, con $\mu < 1$, y tal que C_i es un corte por el eje menor de $C_{i-1}\cdots C_1(P)$, entonces*

$$\mathrm{dm}_R(C_m\cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{\frac{m-L_0}{L_1}} \mathrm{dm}_R(P)$$

donde $L_0 = 2 + \dfrac{\lg_2(\mathrm{asp}(P))}{1-\lg_2(1+\mu)}$ *y* $L_1 = 1 + 2\dfrac{1-\lg_2(1-\mu)}{1-\lg_2(1+\mu)}$.

Demostración. En caso de una cadena corta (lo que significa que $m < L_0$, equivalentemente $m - L_0 < 0$, luego $\dfrac{m-L_0}{L_1} < 0$) el factor $\left(\dfrac{1+\mu}{2}\right)^{\frac{m-L_0}{L_1}}$ es mayor que 1, y la afirmación del lema es trivialmente cierta.

En cualquier caso, dividiremos la cadena $C_mC_{m-1}\cdots C_1$ en $t+1$ subcadenas, determinadas por los subíndices $k_1, k_2, \ldots, k_t$, (para cierto valor t). Es decir, la primera subcadena es $C_{k_1}\cdots C_1$, la segunda $C_{k_2}C_{k_2-1}\cdots C_{k_1+2}C_{k_1+1}$, etcétera. Así tenemos $C_{k_{i+1}}C_{k_{i+1}-1}\cdots C_{k_i+2}C_{k_i+1}$ para $i = 1, \ldots, t-1$, y una subcadena final $C_mC_{m-1}\cdots C_{k_t+1}$. Definiremos los k_i de tal modo que cada subcadena (menos la última) tendrá al menos un operador de cada tipo. También acotaremos el número de operadores en cada subcadena, es decir las longitudes: k_1 de la primera , $k_{i+1}-k_i$ (para $i = 1, \ldots, t-1$) y $m - k_t$ para las otras.

Definimos k_1 como el índice del primer cambio de tipo en $C_mC_{m-1}\cdots C_1$, es decir, $C_{k_1-1}C_{k_1-2}\cdots C_2C_1$ son todos del mismo tipo, horizontal o vertical, distin-

to del de C_{k_1}. Como $L_0 = 2 + \dfrac{\lg_2(\text{asp}(P))}{1-\lg_2(1+\mu)}$, entonces $\lfloor L_0 \rfloor > 1 + \dfrac{\lg_2(\text{asp}(P))}{1-\lg_2(1+\mu)}$. Aplicamos el lema 19 a la cadena $C_{\lfloor L_0 \rfloor} C_{\lfloor L_0 \rfloor - 1} \cdots C_1(P)$, que por tanto tiene al menos dos operadores de distinto tipo. Luego $k_1 \leq \lfloor L_0 \rfloor$. Esto implica que la longitud de la primera subcadena, k_1, verifica $k_1 \leq L_0$, hecho que usaremos más adelante.

Llamemos $P_0 = P$, y $P_i = C_i C_{i-1} \cdots C_1(P)$ para $i = 1, \ldots, m$. Como C_{k_1} y C_{k_1-1} son de distinto tipo, por el lema 20 $\text{asp}(P_{k_1}) = \text{asp}(C_{k_1} C_{k_1-1}(P_{k_1-2})) \leq \dfrac{2(1+\mu)}{(1-\mu)^2}$. Sea k_2 el índice del siguiente cambio de tipo después de k_1, es decir, $C_{k_2-1} C_{k_2-2} \cdots C_{k_1+2} C_{k_1+1}$ son del mismo tipo, distinto del de C_{k_2}. Por ejemplo dos cadenas con tipos HVVVHVVV o VHHHHVVV tienen ambas $k_1 = 4$ y $k_2 = 8$. Notemos que la subcadena $C_{k_2-1} C_{k_2-1} \cdots C_{k_1+2} C_{k_1+1}$ tiene al menos dos operadores. Aplicando el recíproco del lema 19 a la cadena $C_{k_2-1} \cdots C_{k_1+1}(P_{k_1})$, como los operadores son del mismo tipo, tenemos que el número de operadores que contiene, que es $k_2 - k_1 - 1$, es menor o igual que $1 + \dfrac{\lg_2(\text{asp}(P_{k_1}))}{1-\lg_2(1+\mu)}$. Como $\text{asp}(P_{k_1}) \leq \dfrac{2(1+\mu)}{(1-\mu)^2}$, entonces $k_2 - k_1 - 1 \leq 1 + \dfrac{\lg_2 \frac{2(1+\mu)}{(1-\mu)^2}}{1-\lg_2(1+\mu)}$. Lo que implica que

$$
\begin{aligned}
k_2 - k_1 \leq 2 + \frac{\lg_2 \frac{2(1+\mu)}{(1-\mu)^2}}{1-\lg_2(1+\mu)} = 2 + \frac{1 + \lg_2(1+\mu) - 2\lg_2(1-\mu)}{1-\lg_2(1+\mu)} = \\
= \frac{2 - 2\lg_2(1+\mu) + 1 + \lg_2(1+\mu) - 2\lg_2(1-\mu)}{1-\lg_2(1+\mu)} = \\
= \frac{3 - \lg_2(1+\mu) - 2\lg_2(1-\mu)}{1-\lg_2(1+\mu)} = 1 + 2\frac{1-\lg_2(1-\mu)}{1-\lg_2(1+\mu)} = L_1.
\end{aligned}
$$

Luego la longitud de la segunda subcadena verifica $k_2 - k_1 \leq L_1$. El siguiente índice de cambio de tipo k_3 tras k_2 también verifica que $k_3 - k_2 \leq L_1$, por un razonamiento similar (el recíproco del lema 19 aplicado a la cadena $C_{k_3-1} \cdots C_{k_2+1}(P_{k_2})$). Por tanto la tercera subcadena también tiene longitud menor o igual que L_1. Repetimos el razonamiento para los índices de los siguientes cambios de tipo, concluyendo que, si hay t cambios de tipo, de índices $k_1, k_2, \ldots k_t$, tenemos $k_1 \leq L_0$ y $k_i - k_{i-1} \leq L_1$ para $i = 1, \ldots, t$. También tenemos que la subcadena tras el último cambio de tipo, $C_m C_{m-1} \cdots C_{k_t+1}$, está compuesta de operadores del mismo tipo, luego por el recíproco del lema 19 su longitud es $m - k_t \leq 1 + \dfrac{\lg_2(\text{asp}(P))}{1-\lg_2(1+\mu)}$, lo que implica $m - k_t \leq L_1 - 1$ y por supuesto $m - k_t \leq L_1$.

Como las t subcadenas que no son la final están todas compuestas de dos o más operadores, todos del mismo tipo excepto el último, entonces hay al menos t operadores de cada tipo, uno en cada subcadena. Si t_H es el número de operadores de tipo horizontal en $C_m C_{m-1} \cdots C_1$, y t_V los de tipo vertical, tenemos que $t \leq \text{mín}(t_H, t_V)$. Acotaremos inferiormente t contabilizando el número de operadores (la longitud) de la primera subcadena, acotada superiormente por L_0, y de las otras subcadenas, acotada superiormente por L_1. La longitud de la cadena completa, m, es la suma de las longitudes de las subcadenas. Por tanto tenemos que $m \leq L_0 + tL_1$, es decir, $\dfrac{m - L_0}{L_1} \leq t$. Luego $\dfrac{m - L_0}{L_1} \leq \text{mín}(t_H, t_V)$, y, como $\dfrac{1+\mu}{2} \leq 1$, entonces $\left(\dfrac{1+\mu}{2}\right)^{\text{mín}(t_H,t_V)} \leq \left(\dfrac{1+\mu}{2}\right)^{\frac{m-L_0}{L_1}}$. Substituyendo en la desigualdad del lema 17 tenemos que

$$\text{dm}_R(C_m \cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{\text{mín}(t_H,t_V)} \text{dm}_R(P) \leq \left(\frac{1+\mu}{2}\right)^{\frac{m-L_0}{L_1}} \text{dm}_R(P)$$

como queríamos demostrar.

□

Para el decrecimiento del diámetro hasta la precisión A, tenemos que:

Proposición 7. *Si $C_m C_{m-1} \cdots C_1(P)$ es de tolerancia hasta μ, con $\mu \leq 1$ y $m \geq \dfrac{\lg_2 \frac{\text{dm}_R(P)}{A}}{1 - \lg_2(1+\mu)} L_1 + L_0$, entonces $\text{dm}_R(C_m \cdots C_1(P)) \leq A$.*

Demostración. La hipótesis $\dfrac{\lg_2 \frac{\text{dm}_R(P)}{A}}{1 - \lg_2(1+\mu)} L_1 + L_0 \leq m$ es equivalente a la siguiente desigualdad:

$$\lg_2 \frac{\text{dm}_R(P)}{A} \leq \frac{m - L_0}{L_1}(1 - \lg_2(1+\mu))$$

Llamando $L_2 = \dfrac{m - L_0}{L_1}$, esto implica:

$$\lg_2 \frac{\text{dm}_R(P)}{A} \leq L_2(1 - \lg_2(1+\mu))$$

$$L_2 \lg_2(1+\mu) + \lg_2 \frac{\text{dm}_R(P)}{A} \leq L_2$$

$$\lg_2\left((1+\mu)^{L_2}\frac{\mathrm{dm}_R(P)}{A}\right) \leq L_2$$
$$(1+\mu)^{L_2}\frac{\mathrm{dm}_R(P)}{A} \leq 2^{L_2}$$
$$\left(\frac{1+\mu}{2}\right)^{L_2}\mathrm{dm}_R(P) \leq A$$

La desigualdad del lema 21 en estos términos es

$$\mathrm{dm}_R(C_m\cdots C_1(P)) \leq \left(\frac{1+\mu}{2}\right)^{\frac{m-L_0}{L}}\mathrm{dm}_R(P) = \left(\frac{1+\mu}{2}\right)^{L_2}\mathrm{dm}_R(P),$$

Encadenando con lo anterior, tenemos que $\mathrm{dm}_R(C_m\cdots C_1(P)) \leq A$.

□

3.3.3. PRec cumple los requisitos

Para probar que el tercer intento de Bisec (es decir, el bucle de división que usa CID por el eje menor, figura 3.21) cumple con los requisitos *b*) y *c*) de la tabla 3.1, hemos formulado en las subsecciones anteriores las proposiciones 6 y 7. La primera da el máximo desplazamiento de los cortes que puede alcanzar el bucle (en función del parámetro θ que usa PCR para calcular el índice). La segunda da el decrecimiento en diámetro en función del número de cortes (es decir, la profundidad alcanzada por PRec) y la tolerancia (que es viene de la máxima separación entre cortes).

Ahora usamos estos resultados para justificar el valor

$$Z_\Gamma = \frac{\mathrm{mín}(\mathrm{dm}_H(P), \mathrm{dm}_V(P))}{4(4+2\sqrt{2})(n+2)n}$$

que hemos dado al parámetro θ de PCR en PRec (figura 3.21). Mostraremos que se cumplen los requisitos *c*) (PCR sin error) y *b*) (diámetros decrecientes) de la tabla 3.1 en la siguiente proposición, an los apartados *1*) y *2*) respectivamente. Recordemos que n_P es el número de raíces dentro de una región P.

Proposición 8. *Supongamos que P, de borde Γ, no tiene ninguna raíz a $(4+2\sqrt{2})nZ_\Gamma$ o menos de su borde.*

1) Bisec *da dos regiones, P_0 y P_1, tales que $PCR(P_0, Z_\Gamma)$ y $PCR(P_1, Z_\Gamma)$ retorna exitosamente, con un corte de tolerancia menor o igual que* $\mu = \dfrac{n_P + 1}{2(n+2)}$.

2) las subregiones que surgen de P tras aplicar m veces Bisec *tienen diámetro rectangular menor o igual que A, si*

$$m \geq \frac{\lg_2 \frac{\mathrm{dm}_R(P)}{A}}{1 - \lg_2(1+\mu)} L_1 + L_0$$

con $L_1 = 1 + 2\dfrac{1 - \lg_2(1-\mu)}{1 - \lg_2(1+\mu)}$ *y* $L_0 = 2 + \dfrac{\lg_2(\mathrm{asp}(P))}{1 - \lg_2(1+\mu)}$ *para el valor de μ anterior.*

Demostración. Para *a*), discutimos el caso en que Bisec es un bucle de CID horizontal, siendo similar el caso vertical. Tenemos que $Z_\Gamma = \dfrac{\mathrm{dm}_V(P)}{4(4+2\sqrt{2})(n+2)n} < \dfrac{\mathrm{dm}_V(P)}{(8+4\sqrt{2})(n_P+1)n} = \Theta_H$, el valor de la proposición 6, porque $4(4+2\sqrt{2}) > 8+4\sqrt{2}$ y, como $n \geq n_P$, siendo n_P el número de raíces dentro de P, entonces $n+2 > n_P+1$. Por tanto, por la proposición 6 con $\theta = Z_\Gamma$, $\mathrm{CID}(P,i)$ nos da dos subregiones que no retornan con error en PCR, con separación de la línea central de

$$|h_i| \leq (4+2\sqrt{2})(n_P+1)nZ_\Gamma = (4+2\sqrt{2})(n_P+1)n\frac{\mathrm{dm}_V(P)}{4(4+2\sqrt{2})(n+2)n} = \frac{\mathrm{dm}_V(P)}{2}\frac{n_P+1}{2(n+2)}.$$

Por definición, cualquier corte horizontal con una separación de λ tiene una tolerancia de $\dfrac{2\lambda}{\mathrm{dm}_V(P)}$. Por tanto la tolerancia $\dfrac{2|h_i|}{\mathrm{dm}_V(P)}$ del corte dado por $\mathrm{CID}(P,i)$ is $\dfrac{2}{\mathrm{dm}_V(P)}|h_i| \leq \dfrac{2}{\mathrm{dm}_V(P)}\dfrac{\mathrm{dm}_V(P)}{2}\dfrac{n_P+1}{2(n+2)} = \dfrac{n_P+1}{2(n+2)}$.

Para *b*), notemos que las subregiones producidas por m cortes, horizontales o verticales, son el resultado de una cadena de operadores $C_m C_{m-1} \cdots C_2 C_1(P)$, verificando la hipótesis de la proposición 7 para el valor $\mu = \dfrac{n_P+1}{2(n+2)}$. Por esta proposición concluimos que el diámetro rectangular de las subregiones es menor o igual que A.

□

La proposición anterior muestra que el procedimiento Bisec aplicado recursiva-

mente, con $Z_\Gamma = \dfrac{\mathrm{dm}_V(P)}{4(4+2\sqrt{2})(n+2)n}$ para cortes horizontales y $Z_\Gamma = \dfrac{\mathrm{dm}_H(P)}{4(4+2\sqrt{2})(n+2)n}$ para cortes vertical, produce subregiones que no retornan con error, y cuyo diámetro decrece hasta la precisión requerida.

La cota para m, el número de Bisecs requeridos para alcanzar un diámetro menor o igual que A partiendo de una región P, tiene la forma $KL_1 + L_0$, con un sumando $L_0 = 2 + \dfrac{\lg_2(\mathrm{asp}(P))}{1-\lg_2(1+\mu)}$, logarítmico en el aspecto de P, y un factor $K = \dfrac{\lg_2 \frac{\mathrm{dm}_R(P)}{A}}{1-\lg_2(1+\mu)}$, logarítmico en la razón $\dfrac{\mathrm{dm}_R(P)}{A}$. Estas dependencias logarítmicas eran de esperar, porque el término L_0 cuenta el número de cortes requeridos para decrecer, por división, el aspecto de P, si es muy elongada. La región resultante tras L_0 cortes, de bajo aspecto, es después reducida por división por debajo de la precisión requerida A, con un número de cortes dado por K.

Podemos dar otra cota inferior, de expresión más simple (es decir, sin términos L_0 o L_1) para el número de particiones requeridas:

Corolario. *Las regiones obtenidas tras m aplicaciones de* Bisec *tienen diámetro rectangular menor o igual que A, si*

$$m \geq \frac{6-\lg_2 3}{(2-\lg_2 3)^2} \lg_2 \frac{\mathrm{dm}_R(P)}{A} + \frac{\lg_2(\mathrm{asp}(P))}{2-\lg_2 3} + 2.$$

Demostración. La cota de la proposición anterior $\dfrac{\lg_2 \frac{\mathrm{dm}_R(P)}{A}}{1-\lg_2(1+\mu)} L_1 + L_0$, con $L_1 = 1+2\dfrac{1-\lg_2(1-\mu)}{1-\lg_2(1+\mu)}$ y $L_0 = 2 + \dfrac{\lg_2(\mathrm{asp}(P))}{1-\lg_2(1+\mu)}$, depende de la tolerancia $\mu = \dfrac{n_P+1}{2(n+2)}$. Los términos de esta expresión, $\dfrac{\lg_2 \frac{\mathrm{dm}_R(P)}{A}}{1-\lg_2(1+\mu)}$, L_1 y L_0, son funciones crecientes de μ, luego la expresión que forman es también creciente. Como $\mu = \dfrac{n_P+1}{2(n+2)} < \dfrac{1}{2}$, la cota es menor que el valor tomado por la expresión para $^1/_2$, que es

$$\frac{\lg_2 \frac{\mathrm{dm}_R(P)}{A}}{1-\lg_2(1+1/2)} L' + 2 + \frac{\lg_2(\mathrm{asp}(P))}{1-\lg_2(1+1/2)} =$$
$$= \frac{\lg_2 \frac{\mathrm{dm}_R(P)}{A}}{2-\lg_2 3} L' + 2 + \frac{\lg_2(\mathrm{asp}(P))}{2-\lg_2 3}$$

con $L' = 1 + 2\dfrac{1 - \lg_2(1-1/2)}{1 - \lg_2(1+1/2)} = \dfrac{6 - \lg_2 3}{2 - \lg_2 3}$. Por tanto

$$\frac{\lg_2 \frac{\mathrm{dm}_R(P)}{A}}{1 - \lg_2(1+\mu)} L_1 + L_0 \leq \frac{\lg_2 \frac{\mathrm{dm}_R(P)}{A}}{2 - \lg_2 3} \frac{6 - \lg_2 3}{2 - \lg_2 3} + 2 + \frac{\lg_2(\mathrm{asp}(P))}{2 - \lg_2 3} =$$
$$= \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \frac{\mathrm{dm}_R(P)}{A} + \frac{\lg_2(\mathrm{asp}(P))}{2 - \lg_2 3} + 2$$

□

Numéricamente, $\dfrac{6 - \lg_2 3}{(2 - \lg_2 3)^2} = 25'630\ldots$ y $\dfrac{1}{2 - \lg_2 3} = 2'409\ldots$

3.4. Terminación y coste del procedimiento recursivo

En esta sección demostramos que la salida del procedimiento PRec es una secuencia de aproximaciones a las raíces de f dentro de P, con sus respectivas multiplicidades.

El siguiente teorema asegura que el procedimiento recursivo PRec de la figura 3.21 acaba, y que su salida verifica la propiedad deseada. Para hacer referencia a las llamadas anidadas, recordemos que la primera llamada a PRec tiene profundidad de recursión 0, y las llamadas al mismo procedimiento hechas dentro de una llamada de profundidad de recursión m tienen profundidad de recursión $m + 1$. Recordemos también que la región inicial P_I debe tener un borde Γ_I sin raíces a $(4 + 2\sqrt{2})nZ_{\Gamma_I}$ o menos, para que CID pueda aplicarse sin error[3].

Teorema 7. *Para un polinomio f, si P_I es una región plana con $\mathrm{dm}_R(P_I) > A$ y sin raíces a $(4 + 2\sqrt{2})nZ_{\Gamma_I}$ o menos de su borde, entonces el procedimiento* PRec *aplicado a P_I con una precisión de A verifica que:*

a) Acaba, alcanzando una profundidad de recursión máxima de

$$l_{\text{máx}} = \left\lceil \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \lg_2 \frac{\mathrm{dm}_R(P_I)}{A} + \frac{\lg_2(\mathrm{asp}(P_I))}{2 - \lg_2 3} + 2 \right\rceil .$$

[3]Podemos asegurarnos de que este requisito se cumple si aplicamos $\mathrm{PCR}(\Gamma_I, Z_{\Gamma_I})$ con retorno normal. Si retorna con error, P_I tiene una raíz cerca de su borde y PRec no se puede aplicar con seguridad.

b) Al acabar, las regiones planas de la secuencia $\Pi = (P_1, P_2, \ldots, P_k)$ *son aproximaciones de todas las raíces de* f *en* P_I*, cada una conteniendo el número de raíces indicado por* $N = (n_1, n_2, \ldots, n_k)$.

Demostración. Para *a*), supongamos que se llega a una profundidad de recursión m, es decir se hace $\mathrm{PRec}(C_m C_{m-1} \cdots C_2 C_1(P_I), A)$. La región parámetro surge tras hacer m cortes C_i. Cada corte C_i está hecho por Bisec en una llamada a PRec a profundidad i. Todos estos cortes están hechos por el eje menor, con tolerancia hasta $\frac{1}{2}$ por la proposición 8 *a*). Luego por el corolario 3.3.3, si $m \geq \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \frac{\mathrm{dm}_R(P_I)}{A} + \frac{\lg_2(\mathrm{asp}(P_I))}{2 - \lg_2 3} + 2$ entonces el diámetro rectangular de la subregión $C_m \cdots C_1(P_I)$ es menor o igual que A, retornando por tanto no recursivamente por la salida 2. Luego podemos alcanzar una profundidad de recursión m con $m = \left\lceil \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \lg_2 \frac{\mathrm{dm}_R(P_I)}{A} + \frac{\lg_2(\mathrm{asp}(P_I))}{2 - \lg_2 3} + 2 \right\rceil$, pero no más.

Para mostrar la finitud de cada instancia PRec, en general un procedimiento recursivo que hace un número acotado de llamadas recursivas en cada llamada, y que tiene una profundidad de recursión acotada, acaba en un número finito de pasos. Y este es el caso de PRec, porque como puede haber un máximo de n_P errores de corte, entonces el bucle de división itera como mucho $n_P + 1$ veces. Hay dos llamadas recursivas dentro del bucle, luego cada llamada $\mathrm{PRec}(P, A)$ hace $2(n_P + 1)$ llamadas recursivas o menos. Además hemos mostrado que la máxima profundidad de recursión alcanzable por PRec en P_I es $l_{\text{máx}} = \left\lceil \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \lg_2 \frac{\mathrm{dm}_R(P_I)}{A} + \frac{\lg_2(\mathrm{asp}(P_I))}{2 - \lg_2 3} + 2 \right\rceil$. Luego $\mathrm{PRec}(P_I, A)$ acaba en un número finito de pasos.

Ahora probamos la parte *b*). Decimos que decimos que la región plana P *requiere una* A*-altura de* v si el procedimiento $\mathrm{PRec}(P, A)$ alcanza hasta una profundidad de recursión v, pero no más. La figura 3.27 muestra un ejemplo de árbol de llamadas de PRec. Es decir $\mathrm{PRec}(P_I, A)$ recursivamente llama a $\mathrm{PRec}(P_0, A)$ y $\mathrm{PRec}(P_1, A)$, que a su vez llama a $\mathrm{PRec}(P_{00}, A)$ y $\mathrm{PRec}(P_{01}, A)$, y $\mathrm{PRec}(P_{10}, A)$ y $\mathrm{PRec}(P_{10}, A)$, respectivamente, y así sucesivamente. La región de la llamada a profundidad 0, P_I, tiene A-altura 3, y P_0 y P_1, aunque ambas tienen profundidad 1, muestran A-altura 1 y 2, respectivamente.

Toda región plana P sin raíces a $(4 + 2\sqrt{2})nZ_\Gamma$ o menos de su borde Γ tiene A-profundidad finita, menor o igual que $\left\lceil \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \lg_2 \frac{\mathrm{dm}_R(P)}{A} + \frac{\lg_2(\mathrm{asp}(P))}{2 - \lg_2 3} + 2 \right\rceil$

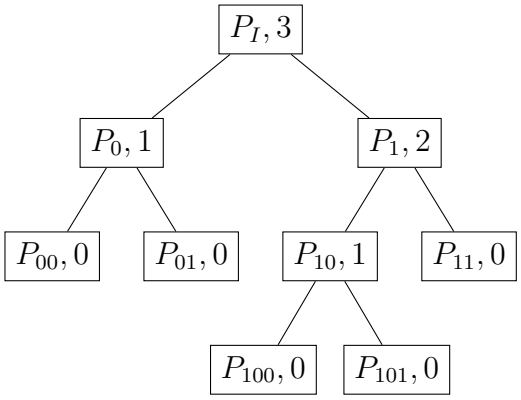


Figura 3.27: Árbol de llamadas de PRec, empezando por PRec(P_I, A). Cada nodo está etiquetado con el nombre de la región y su A-altura.

como hemos visto en *a*), porque el razonamiento hecho para P_I es también válido para una región general P. Demostraremos la parte *b*) por inducción en la A-altura v. Es decir, primero demostraremos *b*) para regiones P de A-altura 0 (el caso base), y luego para regiones de mayor A-altura, suponiendo cierta *b*) para alturas menores. Así, la hipótesis de inducción, que denotamos HI(v), es "Al final del procedimiento PRec(P, A) (para cualquier región plana P sin raíces a $(4 + 2\sqrt{2})nZ_\Gamma$ o menos de su borde Γ, de A-altura v) las regiones de Π tienen diámetro rectangular menor que A, contienen el número de raíces dado por N, y esas son todas las raíces de f contenidas en P". Demostraremos primero el caso base HI(0), y luego el caso general: si HI(v') para cada $v' < v$, entonces HI(v).

Notemos que esta inducción es completa, lo que significa que el caso v-ésimo depende de todos los casos previos, no solo del inmediatamente precedente. Es un ejemplo de inducción estructural [Burstall, 1969, Hopcroft et al., 2000], un método general para probar propiedades de procedimientos recursivos.

El caso base ($v = 0$) consiste en las regiones que no requieren llamadas recursivas, las que no tienen raíces (que retornan por la salida 1), o con una o más raíces pero con diámetro menor que A (que retornan por la salida 2). Retornando por la salida 1 la aserción HI(0) se verifica trivialmente porque P no tiene raíces y Π permanece vacío (como N). Retornando por la salida 2 la aserción se verifica también porque Π contiene solo la región P, y como PCR calcula sin excepción el índice (ya que por la hipótesis HI(0), no hay raíces cerca del borde de P), N contiene el número de raíces dentro de P.

Para el caso general ($v \geq 1$), consideramos las dos llamadas recursivas realizadas, PRec(P_i, A), siendo P_0 y P_1 regiones disjuntas que recubren P, producidas por alguna iteración de CID, que retornan exitosamente en PCR. P tiene A-altura v, es decir, PRec(P, A) alcanza una profundidad de recursión de v. Como PRec(P_0, A) y Prec(P_1, A) están un nivel de recursión por debajo de PRec(P, A), las A-alturas de P_0 y P_1, digamos v_0 y v_1, no son necesariamente iguales, pero ambas son menores o iguales que $v - 1$. Por hipótesis de inducción HI(v_i) (que suponemos probada porque $v_i < v$) al retornar de la llamada recursiva PRec(P_i, A) las regiones pertenecientes a Π_i son todas de diámetro menor que A, contienen el número de raíces indicado por N_i, y esas son todas las raíces en P_i. Por tanto la concatenación de Π_1 y Π_2 (es decir, Π) junto con la de N_1 y N_2 (que es N) verifica que sus regiones tienen diámetros y número de raíces como se ha especificado. Que esas son todas las raíces dentro de P viene del hecho de que las P_i son disjuntas y recubren P. Esto es lo que se quería demostrar, HI(v).

□

Para el coste computacional del método completo para hallar raíces, vamos a examinar las tareas ejecutadas por PRec(P, A) (figura 3.21), que son PCR(P), el cálculo del diámetro $\mathrm{dm}_R(P)$ y la división en subregiones P_0 y P_1, Bisec(P), hecha por una serie de CID . Discutiremos estas tareas, contabilizando sus operaciones, para ver que su coste se debe principalmente al cálculo de $f(z)$ para z complejo (lo que se conoce como una evaluación polinómica, EP). Concluiremos que el coste de PRec está bien cuantificado por el número de EP que requiere.

El borde Γ se construye concatenando segmentos de recta, uniformemente parametrizados, y el muestreo $\Gamma(S_P)$ incluye los extremos de estos segmentos (ver la figura 3.28). Si la región inicial se construye de esta forma, las subregiones producidas por CID también son de esta forma. Con los bordes definidos de esta manera se determinará el coste. Recordemos que se trabaja con una secuencia de valores del parámetro $S_P = (s_0, \ldots, s_n)$, muestreo de $[a, b]$. De esta manera $\Gamma(S_P)$ es una secuencia de puntos del borde de P. Recordemos también que la región inicial y las subregiones son convexas, como se comenta en el párrafo previo a la sección 3.2.1.

Dejando a un lado de momento Ind, examinamos primero el cálculo del diámetro $\mathrm{dm}_R(P) = \sqrt{\mathrm{dm}_H(P)^2 + \mathrm{dm}_V(P)^2}$. Llamemos $minR$ y $maxR$ al máximo y el

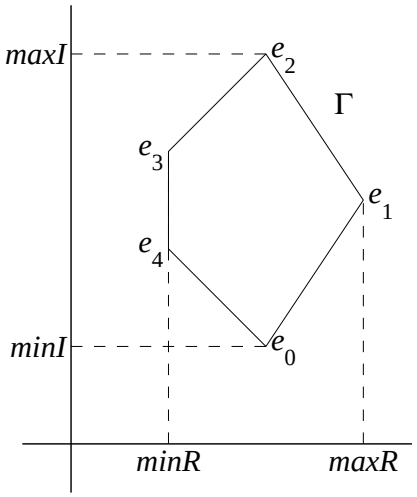


Figura 3.28: La curva Γ es la concatenación de los segmentos de recta $\overline{e_i, e_{i+1}}$ uniformemente parametrizados.

mínimo, respectivamente, de la parte real de Γ, el borde de P. Asimismo, $minI$ y $maxI$ son el máximo y el mínimo de la parte imaginaria. el diámetro horizontal es $maxR - minR$, y el vertical $maxI - minI$. Los valores extremos $minR$, $maxR$, $minI$ y $maxI$ se alcanzaran en puntos que son un extremo e_i de los segmentos que componen Γ, y estos extremos pertenecen a la secuencia $\Gamma(S_P)$ (figura 3.28). Por tanto podemos encontrar los valores extremos escaneando esta secuencia, haciendo cuatro comparaciones en cada punto. Notemos que los puntos de $\Gamma(S_P)$ se calculan dentro del bucle de inserción de PCR, por lo que el coste de evaluar Γ será contabilizado en al apartado correspondiente más adelante. Luego el coste del cálculo del diámetro es el de el escaneo de la secuencia con comparaciones (más dos sustracciones, dos cuadrados, una suma y una raíz cuadrada), por región P.

La división en subregiones Bisec(P) se realiza mediante CID (figura 3.19). Este calcula la línea media m_H o m_V (y sus desplazamientos, si hay alguno). Consideremos el caso de división horizontal, siendo similar la vertical. Usando la media $mI = \dfrac{minI + maxI}{2}$, escaneamos $\Gamma(S_P)$ separando los puntos con parte imaginaria mayor o igual que mI (con los que componemos el borde de $T(P)$) de aquellos cuya parte imaginaria es menor o igual que mI (para hacer $B(P)$). Como mucho dos puntos de Γ tienen parte imaginaria exactamente mI, porque la línea m_H corta a Γ en dos puntos, por la convexidad de P. Es decir $m_H \cap \Gamma = \{c_1, c_2\}$.

Cada c_k, $k = 1, 2$ se calcula a partir de un par de puntos consecutivos de $\Gamma(S_P)$, uno de los cuales está por encima de mI y el otro por debajo, $\Gamma(s_i)$ y $\Gamma(s_{i+1})$ en la figura 3.29. Si estos puntos de corte no pertenecen todavía a $\Gamma(S_P)$, se insertan en las secuencias correspondientes a los bordes de $T(P)$ y $B(P)$, de modo que estas subregiones encajen exactamente en P. Por interpolación lineal, en el caso peor cada coordenada de c_i cuesta cuatro adiciones (o sustracciones) en coma flotante, mas un producto y una división.

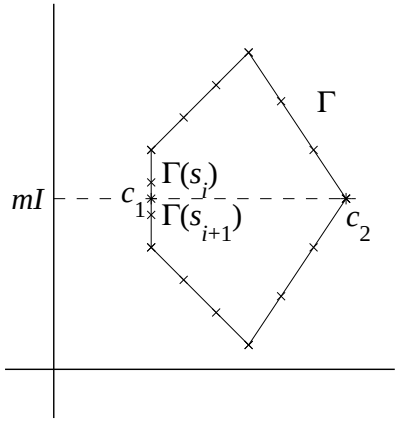


Figura 3.29: La curva Γ tiene dos puntos a altura mI. Las coordenadas de c_1 se encuentran por interpolación lineal.

Por tanto, el primer corte de una división horizontal por CID requiere un escaneo de $\Gamma(S_P)$, comparando la parte imaginaria con mI, y también dieciséis adiciones (o sustracciones) en coma flotante y cuatro divisiones para las coordenadas de c_1 y c_2. Una división vertical es similar. Cada corte desplazado posterior, si hay alguno, requiere un coste igual a esta cantidad. Recordemos que el número de desplazamientos está acotado por $n_P + 1$, siendo n_P el número de raíces dentro de P. En total, Bisec(P) requiere menos de 16 adiciones más 4 multiplicaciones, todo esto multiplicado por $n_P + 1$.

Sobre la tarea PCR, su primera parte es PCR propiamente dicho (figura 3.1), que produce la secuencia S_P de valores del parámetro cuyos puntos correspondientes $f(\Gamma(S_P))$ están conectados, y la segunda parte es un escaneo de $f(\Gamma(S_P))$, para contar el número de cruces por el eje positivo de las abscisas, como se describe en

la introducción. Las subtareas de PCR, en cada iteración del bucle, son la inserción de un valor del parámetro $s = \frac{s_i + s_{i+1}}{2}$ en la secuencia S_P, y la evaluación de los predicados p, q_2 y r. La evaluación de p solo requiere comparaciones, mientras que la de r requiere una sustracción en coma flotante $s_{i+1} - s_i$.

El predicado q:

$$|f(\Upsilon(s_i))| + |f(\Upsilon(s_{i+1}))| \leq 2\,|f'(\Upsilon(s_i))|\,(s_{i+1} - s_i) + |f(\Upsilon(s_{i+1})) - f(\Upsilon(s_i))|$$

es más complicado, requiriendo EP de f y de f' en $\Gamma(s_i)$, una sustracción compleja, dos sumas en coma flotante, dos multiplicaciones y tres cálculos del valor absoluto. La sustracción $s_{i+1} - s_i$ ya se ha contabilizado en el predicado r. Además la EP y el valor absoluto $|f(\Upsilon(s_{i+1}))|$ se contabiliza solo una vez para la evaluación de tanto $q(s_i)$ como $q(s_{i+1})$. Una adición (o sustracción) compleja es equivalente a dos adiciones en coma flotante [Goldberg, 1991b]. Un valor absoluto complejo es equivalente a una suma, dos multiplicaciones y una raíz cuadrada. Luego q requiere siete adiciones, ocho multiplicaciones, tres raíces cuadradas u la EP de f y de f' en $\Gamma(s_i)$. Por tanto, para cada iteración del bucle de PCR gastamos nueve adiciones (o sustracciones), ocho multiplicaciones, una división, tres raíces cuadradas y dos EP.

El número de iteraciones del bucle coincide con el número de valores del parámetro insertados en S_P. Esta cantidad es difícil de estimar con precisión, pero es igual al número de puntos de $f(\Gamma(S_P))$, para el que se puede dar una cota inferior: para que los puntos de $f(\Gamma(S_P))$ estén en sectores conexos, esta secuencia debe tener al menos ocho puntos por cada vuelta de $f(\Gamma)$ alrededor del origen (es decir, por cada raíz de Γ dentro de P). Este es el mínimo número de puntos que se deben poner en sectores consecutivos para cruzar el eje de las abscisas el número adecuado de veces (ver la figura 2.5). Probablemente PCR insertará más puntos que estos para acabar teniendo uno en cada sector. En total PCR requiere al menos $8n_P$ veces una inserción (cada una equivalente a nueve adiciones, ocho multiplicaciones, una división, tres raíces cuadradas y dos EP).

Resumimos en la tabla 3.2 el coste de PRec en cada región que aparece en la descomposición recursiva. Para tener un indicio del coste de estas operaciones cuando se realizan en coma flotante, se consulta la latencia en diferentes procesadores [Fog, 2014]. La latencia es el tiempo que tarda una operación en obtener

el resultado. Las latencias de la división y la raíz cuadrada son similares, entre cinco y diez veces la de la multiplicación. Llamando A al número de cortes que hace CID, y B al número de puntos insertado por PCR, se tiene $A \leq n_P + 1$ y $B \geq 8n_P$. Operando, el coste total sin EP es de $16A + 9B$ en adiciones, $2B$ en multiplicaciones, y $4A + 2B$ en operaciones superiores (división y raíz cuadrada). Comparemos esto con el coste en EP.

	Adiciones	Multiplicaciones	Divisiones	Raíz cuadrada	EP	Veces realizada por llamada
Bisec	16		4			menos de $n_P + 1$ veces
PCR	9	8	1	3	2	más de $8n_P$ veces

Tabla 3.2: Operaciones realizadas por PRec en una región P.

En general, un polinomio de grado n se puede evaluar en un punto complejo con $2\left\lfloor \frac{n+1}{2} \right\rfloor$ adiciones complejas y $\left\lfloor \frac{n+1}{2} \right\rfloor$ multiplicaciones complejas usando un esquema de evaluación de Horner complejo [Knuth, 1981]. Una adición compleja es equivalente a dos adiciones en coma flotante. Una multiplicación compleja es equivalente a cuatro multiplicaciones en coma flotante más dos adiciones en coma flotante. Sumando se tiene que una EP requiere $2 \cdot 2\left\lfloor \frac{n+1}{2} \right\rfloor + 2\left\lfloor \frac{n+1}{2} \right\rfloor = 6\left\lfloor \frac{n+1}{2} \right\rfloor$ (aproximadamente $3(n+1)$) adiciones en coma flotante y $4\left\lfloor \frac{n+1}{2} \right\rfloor$ (aproximadamente $2(n+1)$) multiplicaciones en coma flotante. El coste debido solo a EP es $2 \cdot 3(n+1)B$ en adiciones y $2 \cdot 2(n+1)B$ en multiplicaciones. Para comparar, asignamos un coste de 5 multiplicaciones a las operaciones superiores. El coste sin EP es $16A + 9B$ en adiciones, y $2B + 5(4A + 4B)$ en multiplicaciones, contra $6(n+1)B$ adiciones y $4(n+1)B$ multiplicaciones de EP solamente. Notemos que $A < B$ si hay alguna raíz dentro de P, luego si $n > 5$ el coste en adiciones solo de las EP es ya mayor. El coste en multiplicaciones solo de las EP es mayor que la parte sin EP para $n > 10$, incluso en el caso $n_P = 1$. Además esta diferencia en la contribución al coste entre EP y las otras operaciones se hace más acentuada para valores mayores de n, porque este último coste no se ve afectado por el grado.

Por tanto consideramos que el coste de PRec está bien medido por el número de EP que requiere, despreciando el resto de las operaciones. Acotamos este número en el siguiente teorema. Debe notarse que es una cota en el caso peor, y que es extremadamente pesimista. El razonamiento supone que, en todos los casos, se hacen todos los posibles cortes desplazados, y también que cada raíz tiene una rama del árbol de búsqueda exclusivamente dedicada. Estos supuestos son sumamente onerosos, y cada uno añade un factor de n_P a la cota. Es plausible que un razonamiento más complicado decrezca el coste hasta un orden $O(n_P n \log_2(1/A))$.

A la región inicial se aplica PCR, y luego es dividida aplicando CID en el bucle de división, en cada ocasión con una llamada a PCR sobre cada una de las dos regiones producidas, hasta que estas llamadas retornen exitosamente. Las subregiones a su vez se dividen con un bucle de CID de la misma manera. Por tanto el coste total es la suma de las EP requeridas por el bucle de división en la región inicial, más las EP en las subregiones, cada iteración con dos llamadas a PCR.

Primero enunciamos la proposición 9 sobre el coste del bucle de división, es decir, el coste de varias iteraciones de CID (con su par de PCR en cada iteración) necesarias para dividir con éxito en dos subregiones. Después, en la proposición 10, calculamos el coste de PRec, suponiendo que los errores que ocurren son solo de tipo corte. Este tipo de error solo requiere desplazar el corte, descartando el trabajo hecho por PCR en como mucho dos regiones. En cambio un error de tipo diferido conlleva descartar un subárbol del árbol de búsqueda, porque tal error es retornado una y otra vez hasta subir a una profundidad de recursión en la que es gestionado como error de corte. Finalmente en el teorema 8 calculamos el coste en una ejecución con errores de cualquier tipo.

Proposición 9. *Si P no tiene ninguna raíz a $(4 + 2\sqrt{2})nZ_\Gamma$ o menos de su borde, el número de EP requeridas por el bucle de división es menor o igual que*

$$16(16 + 8\sqrt{2})(n_P + 1)(n^2 + 2n).$$

Demostración. Se realizan dos EP por cada punto insertado por PCR. Acotaremos el número de puntos insertados. Suponemos primeramente que no se hace ningún desplazamiento en el bucle, es decir, el corte inicial CID $(P, 0, Z_\Gamma)$ a lo largo de la línea media da subregiones exitosas en PCR. Luego tratamos el caso general

general, con desplazamientos CID(P, i, Z_Γ) con $i > 0$.

Si no hay error, solo tenemos que acotar el número de EP requeridas por PCR in $T(P)$ y $B(P)$ (para un corte horizontal, el vertical es similar). Consideramos primero $T(P)$, de borde Γ_T. Los valores paramétricos de los puntos insertados deben estar a una distancia mayor que Z_{Γ_T}, porque en caso contrario PCR (figura 3.1) retorna con error. $T(P)$ está contenida en su HV-envolvente, $\mathrm{Env}_{HV}(T(P))$ (el rectángulo delimitado por las líneas de soporte horizontal y vertical de $T(P)$). Como $T(P)$ y $\mathrm{Env}_{HV}(T(P))$ son ambas regiones convexas, la primera contenida en la segunda, el perímetro de $T(P)$ es menor o igual que el perímetro de $\mathrm{Env}_{HV}(T(P))$ (porque tomar el perímetro es un operador creciente en regiones convexas, véase por ejemplo [Lay, 2007]). El perímetro de $\mathrm{Env}_{HV}(T(P))$ es $2(\mathrm{dm}_H(T(P)) + \mathrm{dm}_V(T(P)))$, luego $\mathrm{longarc}(\Gamma_T) \leq 2(\mathrm{dm}_H(T(P)) + \mathrm{dm}_V(T(P)))$. Por la parametrización uniforme, el intervalo de parámetros de Γ_T también tiene de longitud el perímetro de Γ_T. En un intervalo de longitud menor o igual que $2(\mathrm{dm}_H(T(P)) + \mathrm{dm}_V(T(P)))$, como es el intervalo de parámetros de Γ_T, podemos insertar m puntos a una distancia mayor que Z_{Γ_T} solo si $(m+1)Z_{\Gamma_T} \leq 2(\mathrm{dm}_H(T(P)) + \mathrm{dm}_V(T(P))$. Luego el número de puntos insertados es menor o igual que

$$m \leq \frac{2(\mathrm{dm}_H(T(P)) + \mathrm{dm}_V(T(P))}{Z_{\Gamma_T}} - 1 < \frac{2(\mathrm{dm}_H(T(P)) + \mathrm{dm}_V(T(P))}{Z_{\Gamma_T}} =$$
$$= (16 + 8\sqrt{2})(n^2 + 2n)\frac{2(\mathrm{dm}_H(T(P)) + \mathrm{dm}_V(T(P))}{\mathrm{dm}_V(T(P))} \leq 4(16 + 8\sqrt{2})(n^2 + 2n).$$

La última desigualdad se debe a que $\dfrac{\mathrm{dm}_H(P)}{\mathrm{dm}_V(P)} \leq 1$, lo que es el caso porque estamos considerando CID de tipo horizontal, es decir, el corte a lo largo del eje menor es horizontal. El número de puntos insertados en Γ_B verifica una cota similar.

En total, el número de puntos insertados por CID es menor de $8(16+8\sqrt{2})(n^2 + 2n)$, si no hay desplazamiento en el bucle de división. En caso contrario, las subregiones que CID produce son de la forma $T_\lambda(P)$ y $B_\lambda(P)$ con $\lambda \neq 0$, y cada uno de estos desplazamientos requiere descartar un par de tales regiones y considerar un nuevo par, con sus respectivos puntos de inserción. En total puede haber n_P desplazamientos, luego $n_P + 1$ pares de regiones a considerar. En cualquier caso el número de puntos insertados requeridos por todos los desplazamientos de un bucle CID

horizontal para hacer dos subregiones exitosas en PCR es menor o igual que

$$16(16+8\sqrt{2})(n_P+1)(n^2+2n).$$

en el caso de un bucle CID vertical, el razonamiento es similar, intercambiando dm_H y dm_V.

□

La anterior proposición nos da el coste del bucle de división, y es un paso clave en el cálculo del coste de PRec (teorema 8). Llamando $K = 16(16+8\sqrt{2})(n^2+2n)$, el coste anterior, para la división de una región P conteniendo n_P raíces, es de menos de $K(n_P+1)$. Como veremos, esta cota es la causa en última instancia de que PRec tenga una cota de coste de orden $O(n^2 n_P)$.

Como se ha comentado, contabilizaremos el número de EP de PRec(P_I, A) sumando aquellas requeridas por el bucle de división en cada una de las subregiones, aplicando repetidamente la proposición 9. Nos referimos a las subregiones que surgen en PRec (P_I, A) Con doble subíndice $P_{i,j}$ siendo i la profundidad de la región y j su índice entre las 2^i subregiones de esa profundidad (figure 3.30). El borde de $P_{i,j}$ se denotará $\Gamma_{i,j}$. Empezando con $P_{0,0} = P_I$, cada región $P_{i,j}$ da dos regiones hijas $P_{i+1,2j}$ y $P_{i+1,2j+1}$ despues de un corte, de tipo horizontal o vertical según el eje menor de $P_{i,j}$. Podríamos haber precisado más, diciendo que $P_{i+1,2j} = T_\lambda(P_{i,j})$ y $P_{i+1,2j+1} = B_\lambda(P_{i,j})$ si $\mathrm{dm}_V(P_{i,j}) \geq \mathrm{dm}_H(P_{i,j})$ (y un convenio similar con los operadores de corte L_λ y R_λ), pero sería indiferente para el siguiente razonamiento.

Denotamos con $n_{i,j}$ el número de raíces dentro de $P_{i,j}$, con lo que $n_{i,j} = n_{i+1,2j} + n_{i+1,2j+1}$. Por la proposición 9, si no hay raíces a $(4+2\sqrt{2})nZ_{\Gamma_{i,j}}$ o menos de $\Gamma_{i,j}$, el costo $C(P_{i,j})$ en cada nodo del árbol de la figura 3.30 es $C(P_{i,j}) \leq K(n_{i,j}+1)$ si $n_{i,j} \neq 0$. Si por el contrario $n_{i,j} = 0$, entonces $C(P_{i,j}) = 0$ porque en tal caso PRec no gasta EP en seguir procesando $P_{i,j}$. Definiendo $\overline{C}(P_{i,j}) = K(n_{i,j}+1)$ si $n_{i,j} \neq 0$, $\overline{C}(P_{i,j}) = 0$ si $n_{i,j} = 0$, tenemos $C(P_{i,j}) \leq \overline{C}(P_{i,j})$. Recordemos que $C(P_{i,j})$ es el coste suponiendo que no hay raíces cerca del borde de $P_{i,j}$, lo que implica que solo hay errores de corte.

El siguiente lema relaciona el valor de la cota $\overline{C}(P_{i,j})$ en una región con su valor en las subregiones hijas.

Lema 22. *Si P es una región de profundidad i, y P_0, P_1 son sus regiones hijas,*

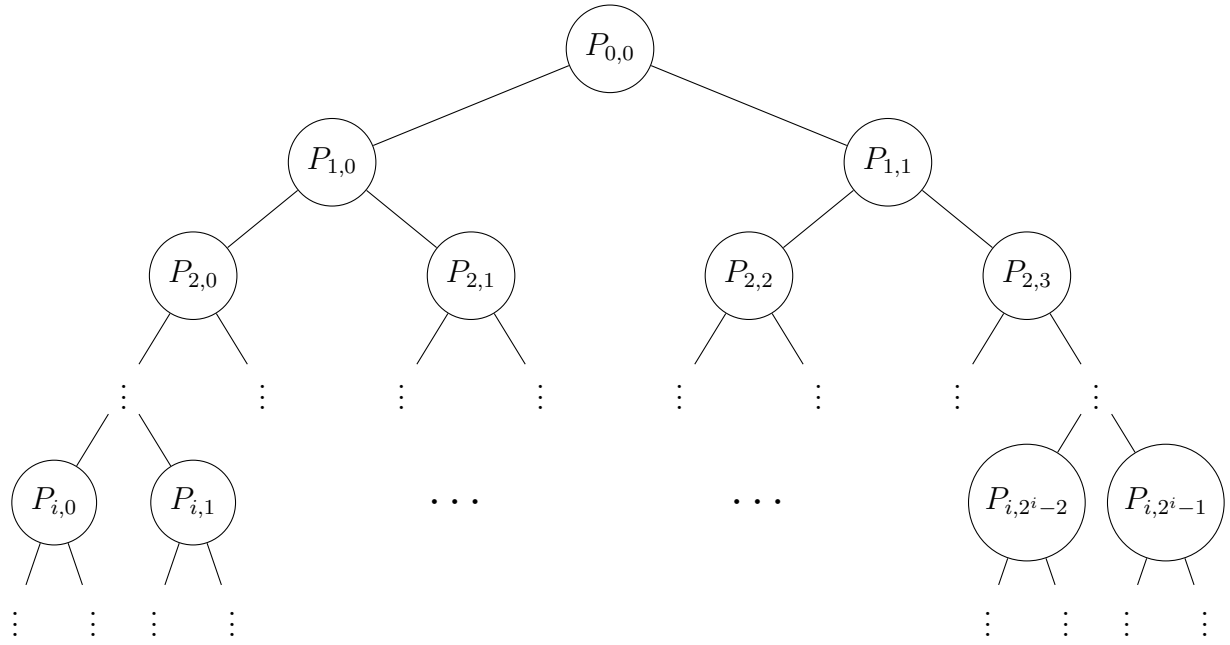


Figura 3.30: Cada región $P_{i,j}$ se escinde en dos subregiones hijas $P_{i+1,2j}$ y $P_{i+1,2j+1}$.

con n, n_0, n_1 raíces respectivamente, $n_0 \neq 0 \neq n_1$, entonces

$$\overline{C}(P_0) + \overline{C}(P_1) = \overline{C}(P) + K$$

Demostración. Notemos que las raíces están particionadas, $n_0 + n_1 = n$. Luego:

$$\begin{aligned} \overline{C}(P_0) + \overline{C}(P_1) &= K(n_0 + 1) + K(n_1 + 1) = \\ &= K(n_0 + n_1 + 1) + K = K(n + 1) + K = \overline{C}(P) + K. \end{aligned}$$

□

Si n_0 o n_1 es cero, entonces $\overline{C}(P_0) + \overline{C}(P_1) = \overline{C}(P)$. Podemos decir que una descomposición balanceada (es decir, con n_0 y n_1 no nulos) incrementa el coste de un nivel al siguiente, pero que una descomposición no balanceada (con cero raíces en alguna subregión) lo mantiene igual.

Extendemos el lema 22, sobre incremento de la cota de coste por cada nodo, a uno sobre incremento de la cota de coste por profundidad. Denotamos el coste

total de las regiones a profundidad i como $C_D(i) = \sum_{j=0}^{2^i-1} C(P_{i,j})$, y el coste total de división como $C_T = \sum_{i=0}^{l_{\text{máx}}} C_D(i)$, siendo $l_{\text{máx}}$ la profundidad máxima de recursión, que por el teorema 7 *a*) está acotada por $\left\lceil \frac{6-\lg_2 3}{(2-\lg_2 3)^2} \lg_2 \frac{\text{dm}_R(P_I)}{A} + \frac{\lg_2(\text{asp}(P_I))}{2-\lg_2 3} + 2 \right\rceil$. De modo similar, llamamos $\overline{C_D}(i) = \sum_{j=0}^{2^i-1} \overline{C}(P_{i,j})$ a la suma de las cotas de coste de las regiones a profundidad i. Recordemos que n_I es el número de raíces dentro de la región inicial P_I.

Lema 23.

$$\overline{C_D}(i) \leq \overline{C_D}(i-1) + \frac{n_I}{2}K$$

Demostración. Decimos que $P_{i,j}$ es una *región activa* si $n_{i,j} \neq 0$. Entre las regiones no activas, diferenciamos *regiones nulas* (si su región padre es una región activa) y *regiones descartadas* (si su región padre es una región no activa, véase la figure 3.31).

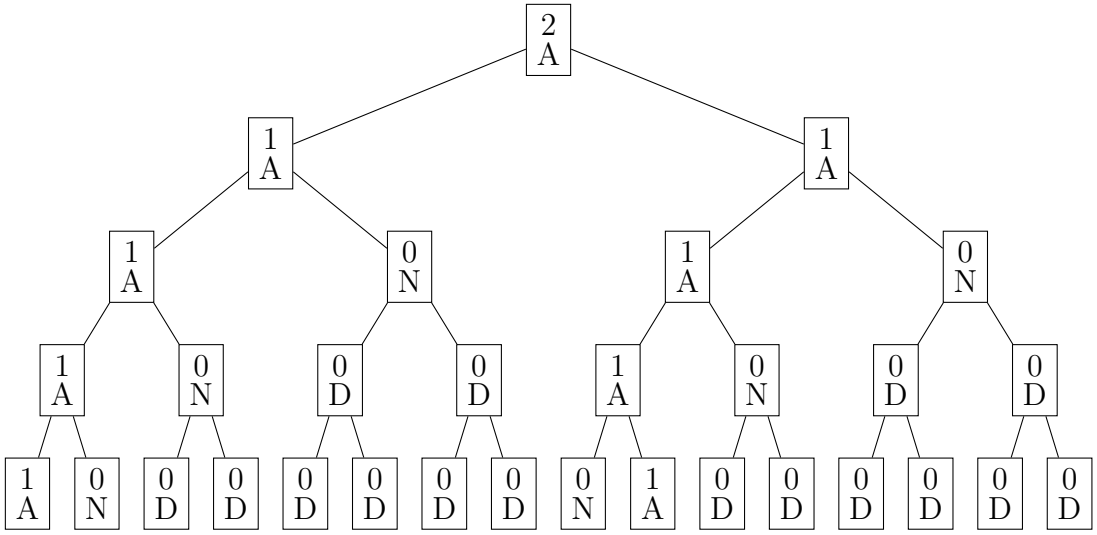


Figura 3.31: La etiqueta de cada región tiene el número de raíces que contiene, y además si es activa (A), nula (N) o descartada (D).

Esta clasificación interesa porque las regiones descartadas no tienen coste en PRec, ni siquiera son construidas porque la descomposición en subregiones se detiene en regiones nulas (figura 3.6). Las regiones nulas son construidas, y PCR

inserta puntos en su borde, pero no requieren ulteriores EP. Y las regiones activas no decrecen el coste de un nivel al siguiente, como afirma el lema 22. Notemos que puede haber como mucho n_I regiones activas a una profundidad dada, y cada una de estas puede ser el padre de como mucho una región nula. Por tanto el número de regiones nulas en cada profundidad es menor o igual que n_I. Ciertamente, en los primeros niveles de profundidad, el número de regiones activas (y nulas) es incluso menor, pero tomamos la cota n_I por uniformidad.

Aplicaremos el lema 22 a cada uno de los 2^{i-1} pares de regiones de profundidad i, compuestos por $P_{i,2j}$ y $P_{i,2j+1}$, hijas de $P_{i,j}$. Para cada par, o los dos miembros son descartados, o al menos un miembro es activo, porque si ambos son nulos, entonces su padre $P_{i,j}$ sería nulo también y por tanto sus hijas serían descartadas, en contradicción con la hipótesis de que son ambas nulas. Si $A(i)$ es el número de regiones activas a profundidad i, como $A(i) \leq n_I$, el número de pares de regiones ambas activas es menor o igual que $\left\lfloor \frac{n_I}{2} \right\rfloor \leq \frac{n_I}{2}$. Cada uno de estos pares incrementa en K la cota de coste con respecto al padre, por el lema 22. Por tanto el coste a profundidad i está acotado por

$$\begin{aligned}
\overline{C_D}(i) = \sum_{j=0}^{2^i-1} \overline{C}(P_{i,j}) = \\
= \overline{C}(P_{i,0}) + \overline{C}(P_{i,1}) + \cdots + \overline{C}(P_{i,2^i-2}) + \overline{C}(P_{i,2^i-1}) \leq \\
\leq \overline{C}(P_{i-1,0}) + \cdots + \overline{C}(P_{i-1,2^{i-1}-1}) + \frac{A(i)}{2} K \leq \\
\leq \overline{C_D}(i-1) + \frac{n_I}{2} K
\end{aligned}$$

como queríamos demostrar.

□

Con estas proposiciones, podemos mostrar la siguiente afirmación sobre el coste de PRed, suponiendo que no haya error diferido. Llamamos $l_{max}(P)$ a la máxima profundidad que se alcanza por el árbol de búsqueda que empieza en P. Por el teorema 7 *a*) está acotada por $\left\lceil \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \lg_2 \frac{\mathrm{dm}_R(P)}{A} + \frac{\lg_2(\mathrm{asp}(P))}{2 - \lg_2 3} + 2 \right\rceil$.

Proposición 10. *Si P no tiene alguna raíz a $(4 + 2\sqrt{2})nZ_\Gamma$ o menos de su borde, el número $C_T(P)$ de EP realizadas por* $\mathrm{RDP}(P, A)$*, si cada error es de tipo corte,*

es menor o igual que

$$C_T(P) \leq 16(16+8\sqrt{2})(n^2+2n)l_{\text{máx}}(P)\left(n_P\frac{l_{\text{máx}}(P)+3}{4}+1\right)$$

siendo $l_{max}(P) \leq \left\lceil \frac{6-\lg_2 3}{(2-\lg_2 3)^2}\lg_2\frac{\text{dm}_R(P)}{A}+\frac{\lg_2(\text{asp}(P))}{2-\lg_2 3}+2\right\rceil$, *y* n_P *el número de raíces dentro de* P.

Demostración. Para obtener la cota anterior, sumamos el número de EP requeridas por la división recursiva de las subregiones, dada por el lema 23, hasta una profundidad máxima de $l_{max}(P)$, por el teorema 7 *a*). Tenemos la siguiente cota del coste de un nivel de profundidad tras la aplicación reiterada del lema 23.

$$\overline{C_D}(i) \leq \overline{C_D}(i-1)+\frac{n_I}{2}K \leq \overline{C_D}(i-2)+2\frac{n_I}{2}K \leq \cdots$$
$$\cdots \leq \overline{C_D}(1)+(i-1)\frac{n_I}{2}K \leq \overline{C_D}(0)+i\frac{n_I}{2}K$$

Sumando este coste a lo largo de todos los niveles de profundidad, el coste total C_T de división está acotado por $\sum_{i=0}^{l_{\text{máx}}(P)-1}\overline{C_D}(i)$, porque $l_{\text{máx}}(P)-1$ es la mayor profundidad a la que se hacen divisiones. Es decir:

$$\sum_{i=0}^{l_{\text{máx}}(P)-1}\overline{C_D}(i) = \overline{C_D}(0)+\sum_{i=1}^{l_{\text{máx}}(P)-1}\overline{C_D}(i) \leq$$
$$\leq \overline{C_D}(0)+\sum_{i=1}^{l_{\text{máx}}(P)-1}\left(\overline{C_D}(0)+i\frac{n_I}{2}K\right) =$$
$$= l_{\text{máx}}(P)\overline{C_D}(0)+\frac{n_I}{2}K\sum_{i=1}^{l_{\text{máx}}(P)-1} i =$$
$$= l_{\text{máx}}(P)\overline{C_D}(0)+\frac{n_I}{2}K\frac{(l_{\text{máx}}(P)-1)l_{\text{máx}}(P)}{2}$$

Tenemos que $\overline{C_D}(0) = \overline{C}(P_I) = K(n_I+1)$, y reemplazando el valor de K:

$$l_{\text{máx}}(P)K(n_I+1)+\frac{n_I}{2}K\frac{(l_{\text{máx}}(P)-1)l_{\text{máx}}(P)}{2} =$$
$$= Kl_{\text{máx}}(P)\left(n_I+1+\frac{n_I}{2}\frac{l_{\text{máx}}(P)-1}{2}\right) =$$

$$= 16(16 + 8\sqrt{2})(n^2 + 2n)l_{\text{máx}}(P)\left(n_I\frac{l_{\text{máx}}(P)+3}{4}+1\right)$$

□

Después de la anterior proposición, con una restrictiva hipótesis sobre el tpo de error, consideramos ahora que los errores pueden ser de cualquier tipo, de corte o diferidos. En un análisis de caso peor, supondremos se comete todo posible error de corte, es decir que cada bucle de división realiza n_P iteraciones, como hemos supuesto antes. Ahora además suponemos que cada error de corte se retorna como error diferido que viene de la mayor profundidad posible. Así no solo contabilizamos el descarte de un corte y su desplazamiento, sino también el descarte del subárbol que cuelga de ese corte. Suponemos también que este subárbol es tan grande como sea posible. Aunque estas son unas suposiciones muy pesimistas no incrementan el orden del coste, porque la profundidad $l_{max}(P)$ del subárbol que cuelga de P no depende de n.

Llamamos $F(P)$ al coste total de PRec en P considerando los dos tipos de error. Cada error diferido está causado por al menos una raíz, que está cerca del borde de una subregión. Notemos que cada raíz puede causar como mucho un error diferido en cada bucle de división, porque en la gestión de este error se inserta un nuevo segmento de corte que no está cerca de esta raíz, por lo que no puede causar un nuevo error diferido. Por tanto $F(P)$ es menor o igual que n_P veces el coste $C(P)$ de PRec sin error diferido: $F(P) \leq n_P C(P)$. Usando la proposición 10.

$$F(P) \leq n_P K' l_{\text{máx}}(P)\left(n_P\frac{l_{\text{máx}}(P)+3}{4}+1\right) = K' g(l_{\text{máx}}(P), n_P)$$

siendo $K' = 16(16 + 8\sqrt{2})(n^2 + 2n)$, y $g(x, y) = y^2\dfrac{x(x+3)}{4} + yx$.

Como las regiones que pueblan el árbol de búsqueda cambian con el descarte de subárboles producido por la gestión de errores diferidos, consideramos el coste acumulado $F(i, j)$ de todas las regiones que aparecen en el lugar (i, j) del árbol de búsqueda. Es decir $F(i,j) = \sum_{k=1}^{k_m} F(P_{i,j}^{(k)})$, siendo $P_{i,j}^{(1)}, P_{i,j}^{(2)}, \ldots, P_{i,j}^{(k_m)}$ las regiones que han ocupado ese lugar del árbol. Esto significa que $P_{i,j}^{(k)}$ es la región que ocupa el lugar (i, j) después de descartar el subárbol que contenía $P_{i,j}^{(k-1)}$. Llamamos $n_{i,j}^{(k)}$ al número de raíces que hay dentro de $P_{i,j}^{(k)}$, y como $F(P_{i,j}^{(k)}) \leq K' g(l_{\text{máx}}(P_{i,j}^{(k)}), n_{i,j}^{(k)})$,

entonces $F(i,j) \leq K' \sum_{k=1}^{k_m} g(l_{\text{máx}}(P_{i,j}^{(k)}), n_{i,j}^{(k)})$.

También definimos el coste por profundidad $F_D(i) = \sum_{j=0}^{2^i-1} F(i,j)$ y el coste total $F_T(P) = \sum_{i=0}^{l_{max}(P)} F_D(i)$, como hicimos con los costes C_D, C_T, suponiendo solo errores de corte. De modo parecido a como hicimos antes, mostraremos que el coste $F_D(i)$ decrece con i.

Lema 24.

$$F(i+1,2j) + F(i+1,2j+1) \leq F(i,j)$$

Demostración. Demostraremos que

$$K' \sum_{k=1}^{k_m} g(l_{\text{máx}}(P_{i+1,2j}^{(k)}), n_{i+1,2j}^{(k)}) + K' \sum_{k=1}^{k_m} g(l_{\text{máx}}(P_{i+1,2j+1}^{(k)}), n_{i+1,2j+1}^{(k)}) \leq$$
$$\leq K' \sum_{k=1}^{k_m} g(l_{\text{máx}}(P_{i,j}^{(k)}), n_{i,j}^{(k)})$$

mostrando que para cada k,

$$g(l_{\text{máx}}(P_{i+1,2j}^{(k)}), n_{i+1,2j}^{(k)}) + g(l_{\text{máx}}(P_{i+1,2j+1}^{(k)}), n_{i+1,2j+1}^{(k)}) \leq g(l_{\text{máx}}(P_{i,j}^{(k)}), n_{i,j}^{(k)}).$$

Notemos que la profundidad máxima alcanzada desde una región, y desde su padre, verifican $l_{\text{máx}}(P_{i+1,2j}^{(k)}) \leq l_{\text{máx}}(P_{i,j}^{(k)}) - 1$ (y $l_{\text{máx}}(P_{i+1,2j}^{(k)}) \leq l_{\text{máx}}(P_{i,j}^{(k)}) - 1$). Notemos también que las raíces están particionadas, $n_{i+1,2j}^{(k)} + n_{i+1,2j+1}^{(k)} = n_{i,j}^{(k)}$. Para facilitar la notación, llamamos $x_0 = l_{\text{máx}}(P_{i+1,2j}^{(k)}), y_0 = n_{i+1,2j}^{(k)}, x_1 = l_{\text{máx}}(P_{i+1,2j+1}^{(k)}), y_1 = n_{i+1,2j+1}^{(k)}$ y $x_2 = l_{\text{máx}}(P_{i,j}^{(k)}), y_2 = n_{i,j}^{(k)}$. Por tanto $x_0 \leq x_2 - 1$ (también $x_1 \leq x_2 - 1$) y $y_0 + y_1 = y_2$ (luego $y_0^2 + y_1^2 \leq y_2^2$ porque y_0 y y_1 son no negativos, por lo que $y_0^2 + y_1^2 \leq y_0^2 + y_1^2 + 2y_0y_1 = (y_0 + y_1)^2$). Se concluye que:

$$g(l_{\text{máx}}(P_{i+1,2j}^{(k)}), n_{i+1,2j}^{(k)}) + g(l_{\text{máx}}(P_{i+1,2j+1}^{(k)}), n_{i+1,2j+1}^{(k)}) =$$
$$= g(x_0,y_0) + g(x_1,y_1) = \frac{x_0(x_0+3)}{4} y_0^2 + x_0y_0 + \frac{x_1(x_1+3)}{4} y_1^2 + x_1y_1 \leq$$
$$\leq \frac{(x_2-1)(x_2+2)}{4} y_0^2 + \frac{(x_2-1)(x_2+2)}{4} y_1^2 + (x_2-1)(y_1+y_0) \leq$$

$$\leq \frac{(x_2-1)(x_2+2)}{4}(y_0+y_1)^2 + (x_2-1)y_2 \leq$$
$$\leq g(x_2,y_2) = g(l_{\text{máx}}(P_{i,j}^{(k)}), n_{i,j}^{(k)}).$$

La primera desigualdad viene de la relación entre x_0, x_1 y $x_2 - 1$, y la segunda entre y_0, y_1 y y_2.

□

De modo similar a como hicimos antes para el coste sin errores diferidos, ahora extendemos el incremento de la cota de coste por nodo del lema 22 a incremento de la cota del coste por nivel de profundidad, para ambos tipos de error. El coste total de las regiones a profundidad i es $F_D(i) = \sum_{j=0}^{2^i-1} F(P_{i,j})$, y el coste total de división es $F_T(P_I) = \sum_{i=0}^{l_{\text{máx}}(P_I)} F_D(i)$, siendo $l_{\text{máx}}(P_I)$ la máxima profundidad de recursión del teorema 7 *a*). Como consecuencia directa del lema 24, tenemos que $F_D(i) \leq F_D(i-1)$.

Teorema 8. *El número de EP realizadas por* $\text{RDP}(P_I, A)$*, para un polinomio* f *de grado* n *con* n_I *raíces dentro de* P_I*, región de borde* $\Gamma : [a,b] \to \mathbb{C}$*, es menor o igual que*

$$(16+8\sqrt{2})(n^2+2n)l_{\text{máx}}(P_I)\left(n_I\frac{l_{\text{máx}}(P_I)+3}{4}+1\right)$$

siendo $l_{max}(P_I) \leq \left\lceil \frac{6-\lg_2 3}{(2-\lg_2 3)^2}\lg_2\frac{\text{dm}_R(P_I)}{A} + \frac{\lg_2(\text{asp}(P_I))}{2-\lg_2 3} + 2 \right\rceil$.

Demostración. Para obtener la cota anterior, sumamos el número de EP requeridas por la división recursiva en subregiones, dada por el lema 24, hasta una profundidad máxima de $l_{max}(P_I)$, por el teorema 7 *a*).

Para el coste de división en cada uno de los $l_{\text{máx}}$ niveles de profundidad, tras la aplicación reiterada del lema 24 tenemos la siguiente cota del coste de un nivel.

$$F_D(i) \leq F_D(i-1) \leq F_D(i-2) \leq \cdots$$
$$\cdots \leq F_D(1) \leq F_D(0)$$

Sumando todos los niveles, el coste total de división $F_T(P_I)$ está acotado por $\sum_{i=0}^{l_{\text{máx}}(P_I)-1} F_D(i)$, porque $l_{\text{máx}}(P_I) - 1$ es la mayor profundidad en la que se hacen divisiones. Tenemos que $\sum_{i=0}^{l_{\text{máx}}(P_I)-1} F_D(i) \leq l_{\text{máx}}(P_I)F_D(0)$, y $F_D(0) = F(P_I) = Kg(l_{\text{máx}}(P_I), n_I)$. Reemplazando el valor de K y g

$$F_T(P_I) \leq (16 + 8\sqrt{2})(n^2 + 2n)l_{\text{máx}}(P_I)\left(n_I \frac{l_{\text{máx}}(P_I) + 3}{4} + 1\right)$$

□

En función del grado n, como

$$l_{max}(P) \leq \left\lceil \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \lg_2 \frac{\text{dm}_R(P_I)}{A} + \frac{\lg_2(\text{asp}(P_I))}{2 - \lg_2 3} + 2 \right\rceil,$$

el coste total es de orden $O(n_I n^2)$. Como $n_I \leq n$, esto es $O(n^3)$, si buscamos todas las raíces. Para un grado dado, como función de la precisión requerida A, es de orden $O\left(\log \frac{1}{A}\right)$.

Con esto concluye el análisis teórico del método geométrico que hemos desarrollado, que llamaremos *Contour*, para hallar raíces de polinomios. Permite restringir la búsqueda a un área de interés en el plano complejo, al contrario que los métodos iterativos más comunes.

Contour combina un predicado de inclusión (para ver si una región plana contiene alguna raíz) basado en el índice, con un método recursivo de partición en subregiones, como otros algoritmos conocidos [Ying and Katz, 1988, Renegar, 1987,Schönhage, 1982,Pan, 1997,Neff and Reif, 1996b]. La contribución de nuestro método es utilizar un predicado de inclusión robusto, que nunca falla. Esto aporta dos características. La primera es que Contour siempre acaba con una aproximación a la solución dentro de la precisión especificada, lo que ha hecho posible desarrollar una implementación práctica. La segunda es que permite una prueba formal de corrección y el cálculo del coste computacional, como se ha hecho. Contour es el primer método geométrico con esas características.

Sobre la complejidad computacional, la cota en el número de evaluaciones polinómicas requeridas por el algoritmo es de orden $O\left(\frac{1}{A}\right)$ con respecto a la precisión

A. Aunque nuestro interés en este desarrollo es principalmente práctico, esta complejidad se puede comparar con otras cotas teóricas, siendo mejor que la de [Pan, 2001a] y similar a [Neff and Reif, 1996b]. La complejidad con respecto al grado, que es $O(n_c^3 n)$, no parece prometedora, siendo $O(n \lg n)$ la de [Neff and Reif, 1996b]. Sin embargo debe notarse que el factor n_c es el número de raíces contenidas en la región de búsqueda, que puede ser mucho menor que el número total de raíces. Notemos también que la cota del teorema 8 es extremadamente pesimista, y muy probablemente puede mejorarse.

En la práctica el coste de nuestro algoritmo puede ser mucho menor, porque se puede relajar la hipótesis en la sección 3.4 de que cada raíz requiere profundidad de recursión maximal. Además, entre los métodos geométricos para hallar raíces, es usual implementar un procedimiento iterativo (de mayor orden de convergencia, como el de Newton) para buscar una raíz cuando el diámetro de la región está por debajo de cierto umbral que asegura su convergencia (por ejemplo el umbral de [Traub and Woźniakowski, 1979]). Esta característica no se ha comentado, pero disminuye el coste sin comprometer la corrección.

Nos hemos centrado en producir una implementación práctica, que pueda operar en condiciones de trabajo, en contraste con otros métodos geométricos que solo tienen implementaciones de laboratorio. Puede probarse en la dirección web `http://gim.unex.es/contour`. Como trabajo futuro, se está desarrollando una implementación paralela, que aprovecha el paralelismo de tareas implícito en los métodos geométricos.

Capítulo 4

Implementación y comparativa

En este capítulo desarrollamos una comparativa del método geométrico *Contour* con otros métodos para hallar raíces. Nuestro método está implementado en coma flotante, y evita los problemas de singularidad que afectan a otros métodos geométricos con un mecanismo de gestión de errores, como se ha descrito en los capítulos anteriores. Está basado en un predicado de inclusión aplicable a regiones de forma arbitraria, y en un procedimiento recursivo que da origen a una búsqueda recursiva. El predicado de inclusión utiliza el índice de curvas planas cerradas, y usaremos el procedimiento eficiente para calcular el índice desarrollado en el capítulo anterior. Además, se ha determinado una cota del coste computacional del predicado.

4.1. Planteamiento

Dada la cantidad de métodos de extracción de raíces de polinomios, es necesaria una clasificación previa para poder comparar su utilidad en una situación dada. Los textos de exposición [Press et al., 1992], o los repositorios de software como *GNU Scientific Library* [Galassi and Gough, 2009] suelen clasificar los métodos numéricos para solucionar ecuaciones con criterios acerca de si un método es válido sólo para polinomios, o para funciones más generales, si da soluciones en la recta real o puede usarse para todo el plano complejo, etc. Estos criterios de clasificación son útiles para buscar entre un repertorio de métodos el más adaptado a un problema concreto, pero no resultan útiles para analizar los algoritmos, ni

para ver los recursos que demanda alcanzar un rendimiento dado, ni facilita la modificación para adaptarlos a situaciones concretas del problema planteado. Otros textos de análisis numérico [Ralston and Rabinowitz, 1978a] optan por un criterio más pedagógico de sencillez de análisis y generalidad de aplicaciones. Aunque este enfoque abstracto tampoco permite una comparativa directa entre diversos métodos.

Nuestra clasificación en iterativos o geométricos, aunque grosera, ayuda a encuadrar los métodos. Este esquema de clasificación es el que se va a usar, con una clase de métodos de aproximación que construyen una sucesión que converge a una solución de la ecuación, y otra clase de localización o búsqueda, que delimitan regiones donde están contenidas las raíces, o algunas de ellas.

A la hora de hacer una comparativa, además de elegir ciertos métodos representativos de los diversos tipos de nuestra clasificación, también es necesario integrar en la batería de pruebas las técnicas habituales para el manejo de la estructura polinomio: el esquema de evaluación Horner [Ralston and Rabinowitz, 1978a], y la aceptación y refinamiento de raíces, con deflación. También es pertinente una discusión sobre el condicionamiento (dependencia de las raíces respecto de la indeterminación de los coeficientes). La batería de pruebas desarrollada es similar a la propuesta por [Traub and Woźniakowski, 1979], ampliada con polinomios LPC y otros propuestos por [Bini and Fiorentino, 2000].

Solo se han considerado métodos clásicos, en el sentido de que tienen una primera fase de localización y luego otra de aproximación. No se han considerado métodos de transformación del dominio del problema (por ejemplo los que reformulan el problema de hallar raíces de un polinomio en el de hallar autovalores de una matriz). De este tipo son los métodos de la potencia inversa o el de Bernouilli (*QD*, diferencia de cocientes) [Bini et al., 2004a, Zeng, 2005, Fortune, 2002]. Tampoco se han considerado polialgoritmos. Un polialgoritmo es una combinación de varios algoritmos diseñada para cierta clase de problema [Knuth, 1981]. Las bibliotecas de funciones numéricas suelen implementar polialgoritmos cuya complicada casuística queda oculta al usuario (métodos "de caja negra"). El más usado es el de Jenkins-Traub [Jenkins and Traub, 1975]. No se han considerado este tipo de métodos buscando la sencillez a la hora de hacer la comparación.

El método para encontrar raíces basadas en estimaciones del número de vueltas de curvas planas descrito en los capítulos precedentes se ha implementado como

una biblioteca `C` llamada *Contour*. Lo hemos diseñado con el objetivo de encontrar las raíces complejas dentro de regiones específicas del plano, evitando los cálculos en otras áreas. Este enfoque produce un ahorro computacional cuando se compara con otros métodos. Probamos *Contour* contra tres métodos de cálculo de raíces: dos iterativos (Newton, el método estándar para resolver ecuaciones no lineales, y Durand-Kerner [Werner, 1982], que aproxima simultáneamente todas las raíces) y otro método geométrico (Lehmer-Schur [Loewenthal, 1993]) . Las pruebas se realizan sobre un subconjunto de la batería clásica de polinomios sugeridas por Jenkins-Traub [Jenkins and Traub, 1975], y además sobre un conjunto de polinomios obtenidos mediante modelado LPC de una señal de voz [Rabiner, 1999]. Cada prueba se ejecuta en dos plataformas diferentes: un PC de escritorio con un procesador Pentium M de 1,7 GHz, y una plataforma de Procesamiento de Señal Digital (DSP C6711 Starter Kit - DSK, de Texas Instruments Inc.).

La próxima sección revisa él método *Contour* haciendo hincapié en sus diferencias con los métodos iterativos. La siguiente sección cubre el diseño del experimento numérico, con una descripción de los conjuntos de polinomios de prueba y varias áreas de búsqueda. Incluye una traza visual del comportamiento de los métodos considerados como ejemplo. Después la sección de resultados numéricos revisa las medidas tomadas, con su discusión, seguida por una sección de conclusiones.

4.2. Descripción de métodos

Los algoritmos iterativos hallan las raíces sucesivamente, pero en orden aleatorio. La posición de la raíz hallada es impredecible. Nuestro método *Contour* no es iterativo sino geométrico, es decir, basado en acotar la zona de búsqueda, y las raíces que se encuentran están dentro de este área. *Contour* puede ser visto como un procedimiento de separación de raíces. Estos procedimientos se utilizan con frecuencia para controlar la convergencia de los métodos iterativos. Por ejemplo, en el problema particular de encontrar las raíces reales de un polinomio, éstas se separan previamente en segmentos por sucesiones de Sturm [Ralston and Rabinowitz, 1978a]. En cada segmento, la cuestión de la imprevisibilidad de la convergencia es más manejable que en toda la recta real. De ahí el énfasis tradicionalmente puesto en los criterios de convergencia para el análisis de los métodos iterativos. Como otro ejemplo de procedimiento de separación de raíces en el caso de raíces comple-

jas, los métodos teóricos recientes tratan la naturaleza impredecible de la iteración mediante una corona de exclusión [Renegar, 1987,Pan, 1997,Pan, 2001b,Bini et al., 2004b], pero no son de uso práctico. Un paquete pensado para el uso práctico que se basa en una fase geométrica de separación de raíces junto de una etapa de aproximación iterativa es *MPSolve* [Bini and Fiorentino, 2000]. De forma similar, nuestro método puede ser visto como el uso intensivo de un procedimiento de separación de raíz, sin el requisito de multiprecisión aritmética de *MPSolve*.

El método *Contour* se inicia en una curva Γ que delimita el área de interés. Estimamos el número de vueltas de la curva $f(\Gamma)$. Si este valor no es nulo, la región plana interior a Γ contiene alguna raíz. En tal caso, región se divide en partes. El borde de cada parte es una nueva curva, y el número de vueltas de su imagen por f nos informa si contiene alguna raíz. Prosiguiendo con esta división recursiva podemos llegar a una precisión arbitraria. Como refinamiento con respecto al análisis teórico del capítulo anterior, si una de las subregiones obtenidas verifica un criterio de convergencia local para el método de Newton, este se aplica para obtener directamente la raíz contenida. Esta fase iterativa es común en las implementaciones de métodos geométricos.

A pesar de que el cálculo del índice requiere tipos de datos geométricos y un tratamiento cuidadoso de los casos singulares, no presenta problemas de convergencia o de estabilidad. Así, las raíces se pueden calcular usando una baja precisión aritmética. Además, la búsqueda de raíces puede ser restringida a un área predefinida, y el coste computacional es proporcional al número de raíces en esa zona. Estos dos hechos hacen que el método *Contour* sea preferible en ciertas situaciones: polinomios con raíces múltiples, o con un área de interés restringida. Cabe señalar que la naturaleza geométrica del método hace que sea naturalmente paralelizable, sin comunicación entre las tareas a cargo de la búsqueda de raíces en zonas disjuntas. Por lo tanto, la ganancia en tiempo de ejecución debe ser lineal con respecto al número de procesadores. Este campo de investigación no se aborda en la presente memoria.

4.3. Diseño del experimento

Se compara *Contour* con una implementación del método de Newton, que utiliza las recetas comunes de deflacción estable, la variación de la semilla inicial y

el pulido de las raíces [Fortune, 2002]. También probamos otro método iterativo, Durand-Kerner [Werner, 1982] para la determinación simultánea de las raíces, así como otro un método geométrico, Lehmer-Schur [Loewenthal, 1993], para enriquecer la comparación con *Contour*. El método de Newton exige precisión doble para evitar inestabilidades numéricas [Jenkins and Traub, 1975]. En la aritmética de punto flotante IEEE 754, esto significa una mantisa de 53 bits y un exponente de 11 bits, dando una precisión de 16 dígitos decimales. En contraste, *Contour,* Durand-Kerner y Lehmer-Schur sólo requieren precisión simple (24 bits de mantisa, exponente 8, con precisión de alrededor de 7 dígitos decimales) en el rango de grados polinómicos examinados (de 3 a 128).

4.3.1. Polinomios aleatorios y de procesamiento de señal

Al montar la batería de polinomios de prueba, se han tenido varios objetivos en mente: en primer lugar, representar el campo de polinomios más amplio posible; en segundo lugar, manifestar el comportamiento de los métodos en los casos problemáticos; y, por último, estudiar los casos que surgen con frecuencia en la práctica.

Para el primer objetivo, el enfoque simple de tomar coeficientes aleatorios es insuficiente para cubrir el conjunto de polinomios como objeto matemático, de una manera representativa. Se sabe que las raíces de estos polinomios, con coeficientes elegidos como variables aleatorias independientes e idénticamente distribuidas, tienden a concentrarse siguiendo el perímetro del círculo unidad, uniformemente espaciados. Además estos polinomios están bien condicionados con respecto a la búsqueda de raíces, y por tanto son fáciles de resolver [Jenkins and Traub, 1975]. Es preferible, para tener una muestra amplia del conjunto de polinomios, elegir aquellos que tengan las raíces, no los coeficientes, distribuidas aleatoriamente de modo independiente con idéntica distribución.

Para el segundo objetivo de la batería de prueba, nos centramos en polinomios con agrupaciones de raíces (*clusters*, grupos de raíces cercanas entre sí), que presentan problemas de convergencia a los buscadores de raíces de uso frecuente. Los polinomios que surgen en aplicaciones prácticas pueden mostrar este comportamiento mal condicionado (con raíces múltiples o cercanas entre sí). Para el tercer objetivo, queremos aplicar nuestro método para el análisis LPC en procesado de

señal. Por lo tanto una colección de estos polinomios debe incluirse en la batería.

Resumiendo, nuestra batería de polinomios de prueba se compone de los siguientes cuatro tipos, todos de coeficientes reales. Para cada tipo, se muestra un polinomio de grado 32 como ejemplo. Se representan los 33 coeficientes (empezando por el término independiente) y las raíces. El círculo unidad se indica en la gráfica de las raíces.

Los polinomios de tipo 1 tienen coeficientes aleatorios entre -1 y 1. A pesar de los comentarios anteriores sobre la distribución de sus raíces alrededor de la circunferencia del círculo unidad, este tipo de polinomio es recurrente en experimentos numéricos, y puede ser útil para la comparación con otros estudios. La figura 4.1 muestra un ejemplo.

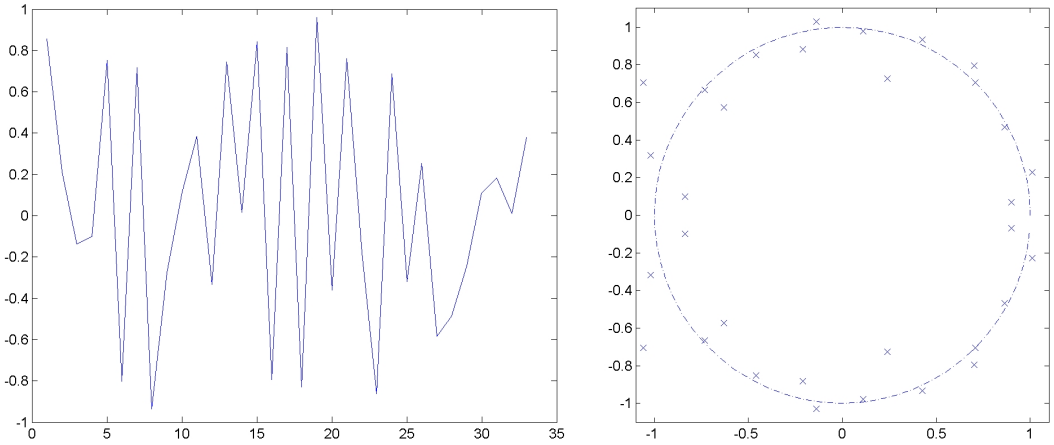


Figura 4.1: Ejemplo de un polinomio de tipo 1 y sus raíces.

Los polinomios de tipo 2 están construidos de tal manera que sus raíces son aleatorias, uniformemente dispersas dentro del círculo unidad, en pares complejos conjugados para tener coeficientes reales. Así se cumple el primer objetivo de prueba (véase la figura 4.2). Como es conocido [Kyurkchiev, 1998], el valor absoluto de los coeficientes de un polinomio de raíces al azar sigue una curva de campana. Por esta razón, los coeficientes de grado por encima de 22 son cercanos a cero.

Para el segundo objetivo (probar las capacidades de convergencia), construimos polinomios con raíces en *clusters*. Cada polinomio tipo 3 tiene el 75 % de sus raíces uniformemente dispersas en el círculo unidad, y el resto uniformemente disperso en un círculo de radio 0'01 centrado en el punto $-0'5 + 0i$. La figura 4.3 muestra

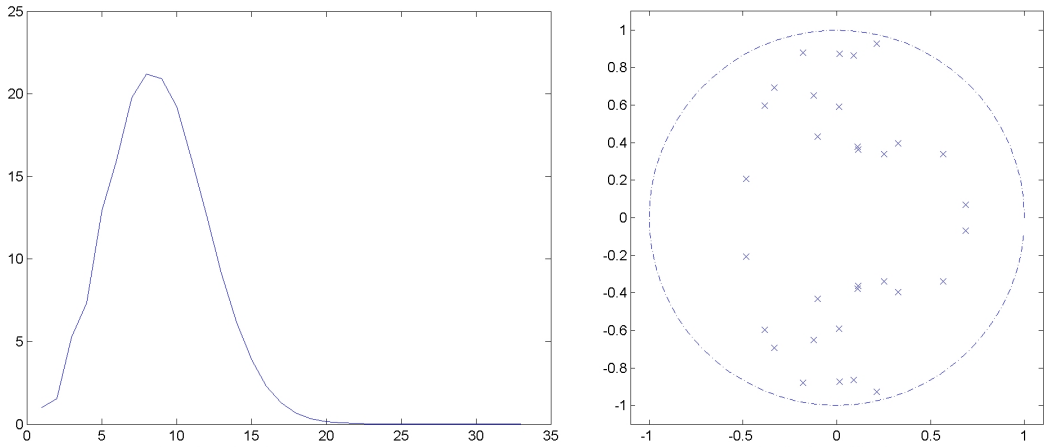


Figura 4.2: Ejemplo de un polinomio de tipo 2 y sus raíces.

un ejemplo.

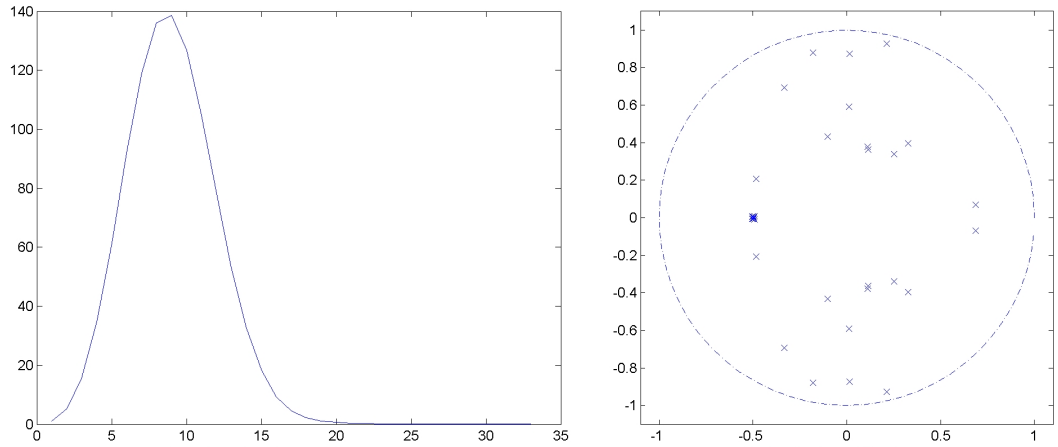


Figura 4.3: Ejemplo de un polinomio de tipo 3 y sus raíces.

Por último, los polinomios del tipo 4 de nuestra batería de pruebas se obtienen a partir del análisis LPC de una señal de voz. Las raíces LPC están situadas dentro del círculo unidad por cuestiones de estabilidad [Markel, 1976]. Las raíces cerca de la circunferencia están en conexión con las componentes espectrales de la señal. Por ejemplo, el polinomio de tipo 4 mostrado en la figura 4.4 corresponde a una ventana de 512 muestras de un tramo vocálico, y resulta tener cinco pares de raíces de módulo mayor que 0'98, cuyos argumentos corresponden a los picos en el espectro de frecuencia de la señal de voz [Rabiner, 1999].

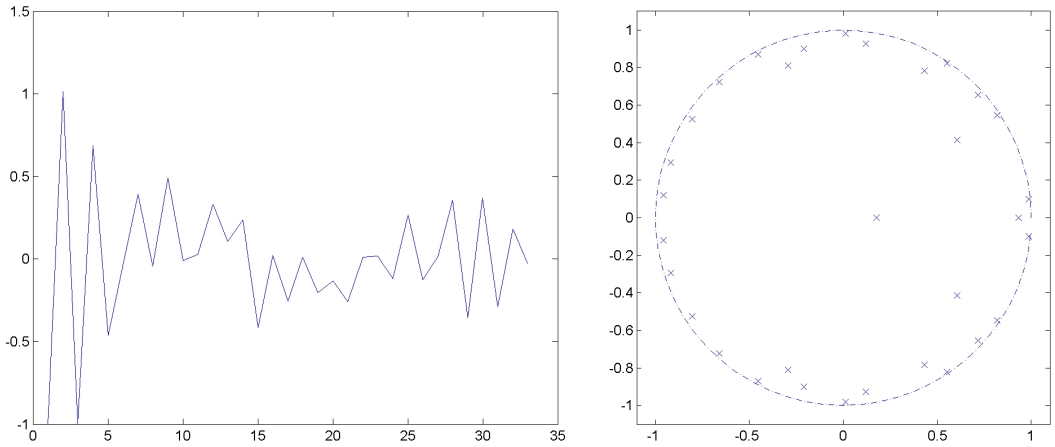


Figura 4.4: Ejemplo de un polinomio de tipo 4 y sus raíces.

En los cuatro tipos de polinomios, las raíces vienen en pares conjugados complejos, ya que tienen coeficientes reales. Ampliar de la batería de pruebas con polinomios de coeficientes complejos es directo, pero muestran un comportamiento similar a los reales en la densidad de las raíces, y previsiblemente en el rendimiento de los algoritmos, por lo que no se ha acometido.

4.3.2. Áreas para los métodos geométricos

Para probar los métodos geométricos, *Contour* y Lehmer-Schur, elegimos tres áreas del plano complejo para buscar raíces. Un área incluye todas las raíces, lo que es útil para comparar con los tiempos de los métodos iterativos, que hallan todas las raíces. Por otra parte, para las aplicaciones en LPC tenemos que encontrar las raíces situadas cerca del perímetro del círculo unidad, por lo tanto es pertinente considerar un área que cubra este margen. Para ello, probamos los métodos geométricos sobre estas tres regiones del plano complejo: el cuadrado de vértices $1+i$, $1-i$, $-1+i$, $-1-i$; la corona semicircular $0'95 < r < 1$, y la corona semicircular $0'99 < r < 1$, llamadas respectivamente zona A, zona B y zona C.

La zona A es un cuadrado centrado en el origen que encierra el círculo unidad, incluyendo todas las raíces de los polinomios de tipo 2, 3 y 4, y la mayor parte de las de los de tipo 1. El propósito de este área es comprobar el coste de encontrar todas las raíces por los dos métodos geométricos. Para esta tarea se podría restringir la búsqueda al semiplano superior, porque las raíces aparecen en pares conjugados.

Sin embargo, el tiempo necesario para encontrar las raíces en el área más pequeña es simplemente la mitad que en la zona A, por lo que es igualmente válida para comparación.

La zona B es una corona de anchura 0'05, y la zona C una análoga de anchura 0'01. Para encontrar sólo las raíces situadas cerca del perímetro del círculo unidad, la búsqueda está limitada a la mitad superior de la corona circular centrada en el origen, delimitada exteriormente por este perímetro. Estas zonas contienen las raíces de interés en las aplicaciones de procesamiento de señal. Para aplicar el método *Contour,* usamos ocho trapecios para cubrir las áreas, encontramos las raíces contenidas en cada uno de ellos, sumando los tiempos tomados. En un entorno paralelo estos trapecios pueden ser asignados a procesadores independientes. Del mismo modo, para el método geométrico Lehmer-Schur cubrimos el área de interés utilizando círculos, ya que este método sólo se puede aplicar a regiones de esta forma. La figura 4.5 muestra las áreas de prueba.

4.3.3. Métodos a ensayar

Antes de exponer el experimento numérico, se muestra un bosquejo de los cálculos requeridos por estos métodos (Newton, *Contour,* Lehmer-Schur y Durand-Kerner). Así se dará un marco de referencia para la interpretación de los resultados.

El método de Newton da 284 pasos para encontrar las raíces del polinomio de ejemplo del tipo 3 (LPC) de grado 32 mostrado en la figura 1.5, con una precisión de 4 cifras decimales (10^{-5}). Un paso consiste en la evaluación del polinomio y de su derivada en un punto complejo. La evaluación de un polinomio de grado n se puede hacer con aproximadamente $n/2$ multiplicaciones complejas [Knuth, 1981]. Si denotamos x_k la estimación k-ésima de una raíz del polinomio $f(x)$, un paso consiste en calcular la siguiente estimación:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Los 284 pasos x_k para el polinomio de ejemplo (equivalente a aproximadamente 4500 multiplicaciones) se muestran en la figura 4.6 en forma de puntos, y las raíces como cruces.

Si la convergencia se produce, los pasos sucesivos dan estimaciones mejoradas, y por tanto se colocan a lo largo de un camino que conduce a cierta raíz. En nuestra

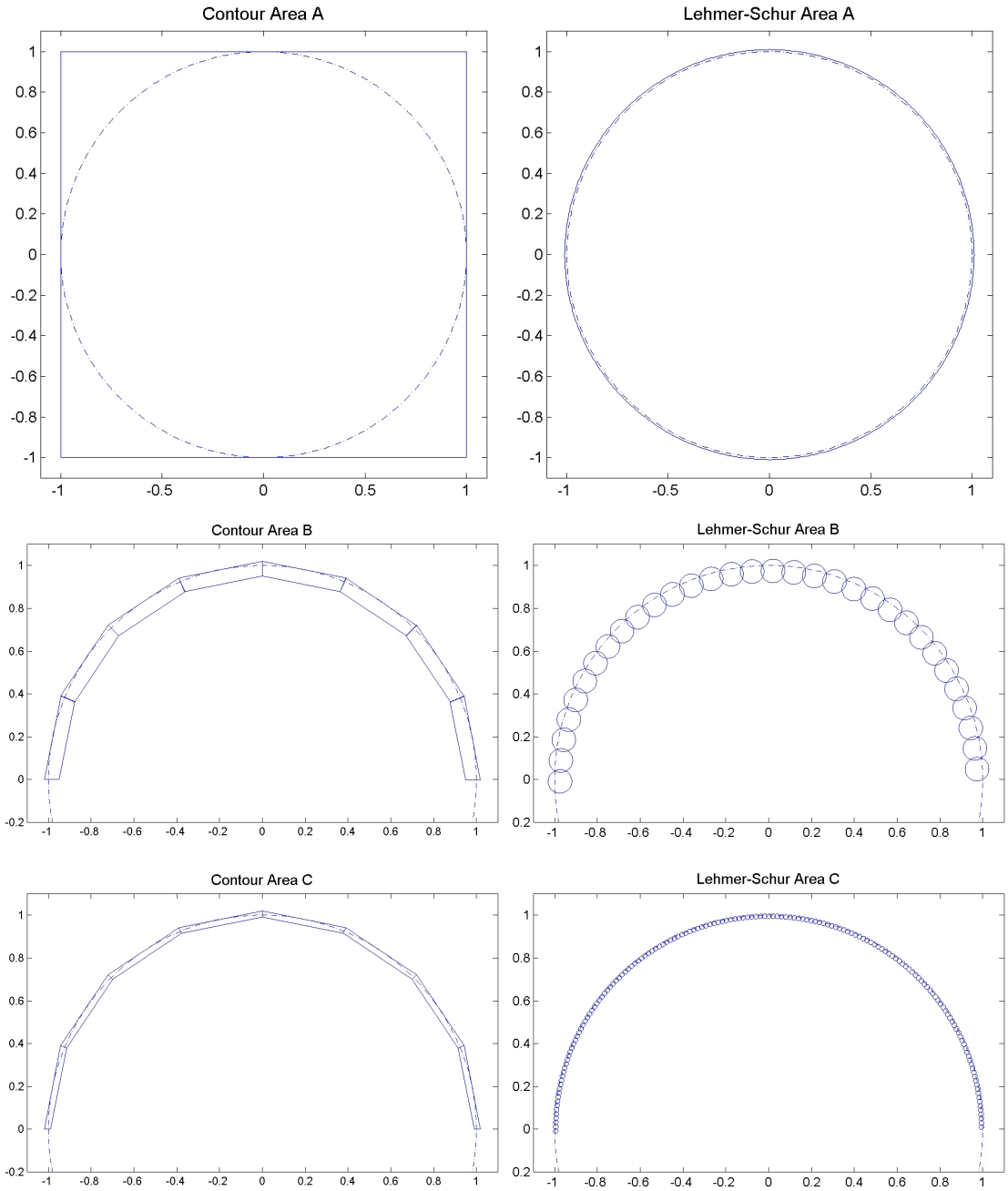


Figura 4.5: Las tres áreas de prueba. El círculo unitario se perfila como referencia.

implementación, el punto inicial es el complejo $-1 + 0'1i$, donde la mayoría de los caminos empiezan. Pero si esta aproximación inicial no converge a una raíz en cierto

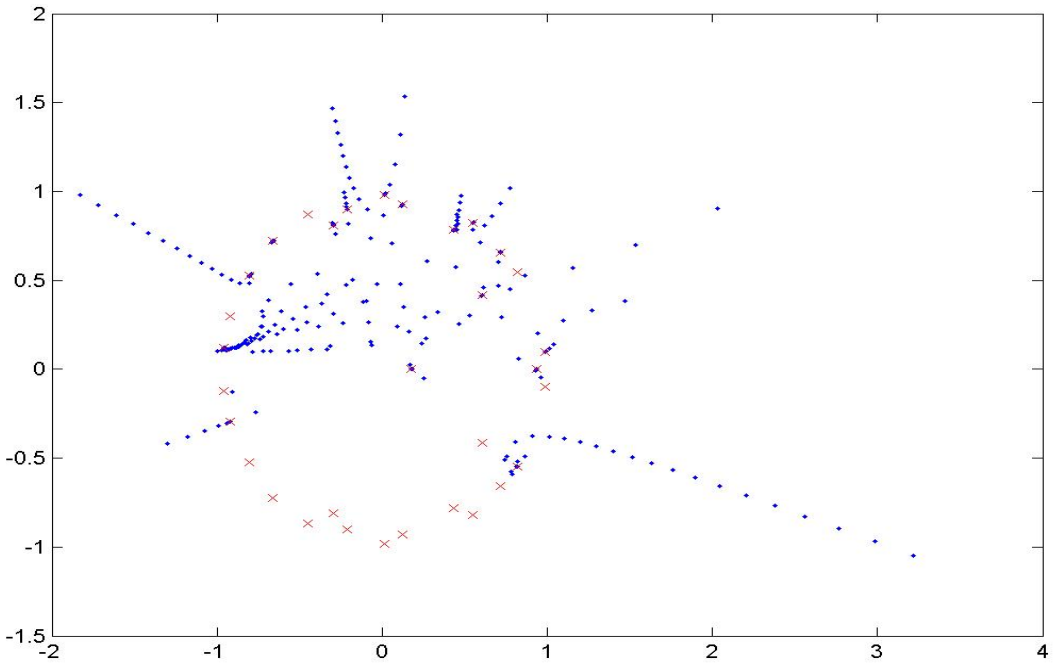


Figura 4.6: Pasos requeridos por el método de Newton.

número de pasos, se utiliza otro punto inicial. En el ejemplo, esto se representa por los caminos que comienzan en $3'2 - i$ o $-1'9 + i$. Cuando se encuentra una raíz, ella y su compleja conjugada son deflactadas del polinomio.

En contraste, el método *Contour* utiliza evaluaciones polinómicas en los puntos correspondientes a bordes de regiones. A continuación, calcula su índice, y por tanto el número de raíces que el borde encierra. Como ejemplo, se parte de la frontera de la zona A y representamos las evaluaciones requeridas por este contorno y sus subdivisiones *quadtree*. El mismo polinomio LPC anterior requiere 2790 evaluaciones (44000 multiplicaciones), que se muestran como puntos en la figura 4.7.

Como se ha comentado antes, el centro de algunos subcontornos verifica un criterio de convergencia de un método iterativo, por ejemplo $0'9 + 0'6i$. Las aproximaciones sucesivas de la fase iterativa del método *Contour* se acercan a la raíz contenida en ese subcontorno. Algunos otros subcontornos (por ejemplo, los que contienen la raíz real 0'93) tienen centros en los que la fase iterativa no converge, y por tanto hay que recurrir a la división recursiva hasta alcanzar la precisión especificada.

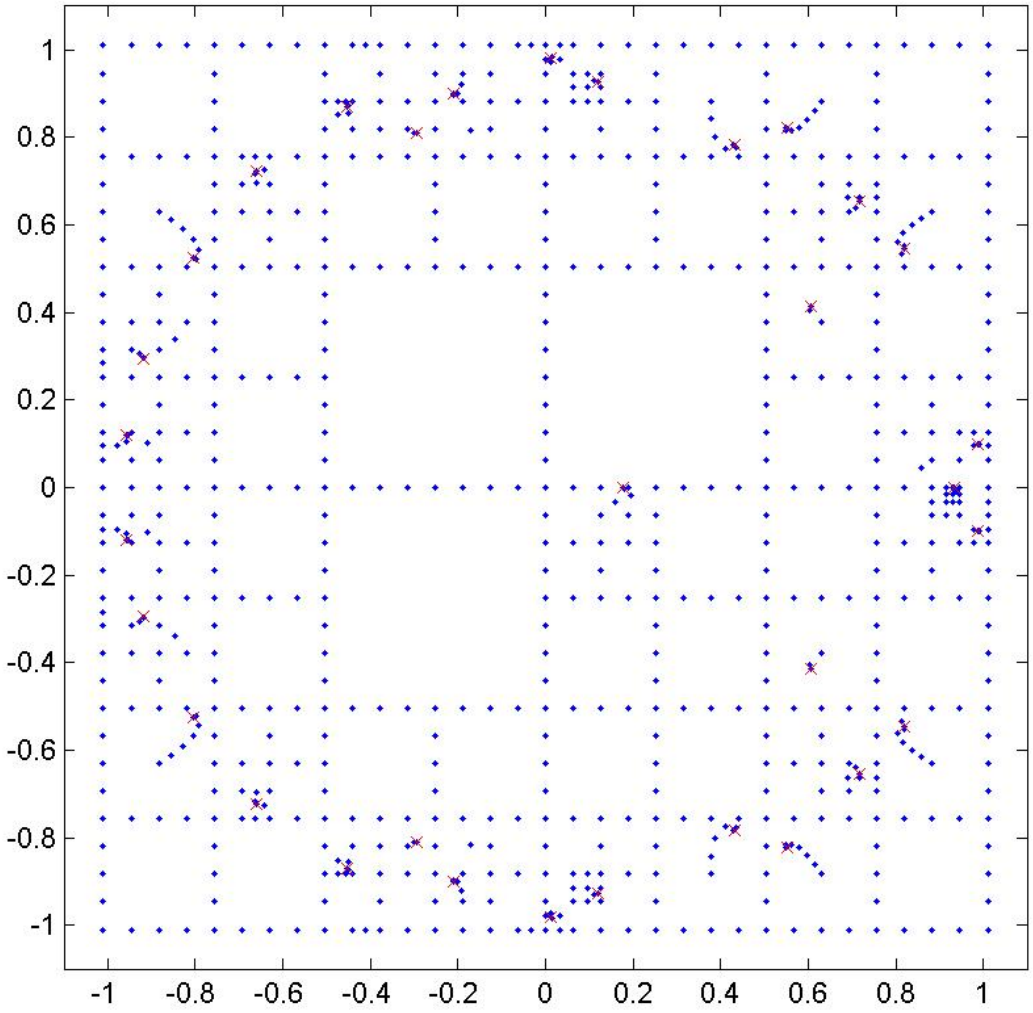


Figura 4.7: Evaluaciones polinómicas requeridas por el método *Contour* en la zona A.

Por otro lado, el método *Contour* aplicado a la zona B requiere sólo 150 evaluaciones polinómicas, los puntos de la figura 4.8 (2400 multiplicaciones). Bajo el supuesto de que un paso de Newton es aproximadamente equivalente a una evaluación polinómica, la demanda computacional del método *Contour* en la zona A es diez veces mayor que la del método de Newton. Sin embargo en la zona B el método *Contour* requiere aproximadamente la mitad de los cálculos que el de Newton. La mejora sobre el método de Newton viene de la reducción de superficie donde el método *Contour* busca las raíces.

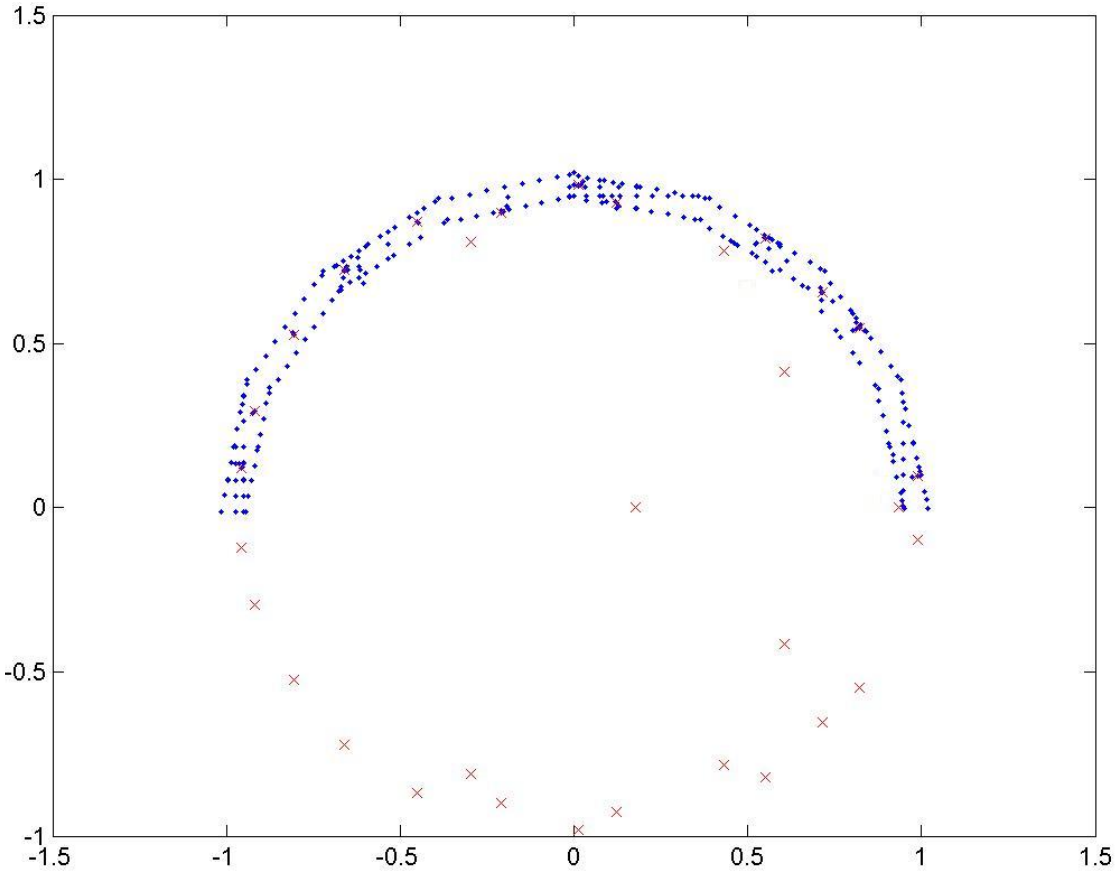


Figura 4.8: Evaluaciones polinómicas requeridas por el método *Contour* en la zona B.

El número de multiplicaciones requeridas por este polinomio de ejemplo usando los otros dos métodos es mayor, como veremos a continuación.

El método iterativo de Durand-Kerner reestima todas las raíces simultáneamente en cada iteración. Si n es el grado del polinomio $f(x)$ y si $(x_1^k, x_2^k, \ldots, x_n^k)$ son n números complejos, considerados como estimaciones de las raíces en la iteración k-ésima del método, los nuevos valores en la iteración $(k+1)$-ésima son:

$$x_i^{k+1} = x_i^k - \frac{f(x_i^k)}{\prod_{j=1, j\neq i}^{n}(x_i^k - x_j^k)} \qquad i = 1, \ldots, n$$

Los valores iniciales $(x_1^0, x_2^0, \ldots, x_n^0)$ se eligen en un círculo que rodea todas las raíces. Cada iteración requiere n evaluaciones polinómicas y n^2 multiplicaciones. Se llega a las raíces del polinomio ejemplo LPC en 30 iteraciones (equivalente a

46000 multiplicaciones). Estos valores se muestran en la figura 4.9, con una línea que une los valores sucesivos x_i^k, x_i^{k+1} de cada estimación.

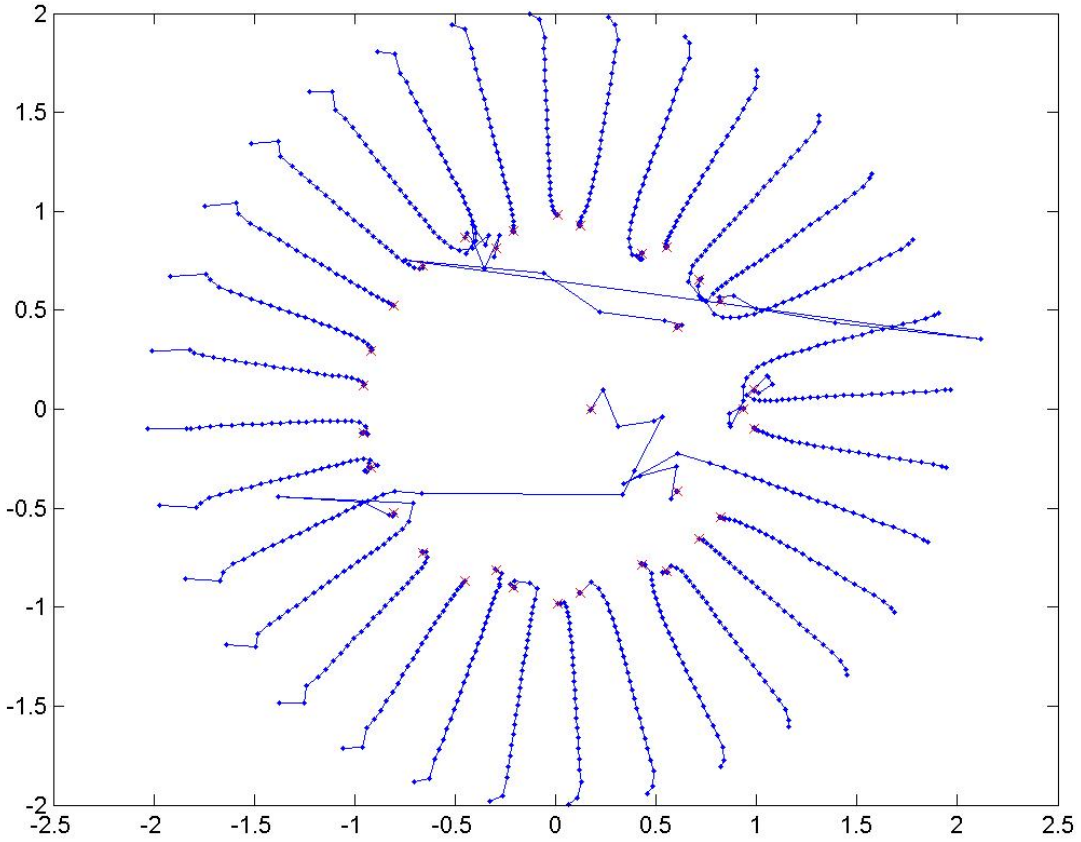


Figura 4.9: Iteraciones requeridas por el método Durand-Kerner.

Finalmente, el método Lehmer-Schur se basa en el criterio de Schur-Cohn [Pan, 1997], que decide si hay alguna raíz de un polinomio de grado n dentro de un círculo, con un coste de n evaluaciones (n^2 multiplicaciones). El método aplica este criterio de forma recursiva: si un círculo contiene alguna raíz, se cubre con 9 círculos de menor diámetro, y cada uno de ellos se somete al criterio. Si hay resultado positivo, el círculo pequeño se divide de manera similar, y así sucesivamente hasta alcanzar la precisión requerida. En el polinomio LPC de ejemplo, para encontrar todas las raíces se requiere la aplicación del criterio de Schur-Cohn a 6364 círculos (equivalente aproximadamente a un millón de multiplicaciones), lo que se muestra en la figura 4.10:

El círculo inicial es el círculo unidad, y se cubre con un círculo de centro el

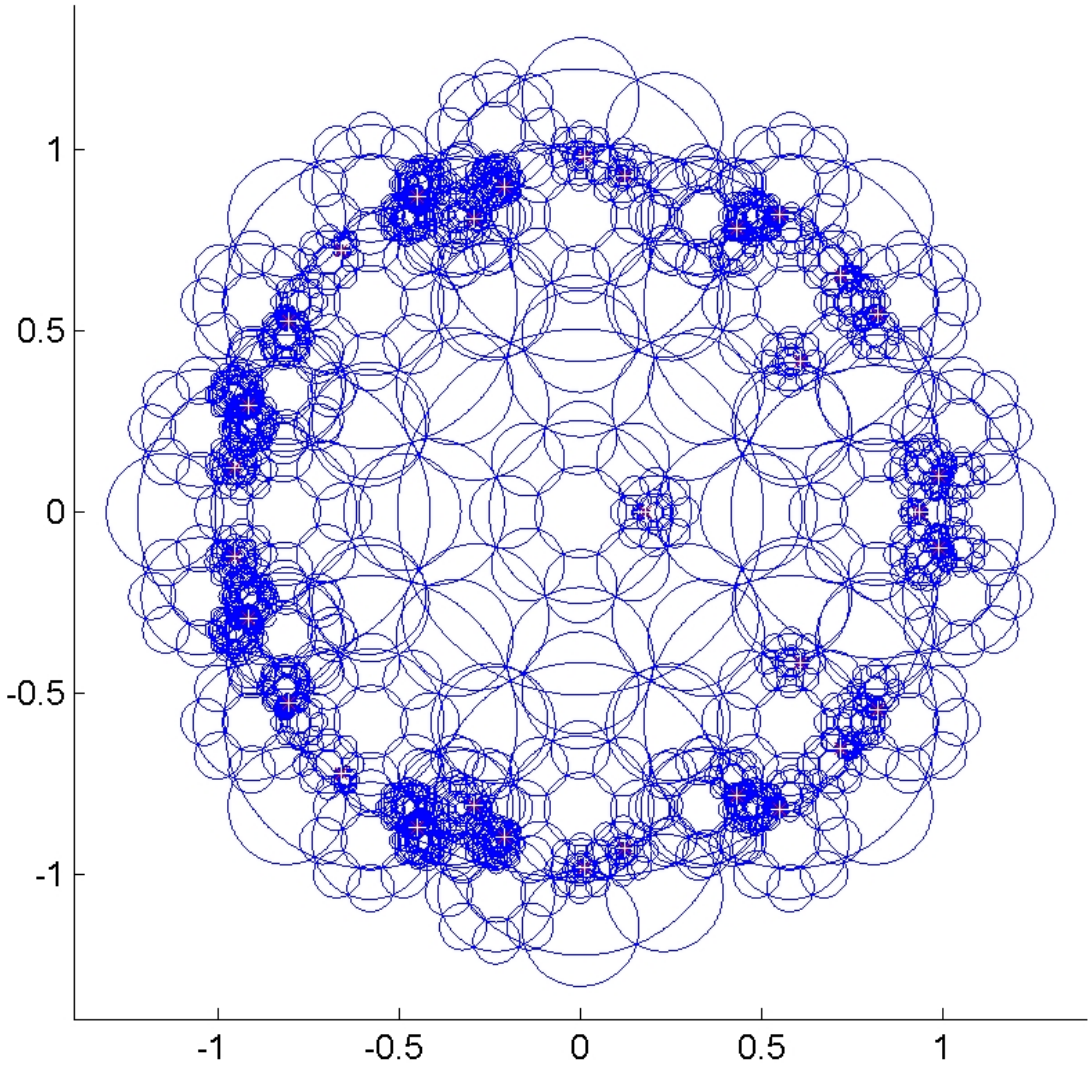


Figura 4.10: Aplicaciones del criterio requeridas por Lehmer-Schur en la zona A.

origen, de radio de 0'5, y ocho más en torno a éste a izquierda, derecha, arriba, abajo y posiciones intermedias. Es decir, sus centros son aproximadamente 0'75, $0'6 + 0'6i$, $0'75i$, $-0'6 + 0'6i$, $-0'75$, $-0'6 - 0'6i$, $-0'75i$, $0'6 - 0'6i$. Si algunos de estos círculos contiene alguna raíz, se lleva a cabo un recubrimiento por círculos similares, como se ha descrito. En la figura, hay más círculos alrededor de las raíces, que están marcadas con una cruz blanca. El recubrimiento con círculos implica una cierta superposición, y por tanto una raíz se puede encontrar varias

veces, si está en la intersección de dos círculos. Esta situación es más infrecuente en la búsqueda de raíces de la zona B, que requiere 851 aplicaciones del criterio Schur-Cohn (aproximadamente 10^5 multiplicaciones), mostradas en la figura 4.11.

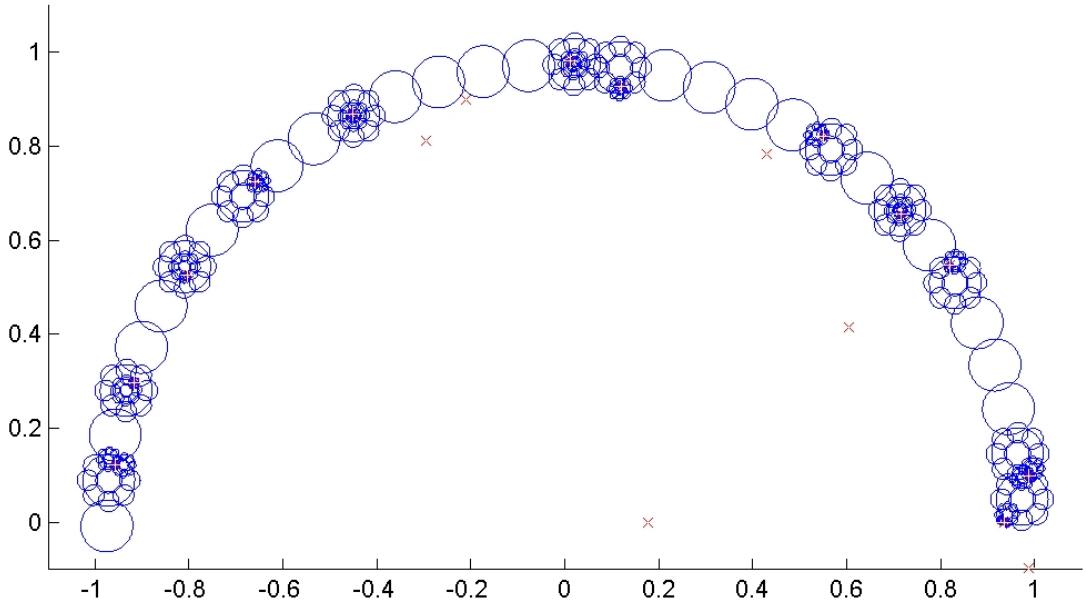


Figura 4.11: Aplicaciones del criterio requeridas por Lehmer-Schur en la zona B.

4.4. Resultados numéricos

El objetivo de nuestro experimento numérico es medir el ahorro computacional producido usando el método *Contour* para encontrar raíces en áreas reducidas. Para cada uno de los cuatro tipos de polinomios de prueba, y cada grado de 3 a 128, se generan diez polinomios, y promediamos el tiempo necesario para encontrar sus raíces con una aproximación de 10^{-4}. Esto significa la búsqueda de raíces de $4 \times 125 \times 10 = 5000$ polinomios por ocho procedimientos: los dos iterativos (Newton y Durand-Kerner), y los dos métodos geométricos (*Contour* y Lehmer-Schur) dentro de cada una de las zonas A, B y C. Las pruebas se han realizado en un PC de sobremesa (Pentium M 1.7 GHz), en un proceso planificado con la prioridad más alta para evitar interrupciones del sistema. Las mismas pruebas se han realizado en un procesador digital de señal, la plataforma DSK (DSP C6711)

de Texas Instruments. Las figuras 4.12 y 4.13 muestra los tiempos medidos en el entorno PC.

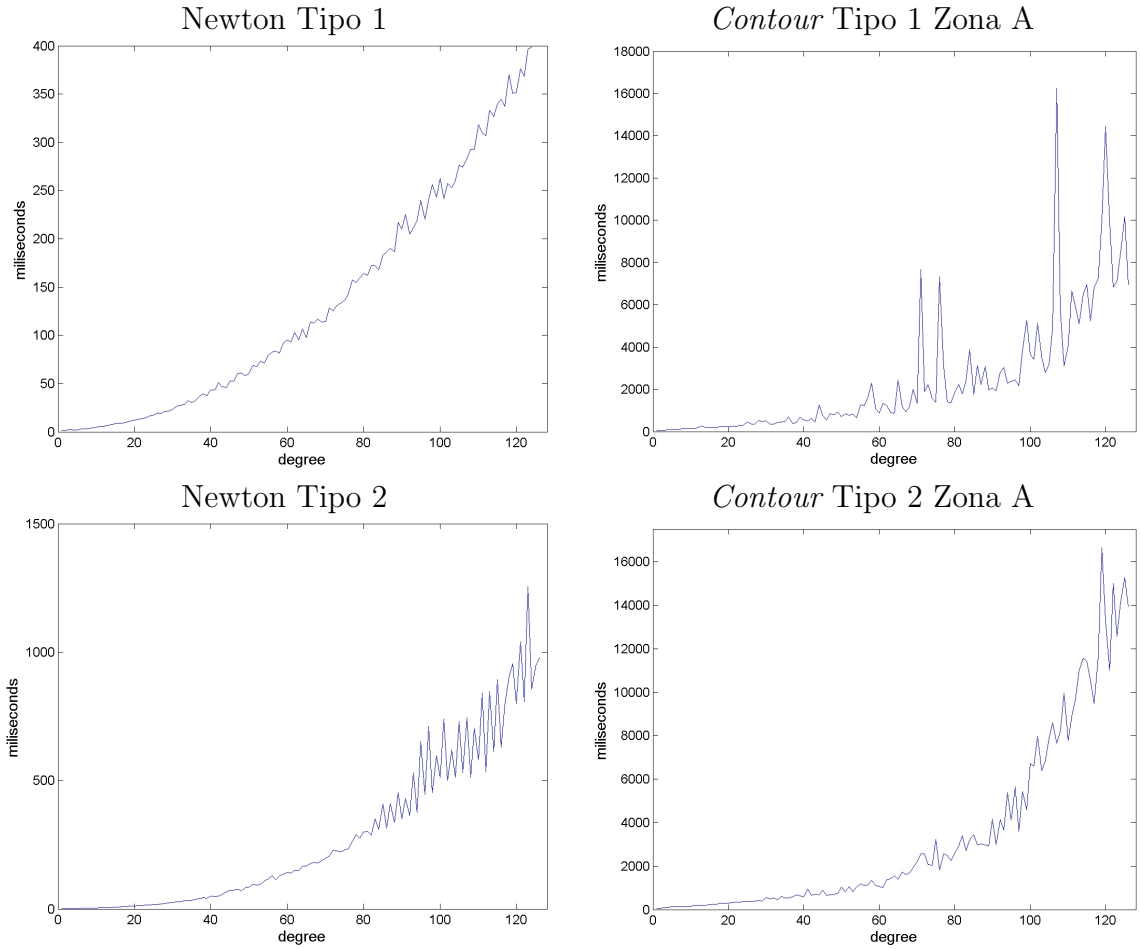


Figura 4.12: Resultados de tiempo para Newton y *Contour* en la zona A (tipos polinómicos 1 y 2) en PC.

Como puede verse, el método *Contour* en la zona A, para los cuatro tipos, tarda aproximadamente diez veces más en encontrar todas las raíces que el método de Newton (en polinomios de tipo 1, en encontrar las raíces en el interior de la zona A). Ciertamente, dentro de cada tipo y método, los polinomios de mayor grado son más costosos de resolver. Una característica que merece destacar en la imagen es que el método de Newton tiene un mejor rendimiento en los polinomios de tipo 1 y 4 (coeficientes aleatorios y LPC). Estos tienen sus raíces dispersas de manera aproximadamente uniforme en todo el perímetro del círculo unidad, facilitando

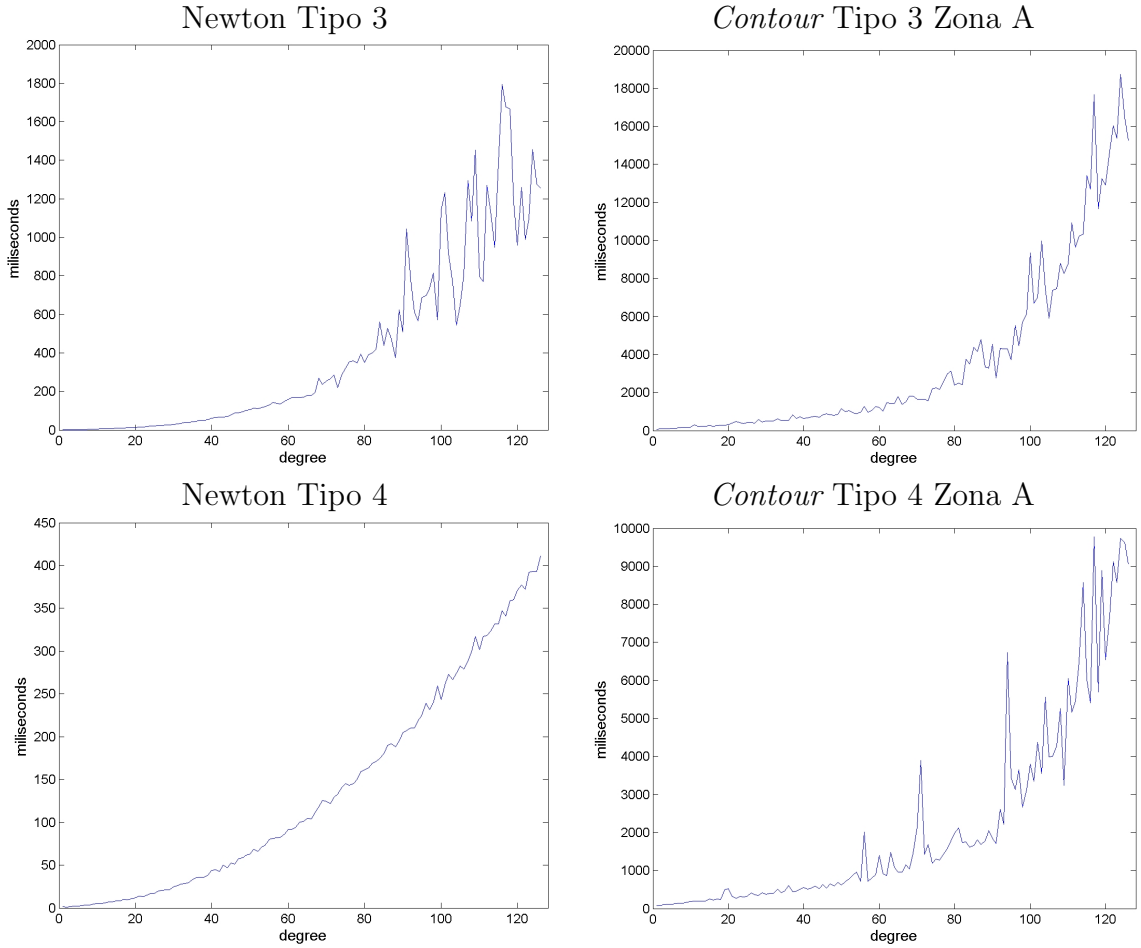


Figura 4.13: Resultados de tiempo para Newton y *Contour* en la zona A (tipos polinómicos 3 y 4) en PC.

la convergencia del método iterativo. Los polinomios de tipos 2 y 3 tienen raíces próximas entre sí lo que obstruye la convergencia del método de Newton.

En cuanto a la variabilidad de las medidas, el promedio de tiempos en los diez polinomios de cada grado para el método de Newton nos da gráficas suaves en los tipos 1, 2 y 4. El aspecto de dientes de sierra en el tipo 2 para grados entre 80 y 128 es debido al hecho de que, por construcción, con probabilidad casi uno, los polinomios con raíces aleatorias de grado par no tienen ninguna raíz real. Los de grado impar tienen con seguridad al menos una raíz real. La convergencia a raíces reales es más lenta que a complejas en el método de Newton [Press et al., 1992]. Como esta característica es compartida por los diez polinomios de los experimentos

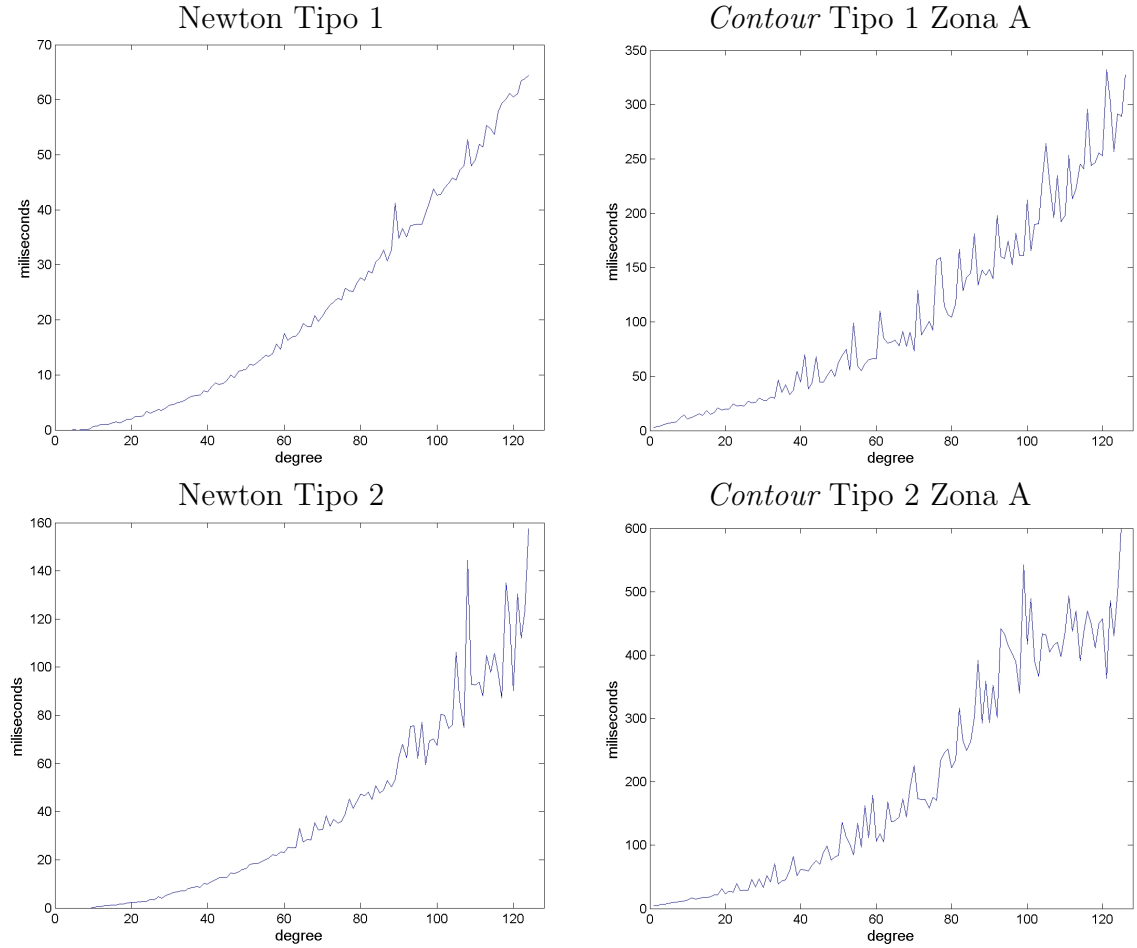


Figura 4.14: Resultados de tiempo para Newton y *Contour* en la zona A (tipos 1 y 2) en DSK.

de grado impar, el sesgo no desaparece después de la media. Además, los polinomios de tipo 3 (con un *cluster*) para el método de Newton, y los de todo tipo para el método *Contour,* dan medidas con gran variabilidad por encima del grado 80. Esto significa que el promedio de diez polinomios no es suficiente para disipar la influencia de casos especialmente difíciles. Se ha optado por no aumentar el número de casos que se promedia porque la tendencia central es claramente visible, y también porque así se representa la dispersión de la medida sin necesidad de calcular estadísticos superiores. En el entorno DSK las cifras son comparables (figuras 4.14 y 4.15).

Aunque las gráficas anteriores descalifican a *Contour* frente a Newton como

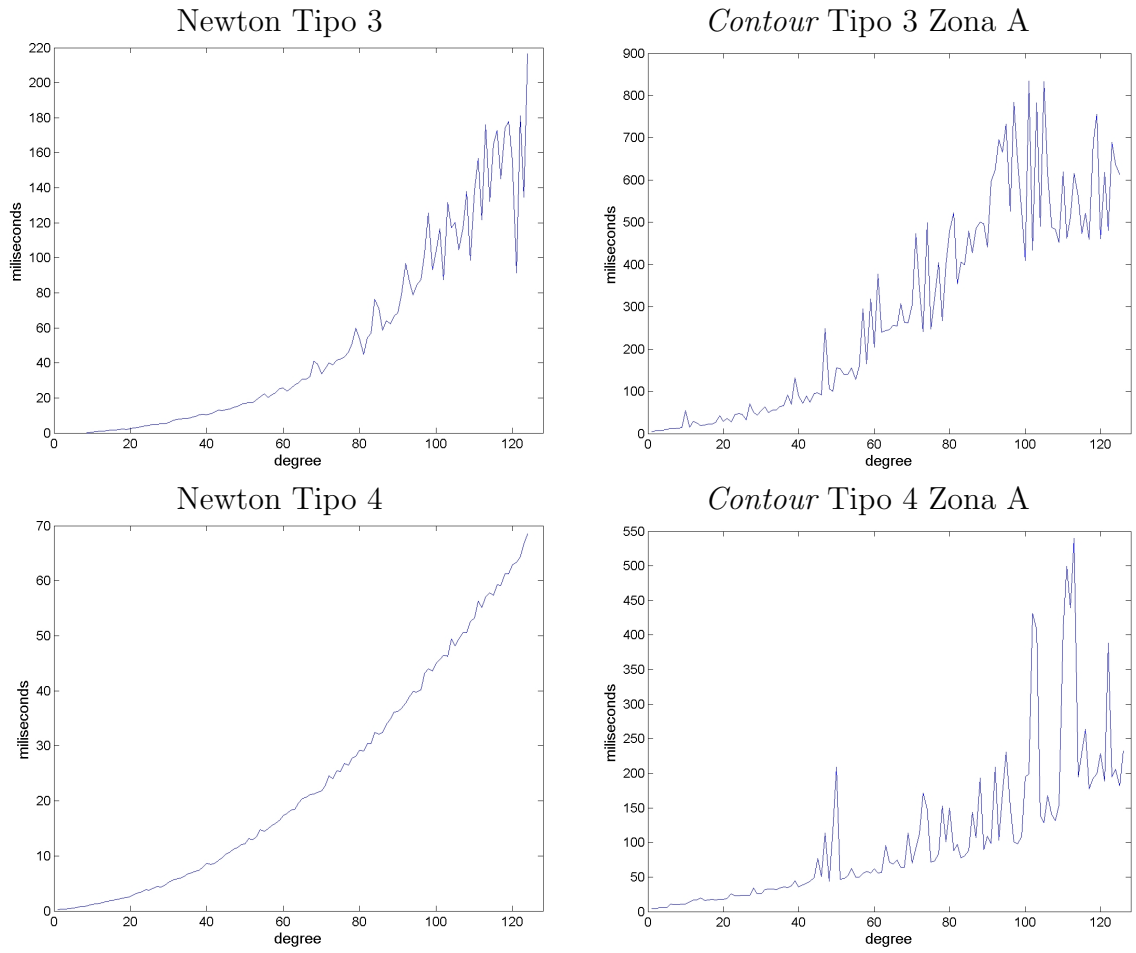


Figura 4.15: Resultados de tiempo para Newton y *Contour* en la zona A (tipos 3 y 4) en DSK.

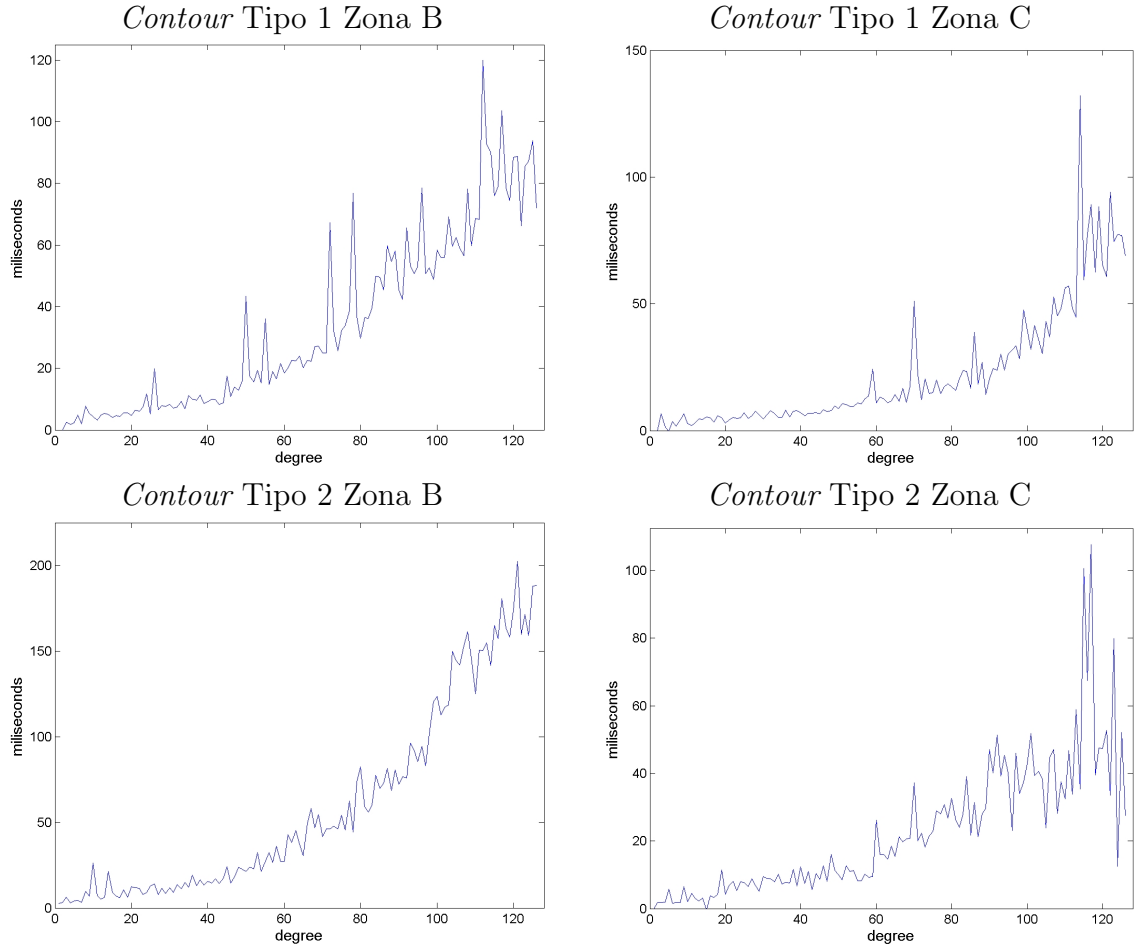


Figura 4.16: Resultados de tiempo para *Contour* en las zonas B y C (tipos 1 y 2) en PC.

método para encontrar todas las raíces, en áreas más pequeñas esta calificación se invierte, como muestran las figuras 4.16 y 4.17.

El método *Contour* es más rápido que Newton para encontrar raíces en áreas restringidas. En una aproximación de grano grueso, se puede suponer que los cálculos requeridos por *Contour* son proporcionales al área cubierta. Los tiempos más bajos mostrados en las figuras 4.16 y 4.17 para la zona C reflejan el hecho de que es más pequeña que la zona B. Como la zona A tiene un tamaño de cuatro unidades, y la zona B de $\pi\dfrac{1^2 - 0'95^2}{2} \approx 0'15$ unidades, la mejora en tiempo es de aproximadamente un orden de magnitud. Esta estimación grosera se basa en el área, que sólo para polinomios de tipo 2 y 3 es proporcional a la cantidad de

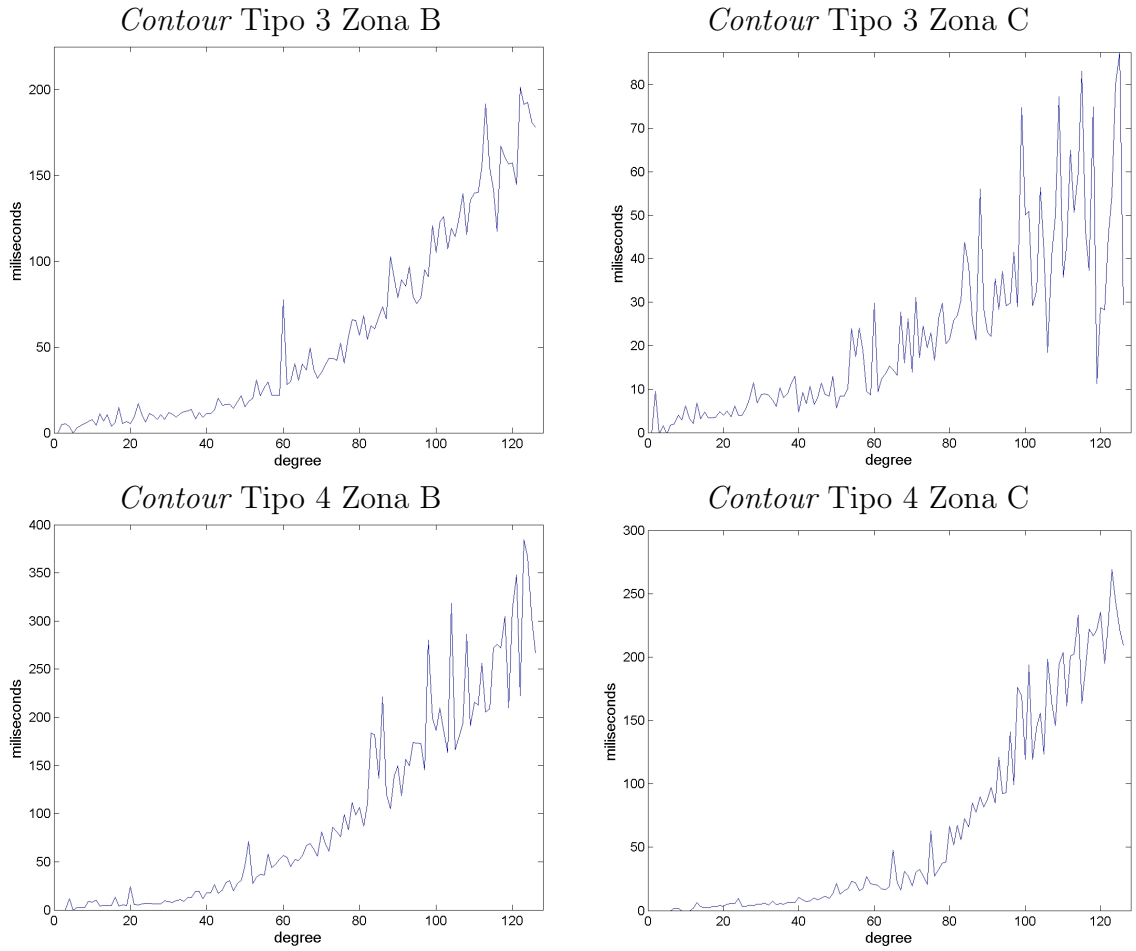


Figura 4.17: Resultados de tiempo para *Contour* en las zonas B y C (tipos 3 y 4) en PC.

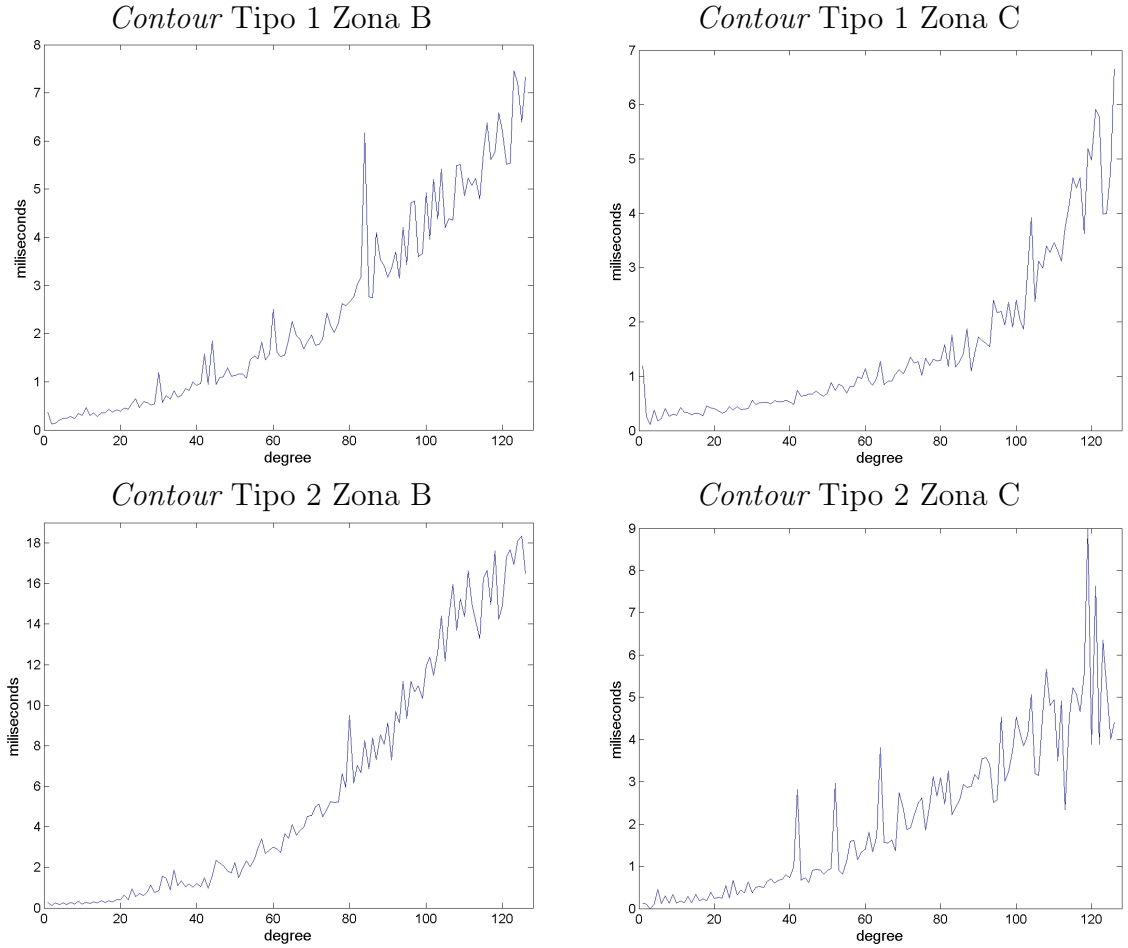


Figura 4.18: Resultados de tiempo para *Contour* en las zonas B y C (tipos 1 y 2) en DSK.

raíces. Para polinomios de tipo 1, las raíces tienden a rellenar el espacio anular, y por tanto el coste computacional medido es mayor que esta estimación. Notemos que para el tipo 4 (LPC) la gráfica aumenta su pendiente a partir de los grados 75-80, aproximadamente. Esto es debido a que a partir de estos grados hay un mayor número de raíces en la zona C de las que le corresponderían teniendo en cuenta solo su área.

Las medidas tomadas en el entorno DSK son consistentes con lo anterior, y se muestran en las figuras 4.18 y 4.19.

Los otros dos métodos bajo estudio muestran tiempos más largos, y mostramos las gráficas con los segundos como unidades de las ordenadas. Durand-Kerner no

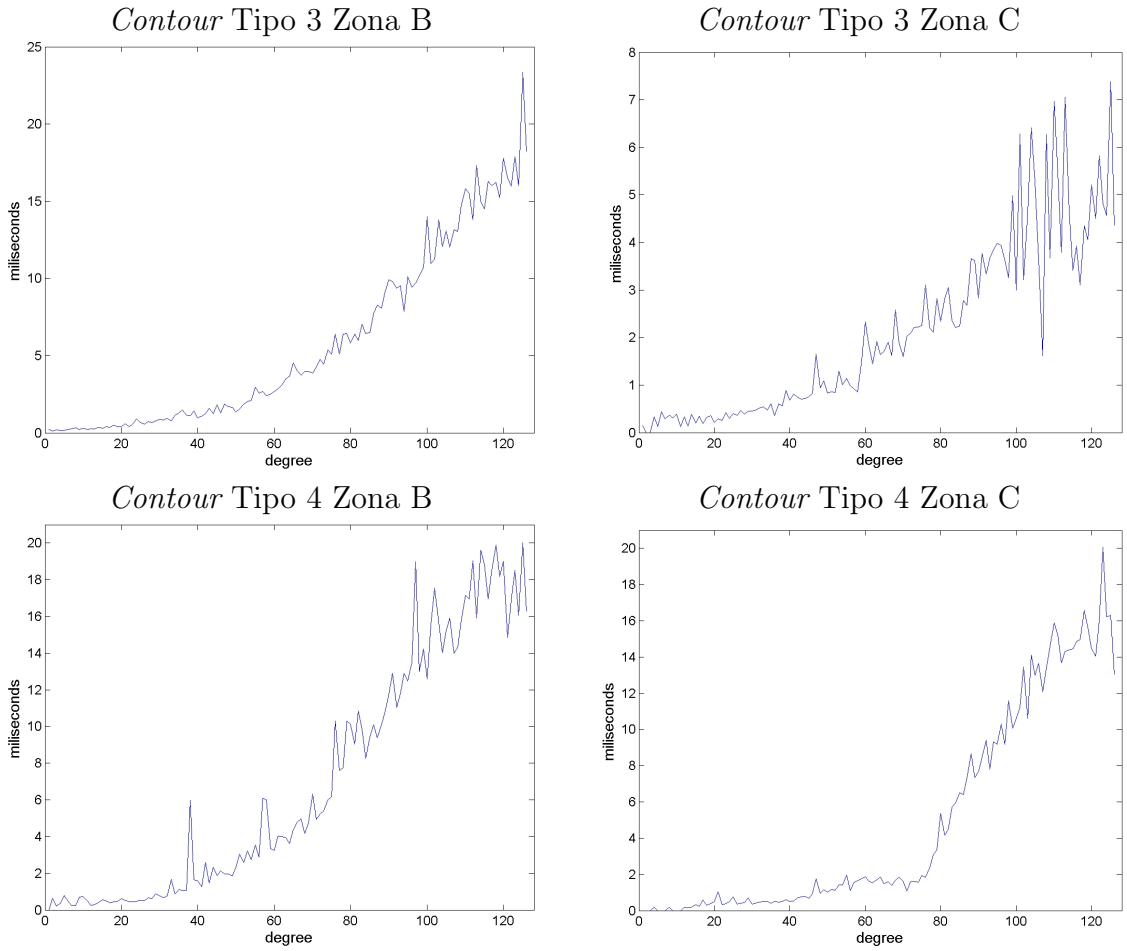


Figura 4.19: Resultados de tiempo para *Contour* en las zonas B y C (tipos 3 y 4) en DSK.

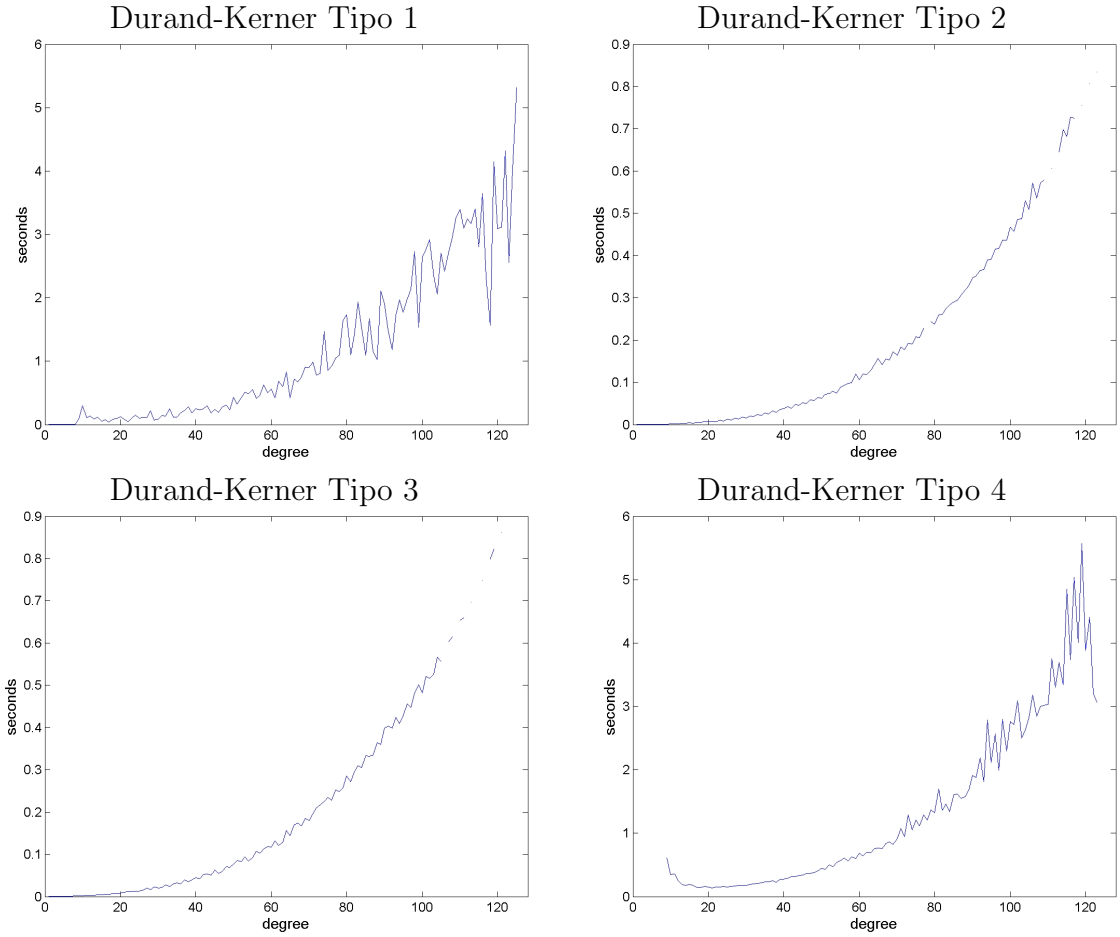


Figura 4.20: Resultados de tiempo para Durand-Kerner en todos los tipos polinómicos, en PC.

converge para los polinomios de prueba con grado por encima de 110 de tipo 2 y 3, tal vez porque aparecen ciclos pseudoestables [Kyurkchiev, 1998]. En los polinomios de tipo 1 y 4 este método converge, pero es relativamente más lento que en los otros métodos (figura 4.20).

En el entorno DSK, el método Durand-Kerner muestra problemas de convergencia en grados por encima de 90 (60 en polinomios de tipo 3, figura 4.21). Esto es coherente con las cifras en PC.

Las cifras de Lehmer-Schur en la zona A crecen mucho, alcanzando varios minutos, y no se muestran. Este crecimiento, mayor que cuadrático, se debe a la superposición de los círculos en los que se divide la zona de búsqueda, multiplicando

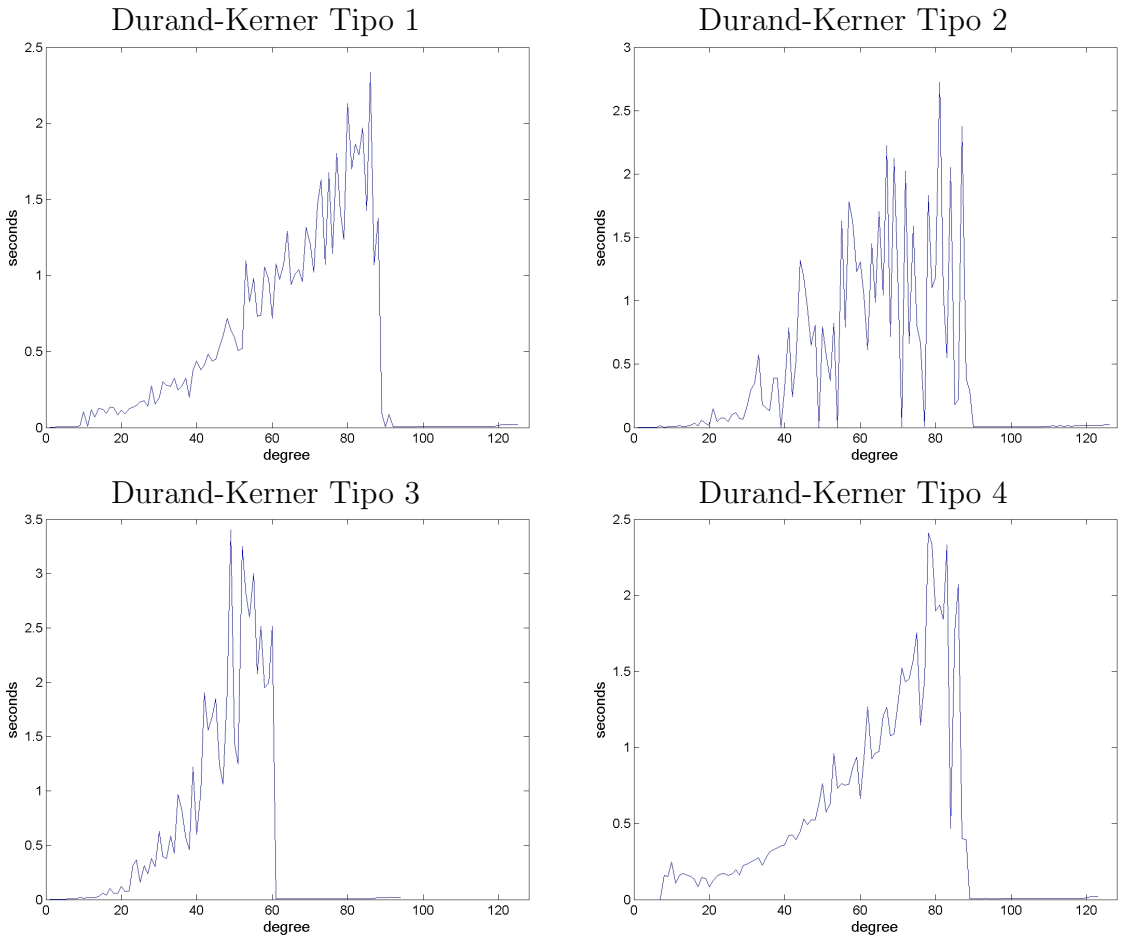


Figura 4.21: Resultados de tiempo para Durand-Kerner en todos los tipos polinómicos, en DSK.

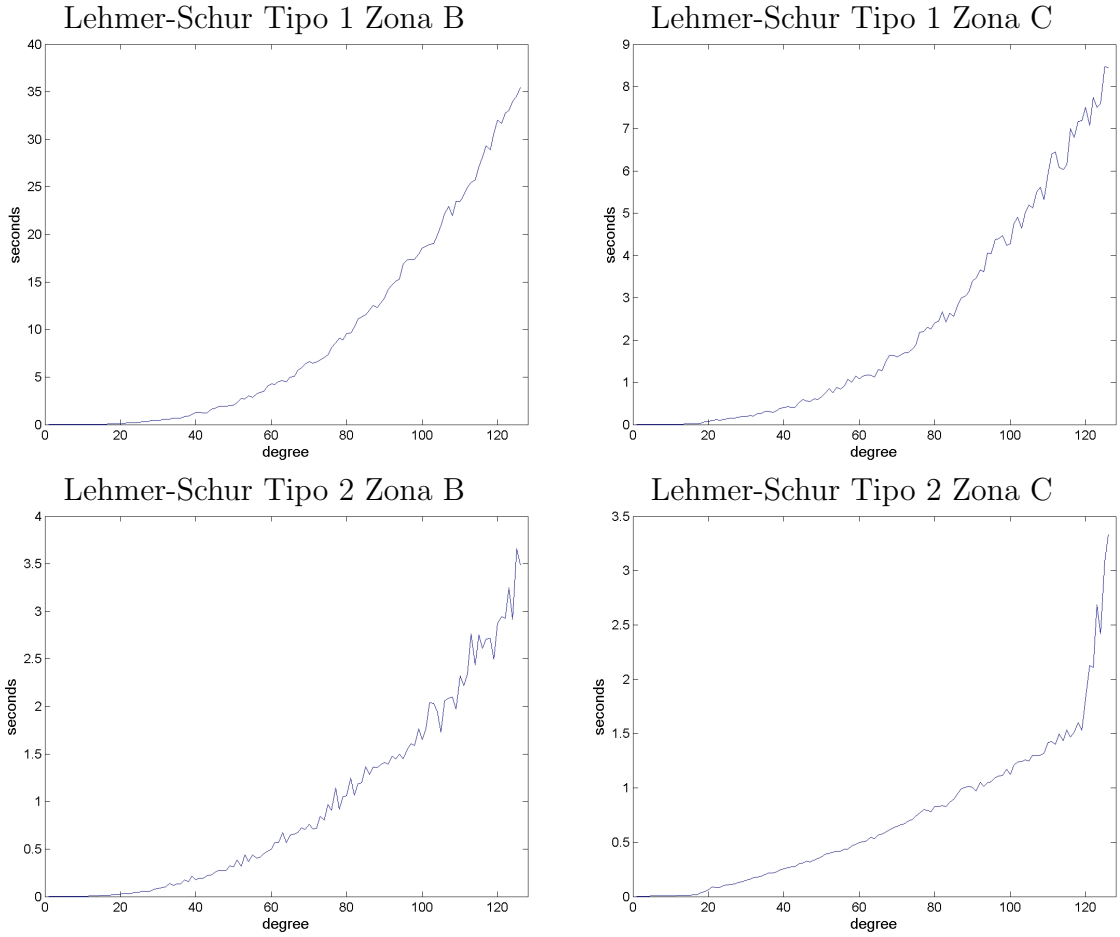


Figura 4.22: Resultados de tiempo para Lehmer-Schur en las zonas B y C (tipos 1 y 2) en PC.

el número de veces que se encuentra cada raíz, como se comentó anteriormente. Esta situación es menos frecuente en las zonas B y C, como muestra la figura 4.22.

Notemos que, para los polinomios de tipo 2 y 3, la probabilidad de que la zona C contenga alguna raíz es baja. Así que el tiempo empleado es debido a la aplicación del método solo a los primeros círculos que definen esta zona (figura 4.5). Sólo a partir de grado 120 aproximadamente los polinomios tienen alguna raíz en la zona C. Esto no ocurre en los tipos 1 y 4, como sugieren la figuras 4.22 y 4.23.

En cualquier caso, el método de Lehmer-Schur muestra un pobre rendimiento, debido a la superposición de los círculos de partición. La deflacción de las raíces

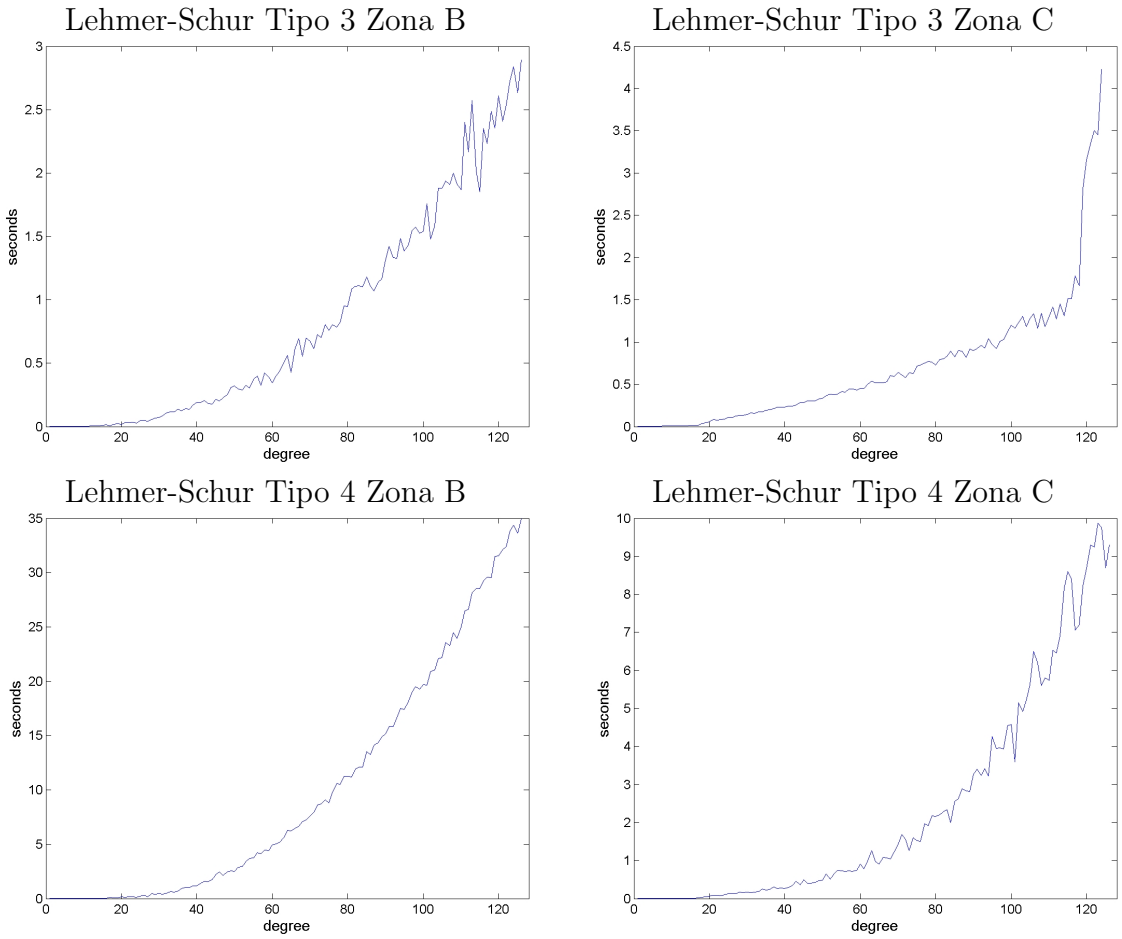


Figura 4.23: Resultados de tiempo para Lehmer-Schur en las zonas B y C (tipos 3 y 4) en PC.

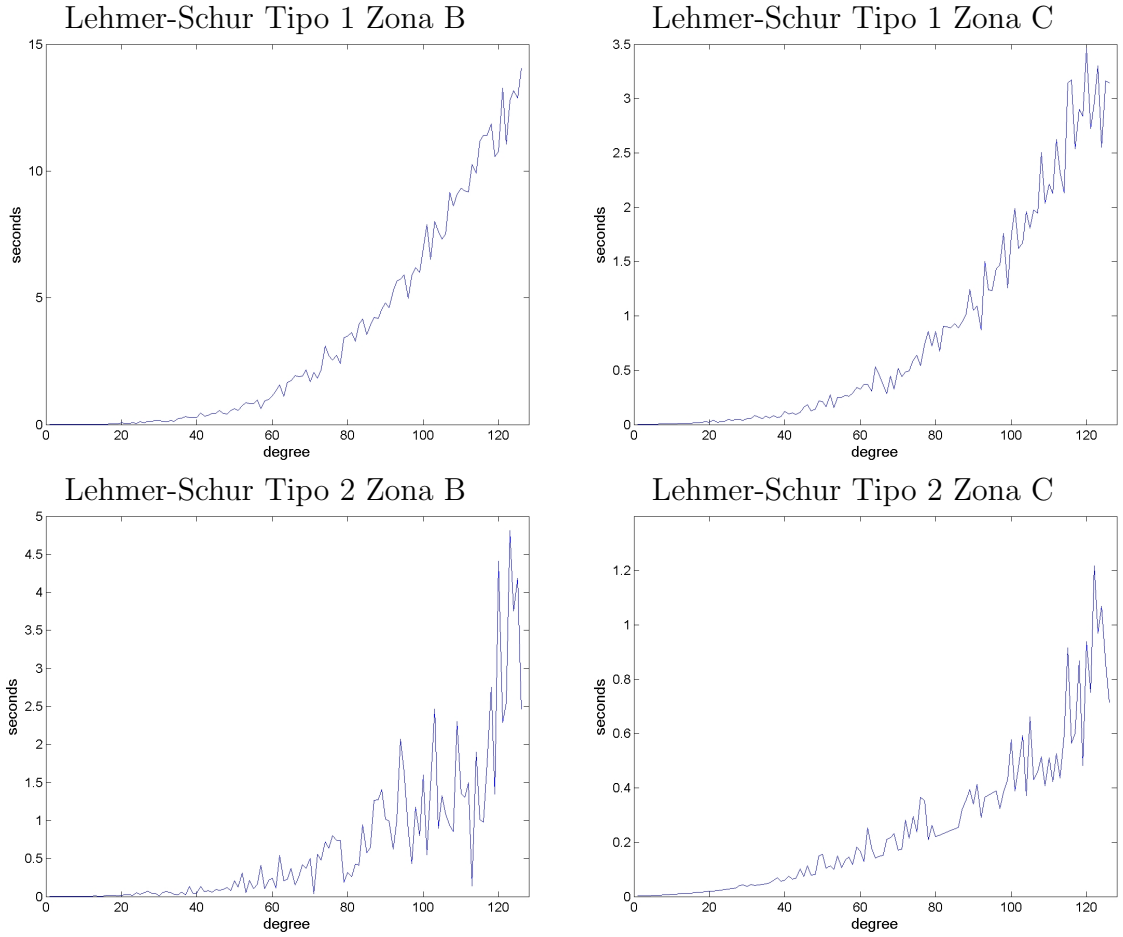


Figura 4.24: Resultados de tiempo para Lehmer-Schur en las zonas B y C (tipos 1 y 2) en DSK.

encontradas, como en el método de Newton, puede resolver el problema de encontrar repetidamente la misma raíz, pero esto requiere cálculos de alta precisión. Hacer la comparativa con estas mejoras se sale del objetivo de este trabajo.

Al igual que en el caso de PC, las cifras de Lehmer-Schur en DSK en los tipos 2 y 3, zona C muestran una línea de base, con el coste de aplicación del método de los círculos iniciales, y sobre esta línea alguna raíz ocasional añade un pico en grados por encima de 100 aproximadamente (figuras 4.24 y 4.25). En los tipos 1 y 4 este fenómeno no ocurre debido a su distribución de raíces.

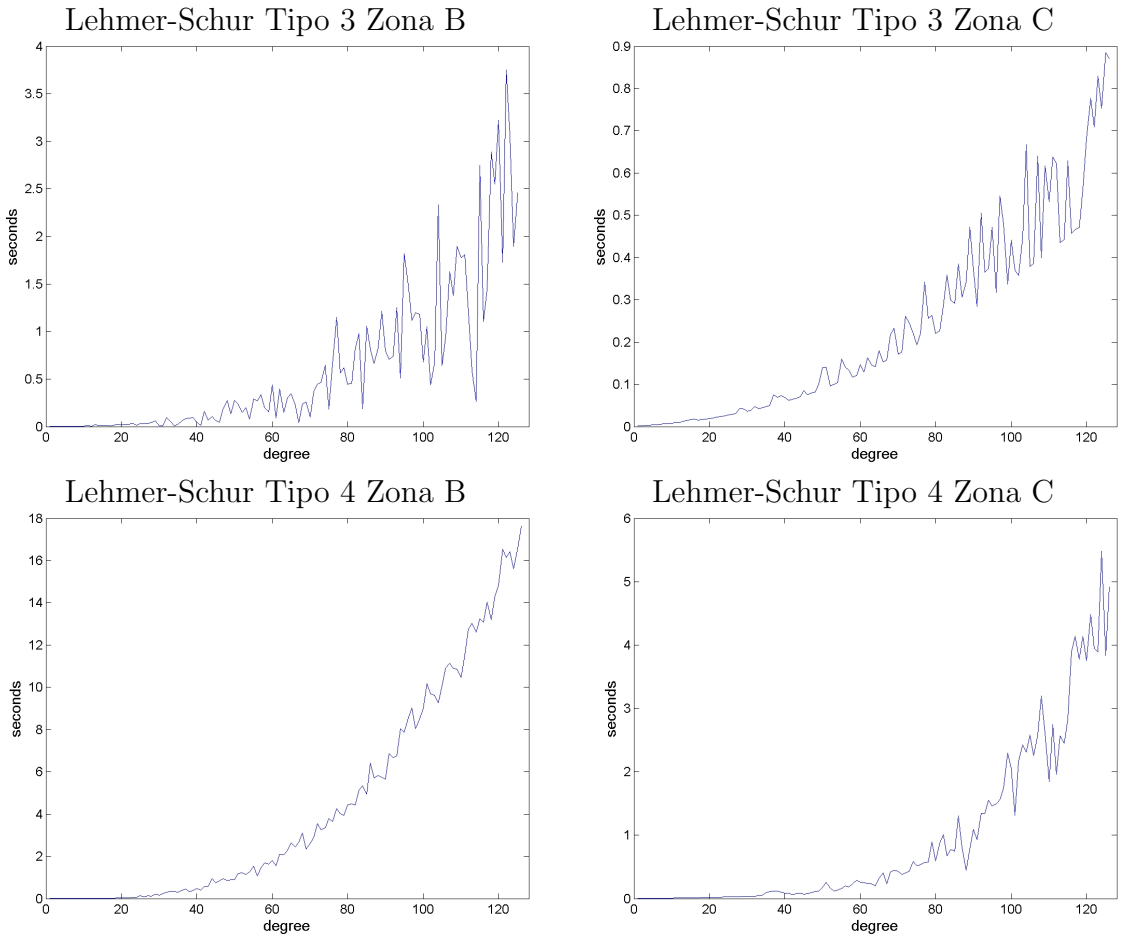


Figura 4.25: Resultados de tiempo para Lehmer-Schur en las zonas B y C (tipos 3 y 4) en DSK.

4.4.1. Conclusiones y trabajo futuro

Llegamos a la conclusión de que el uso del método *Contour* es ventajoso sobre el de Newton si el área de interés contiene una fracción de todas las raíces (aproximadamente un décimo del total). Esta conclusión se aplica a los cuatro tipos de polinomios probados. El área debe estar definida previamente, y los polinomios derivados de LPC y otras técnicas de modelado de señal tienen asociada un área de este tipo, que contiene las raíces deseadas. La presencia de *clusters* de raíces no afecta a la eficacia del método *Contour,* en contraste con el método de Newton.

Para la tarea de encontrar todas las raíces, el método de Durand-Kerner es preferible sobre el de Newton en polinomios con *clusters*, por lo menos hasta el grado 100, donde la convergencia del primer método comienza a fallar. Para los otros tipos de polinomio Newton es la mejor opción.

Para encontrar raíces en zonas seleccionadas, *Contour* es preferible sobre el otro método geométrico considerado, ya que utiliza una descomposición de la zona de interés en subáreas disjuntas, evitando por lo tanto encontrar repetidamente la misma raíz.

El método *Contour* es intrínsecamente paralelo, porque las subdivisiones de la zona de interés se pueden asignar a diferentes procesadores, aunque este aspecto no se ha puesto a prueba en este trabajo.

Para ampliar este trabajo comparativo, se debería probar *Contour* contra otros métodos específicos para raíces de módulo máximo. Estos métodos se aplican frecuentemente en proceso de señal (como el método de Bernouilli, o el de cociente-diferencia QD [Pan, 1997]). Además, la batería de pruebas se puede ampliar con los conjuntos estándar de polinomios de [Jenkins and Traub, 1975] y [Zeng, 2005], y con polinomios utilizados en la modelización de sistemas complejos (como el diseño Schelkunoff en teoría de antenas [Orchard et al., 1985] o el problema cinemático inverso en robótica [Craig, 2005], [Sommese and Wampler, 2005]). Por otra parte, aumentar el grado del polinomio por encima de 128 requiere aritmética de precisión múltiple, lo que se sale de lo usual en métodos numéricos de propósito general.

Bibliografía

[Aberth, 1973] Aberth, O. (1973). Iteration methods for finding all zeros of a polynomial simultaneously. *Mathematics of computation*, 27(122):339–344.

[Aho et al., 1983] Aho, A., Hopcroft, J., and Ullman, J. (1983). *Data structures and algorithms:*. Addison-Wesley series in computer science and information processing. Addison-Wesley.

[Anderson et al., 1999] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition.

[Atal and Hanauer, 1971] Atal, B. S. and Hanauer, S. L. (1971). Speech analysis and synthesis by linear prediction of the speech wave. *The journal of the acoustical society of America*, 50(2B):637–655.

[Ball, 1997] Ball, K. (1997). An elementary introduction to modern convex geometry. In *Flavors of geometry*, volume 31 of *Math. Sci. Res. Inst. Publ.*, pages 1–58. Cambridge Univ. Press, Cambridge.

[Bermúdez et al., 2000] Bermúdez, J., Sancho, J., and Vilda, P. (2000). *Reconocimiento de Voz y Fonética Acústica*. Ra-Ma.

[Bini and Fiorentino, 2000] Bini, D. and Fiorentino, G. (2000). Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numerical Algorithms*, 23:127–173. 10.1023/A:1019199917103.

[Bini et al., 2004a] Bini, D., Gemignani, L., and Pan, V. (2004a). Inverse power and durand-kerner iterations for univariate polynomial root-finding. *Computers & Mathematics with Applications*, 47(2-3):447 – 459.

[Bini and Pan, 1994] Bini, D. and Pan, V. Y. (1994). *Polynomial and matrix computations. Vol. 1.* Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA. Fundamental algorithms.

[Bini and Pan, 1996] Bini, D. and Pan, V. Y. (1996). Graeffe's, chebyshev-like, and cardinal's processes for splitting a polynomial into factors. *Journal of Complexity*, 12(4):492–511.

[Bini et al., 2004b] Bini, D. A., Gemignani, L., Victor, and Pan, Y. (2004b). Improved initialization of the accelerated and robust qr-like polynomial root-finding. *Electron. Trans. Numer. Anal*, 17:2004.

[Blum et al., 1998] Blum, L., Cucker, F., Shub, M., and Smale, S. (1998). *Complexity and real computation*. Springer-Verlag, New York. With a foreword by Richard M. Karp.

[Boersma, 2002] Boersma, P. (2002). Praat, a system for doing phonetics by computer. *Glot international*, 5(9/10):341–345.

[Box et al., 2008] Box, G., Jenkins, G., and Reinsel, G. (2008). *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics. John Wiley & Sons.

[Brassard and Bratley, 1988] Brassard, G. and Bratley, P. (1988). *Algorithmics - theory and practice*. Prentice Hall.

[Burr and Krahmer, 2012] Burr, M. A. and Krahmer, F. (2012). Sqfreeeval: an (almost) optimal real-root isolation algorithm. *Journal of Symbolic Computation*, 47(2):153–166.

[Burstall, 1969] Burstall, R. (1969). Proving properties of programs by structural induction. *The Computer Journal*, 12(1):41–48.

[Cardinal, 1996] Cardinal, J.-P. (1996). On two iterative methods for approximating the roots of a polynomial. *LECTURES IN APPLIED MATHEMATICS-AMERICAN MATHEMATICAL SOCIETY*, 32:165–188.

[Chou and Ko, 1995] Chou, A. W. and Ko, K.-I. (1995). Computational complexity of two-dimensional regions. *SIAM J. Comput.*, 24(5):923–947.

[Collins, 1977] Collins, G. E. (1977). Infallible calculation of polynomial zeros to specified precision. In *Mathematical software, III (Proc. Sympos., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1977)*, pages 35–68. Publ. Math. Res. Center, No. 39. Academic Press, New York.

[Cormen et al., 2001] Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). *Introduction To Algorithms*. MIT Press.

[Cortés-Fácila et al., 2014] Cortés-Fácila, A., García-Zapata, J., and Díaz-Martín, J. (2014). Contour: a zero-finding method for high-degree polynomials.

[Craig, 2005] Craig, J. (2005). *Introduction to robotics: mechanics and control*. Addison-Wesley series in electrical and computer engineering: control engineering. Pearson/Prentice Hall.

[Dedieu and Yakoubsohn, 1993] Dedieu, J.-P. and Yakoubsohn, J.-C. (1993). Computing the real roots of a polynomial by the exclusion algorithm. *Numerical Algorithms*, 4(1):1–24.

[Deller et al., 2000] Deller, J., Hansen, J., and Proakis, J. (2000). *Discrete-Time Processing of Speech Signals*. An IEEE Press classic reissue. Wiley.

[Delves and Lyness, 1967] Delves, L. and Lyness, J. (1967). A numerical method for locating the zeros of an analytic function. *Mathematics of computation*, 21(100):543–560.

[Demmel, 1997] Demmel, J. (1997). *Applied Numerical Linear Algebra*. Miscellaneous Bks. Society for Industrial and Applied Mathematics.

[Demmel, 1987] Demmel, J. W. (1987). On condition numbers and the distance to the nearest ill-posed problem. *Numer. Math.*, 51(3):251–289.

[Diaz Martin et al., 2001] Diaz Martin, J., Rodriguez Garcia, J., Garcia Zapata, J., and Gomez Vilda, P. (2001). Robust voice recognition as a distributed service. In *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, volume 2, pages 571 –575 vol.2.

[Dickenstein and Emiris, 2005] Dickenstein, A. and Emiris, I. Z. (2005). *Solving polynomial equations: Foundations, algorithms, and applications*, volume 14. Springer.

[Do Carmo, 1976] Do Carmo, M. P. (1976). *Differential geometry of curves and surfaces*, volume 2. Prentice-Hall Englewood Cliffs.

[Duncan and Jack, 1988] Duncan, G. and Jack, M. (Feb. 1988). Formant estimation algorithm based on pole focusing offering improved noise tolerante and feature resolution. In *IEE Proceedings, vol. 135, Pt. F, no. 1, pp 18-32,*.

[Edelman and Murakami, 1995] Edelman, A. and Murakami, H. (1995). Polynomial roots from companion matrix eigenvalues. *Mathematics of Computation*, 64(210):763–776.

[Elkadi and Mourrain, 2005] Elkadi, M. and Mourrain, B. (2005). Symbolic-numeric methods for solving polynomial equations and applications. In *Solving polynomial equations*, pages 125–168. Springer.

[Emiris et al., 2010] Emiris, I. Z., Pan, V. Y., and Tsigaridas, E. P. (2010). *Algebraic and numerical algorithms*. Chapman & Hall/CRC.

[Farmer and Loizou, 1975] Farmer, M. and Loizou, G. (1975). A class of iteration functions for improving, simultaneously, approximations to the zeros of a polynomial. *BIT Numerical Mathematics*, 15(3):250–258.

[Flanagan, 1960] Flanagan, J. (1960). Spectrum segmentation system for the automatic extraction of formant frequencies from human speech. US Patent 2,938,079.

[Fog, 2014] Fog, A. (2014). Lists of instruction latencies, throughputs and micro-operation breakdowns for Intel, AMD and VIA CPUs. Technical report, Technical University of Denmark.

[Forster, 1992] Forster, W. (1992). Some computational methods for systems of nonlinear equations and systems of polynomial equations. *J. Global Optim.*, 2(4):317–356.

[Fortune, 2002] Fortune, S. (2002). An iterated eigenvalue algorithm for approximating roots of univariate polynomials. *J. Symbolic Comput.*, 33(5):627–646. Computer algebra (London, ON, 2001).

[Galassi and Gough, 2009] Galassi, M. and Gough, B. (2009). *GNU Scientific Library: Reference Manual*. GNU manual. Network Theory Limited.

[García Zapata and Díaz Martín, 2008] García Zapata, J. and Díaz Martín, J. (2008). A geometrical root finding method for polynomials, with complexity analysis. Technical report, Departamento de Matemáticas, Universidad de Extremadura.

[García Zapata and Díaz Martín, 2012] García Zapata, J. and Díaz Martín, J. (2012). A geometrical root finding method for polynomials, with complexity analysis. *Journal of Complexity*, 28:320–345.

[García Zapata et al., 2004a] García Zapata, J., Martín, D., and Gómez Vilda, P. (2004a). Fast formant estimation by complex analysis of lpc coefficients. In *EUSIPCO 2004 12th European Signal Processing Conference*, pages 2015–2018, Wien.

[García Zapata et al., 2004b] García Zapata, J.-L., Díaz Martín, J., and Gómez Vilda, P. (2004b). Parallel root-finding method for lpc analysis of speech. In Sojka, P., Kopecek, I., and Pala, K., editors, *Text, Speech and Dialogue*, volume 3206 of *Lecture Notes in Computer Science*, pages 529–536. Springer Berlin / Heidelberg. 10.1007/978-3-540-30120-2.

[García Zapata and Díaz Martín, 2014] García Zapata, J. L. and Díaz Martín, J. C. (2014). Finding the number of roots of a polynomial in a plane region using the winding number. *Computers & Mathematics with Applications*, 67(3):555 – 568.

[Garner and Holmes, 1998] Garner, P. N. and Holmes, W. J. (May 1998). On the robust incorporation of formant features into hidden markov models for automatic speech recognition. In *Proc. ICASSṔ98, Seattle, Washington, USA,*.

[Gómez, 1998] Gómez, P., e. a. (1998). A dsp-based modular architecture for noise cancellation and speech recognition. In *Proc. of the 1998 IEEE Int. Symp. on Circuits and Systems, ISCAŚ98, Monterey, CA, USA, pp. V.178-181.*

[Goldberg, 1991a] Goldberg, D. (1991a). What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23(1):5–48.

[Goldberg, 1991b] Goldberg, D. (1991b). What every computer scientist should know about floating-point arithmetic. *ACM Comput. Surv.*, 23(1):5–48.

[Golub and Van Loan, 1996] Golub, G. and Van Loan, C. (1996). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press.

[Gourdon, 1993] Gourdon, X. (1993). Algorithmique du theoreme fondamental de l'algebre. Technical report, Institut national de recherche en informatique et en automatique (INRIA), Unité de recherche de Rocquencourt.

[Haykin, 1995] Haykin, S. (1995). *Adaptive Filter Theory (3rd Edition)*. Prentice Hall, 3rd edition.

[Henrici, 1988] Henrici, P. (1988). *Applied and computational complex analysis. Vol. 1.* Wiley Classics Library. John Wiley & Sons Inc., New York. Power series—integration—conformal mapping—location of zeros, Reprint of the 1974 original, A Wiley-Interscience Publication.

[Henrici and Gargantini, 1969] Henrici, P. and Gargantini, I. (1969). Uniformly convergent algorithms for the simultaneous approximation of all zeros of a polynomial. In *Constructive Aspects of the Fundamental Theorem of Algebra (Proc. Sympos., Zürich-Rüschlikon, 1967)*, pages 77–113.

[Herlocker and Ely, 1995] Herlocker, J. and Ely, J. (1995). An automatic and guaranteed determination of the number of roots of an analytic function interior to a simple closed curve in the complex plane. *Reliable Computing*, 1(3):239–249.

[Higham, 2002] Higham, N. J. (2002). *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition.

[Holmes and Russell, 1999] Holmes, W. J. and Russell, M. J. (Jan. 1999.). Probabilistic-trajectory segmental hmms. *Computer Speech & Language*, vol. 13, no. 1:3–37.

[Hopcroft et al., 2000] Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2000). *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2nd edition.

[Householder, 1970] Householder, A. S. (1970). *The numerical treatment of a single nonlinear equation*. McGraw-Hill New York.

[Huang, 2004] Huang, D.-S. (2004). A constructive approach for finding arbitrary roots of polynomials by neural networks. *Neural Networks, IEEE Transactions on*, 15(2):477–491.

[IEEE, 2008] IEEE (2008). IEEE standard for floating-point arithmetic. *IEEE Std 754-2008*, pages 1–70.

[Jenkins and Traub, 1975] Jenkins, M. A. and Traub, J. F. (1975). Principles for testing polynomial zerofinding programs. *ACM Trans. Math. Softw.*, 1(1):26–34.

[Kalantari, 2009] Kalantari, B. (2009). *Polynomial root-finding and polynomiography*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ.

[Kamath, 2010] Kamath, N. (2010). *Subdivision algorithms for complex root isolation: Empirical comparisons*. PhD thesis, University of Oxford.

[Kang and Coulter, 1976] Kang, G. and Coulter, D. (1976). 600 bps voice digitizer. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '76.*, volume 1, pages 91 –94.

[Kapilow et al., 1999] Kapilow, D. A., Stylianou, Y., and Schroeter, J. (1999). Detection of non-stationarity in speech signals and its application to time-scaling. In *EUROSPEECH*.

[Kawahara et al., 1999] Kawahara, H., Masuda-Katsuse, I., and de Cheveigné, A. (1999). Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, 27(3-4):187 – 207.

[Kirrinnis, 1998] Kirrinnis, P. (1998). Partial fraction decomposition in c (¡i¿z¡/i¿) and simultaneous newton iteration for factorization in c [¡i¿z¡/i¿]. *journal of complexity*, 14(3):378–444.

[Knuth, 1981] Knuth, D. E. (1981). *The art of computer programming. Vol. 2.* Addison-Wesley Publishing Co., Reading, Mass., second edition. Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

[Knuth, 1992] Knuth, D. E. (1992). *Axioms and hulls.* springer-Verlag Berlin.

[Ko et al., 2008] Ko, K. H., Sakkalis, T., and Patrikalakis, N. M. (2008). A reliable algorithm for computing the topological degree of a mapping in R^2. *Appl. Math. Comput.*, 196(2):666–678.

[Kolmogorov and Fomın, 1975] Kolmogorov, A. N. and Fomın, S. V. (1975). *Introductory real analysis.* Dover Publications Inc., New York. Translated from the second Russian edition and edited by Richard A. Silverman, Corrected reprinting.

[Kopec, 1986] Kopec, G. (1986). Formant tracking using hidden markov models and vector quantization. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(4):709 – 729.

[Kravanja and Van Barel, 2000] Kravanja, P. and Van Barel, M. (2000). *Computing the Zeros of Analytic Functions*, volume 1727 of *Lecture Notes in Mathematics*. Springer.

[Kyurkchiev, 1998] Kyurkchiev, N. (1998). *Initial approximations and root finding methods.* Mathematical research. Wiley-VCH.

[Lang and Frenzel, 1994] Lang, M. and Frenzel, B.-C. (1994). Polynomial root finding. *Signal Processing Letters, IEEE*, 1(10):141–143.

[Lay, 2007] Lay, S. R. (2007). *Convex sets and their applications*. Courier Dover Publications.

[Lehmer, 1961] Lehmer, D. H. (1961). A machine method for solving polynomial equations. *J. ACM*, 8:151–162.

[Lenstra Jr, 1999] Lenstra Jr, H. W. (1999). Finding small degree factors of lacunary polynomials. *Number theory in progress*, 1:267–276.

[Loewenthal, 1993] Loewenthal, D. (1993). Improvements on the lehmer-schur root detection method. *Journal of Computational Physics*, 109(2):164 – 168.

[Malajovich and Zubelli, 2001] Malajovich, G. and Zubelli, J. P. (2001). On the geometry of graeffe iteration. *Journal of Complexity*, 17(3):541–573.

[Marden, 1966] Marden, M. (1966). *Geometry of polynomials*, volume 1. AMS Bookstore.

[Markel, 1976] Markel, J. D., G. A. (1976.). *Linear Prediction of Speech*. Springer-Verlag, Berlin.

[McCandless, 1974] McCandless, S. (Apr. 1974). An algorithm for automatic formant extraction using linear prediction spectra. *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-22, no. 2:135–141.

[McCarthy, 2011] McCarthy, J. M. (2011). Kinematics, polynomials, and computers-a brief history. *Journal of Mechanisms and Robotics*, 3(1):010201.

[Mekwi, 2001] Mekwi, W. R. (2001). *Iterative methods for roots of polynomials*. PhD thesis, University of Oxford.

[Morris and Clements, 2002] Morris, R. W. and Clements, M. A. (2002). Reconstruction of speech from whispers. *Medical Engineering & Physics*, 24(7-8):515 – 520. ¡ce:title¿Models & Analysis of Vocal Emissions¡/ce:title¿.

[Neff, 1994] Neff, C. A. (1994). Specified precision polynomial root isolation is in nc. *Journal of Computer and System Sciences*, 48(3):429–463.

[Neff and Reif, 1996a] Neff, C. A. and Reif, J. H. (1996a). An efficient algorithm for the complex roots problem. *Journal of Complexity*, 12(2):81–115.

[Neff and Reif, 1996b] Neff, C. A. and Reif, J. H. (1996b). An efficient algorithm for the complex roots problem. *J. Complexity*, 12(2):81–115.

[Noureddine and Fellah, 2005] Noureddine, S. and Fellah, A. (2005). Pinpointing the real zeros of analytic functions. In *High Performance Computational Science and Engineering*, pages 123–142. Springer.

[Ogata, 2010] Ogata, K. (2010). *Modern Control Engineering*. Instrumentation and controls series. Prentice Hall.

[Olive, 1971] Olive, J. P. (1971). Automatic formant tracking by a newton-rapshon technique. *Journal of the Acoustic Society of America*, vol. 50, no. 2:661–670,.

[Olive, 1992] Olive, J. P. (1992). Mixed spectral representation—formants and linear predictive coding. *The Journal of the Acoustical Society of America*, 92(4):1837–1840.

[Oppenheim et al., 1996] Oppenheim, A. V., Willsky, A. S., and Hamid, W. S. (1996). *Signals and Systems (International Edition)*. Pearson Education, 2 edition.

[Orchard et al., 1985] Orchard, H., Elliott, R., and Stern, G. (1985). Optimising the synthesis of shaped beam antenna patterns. *Microwaves, Antennas and Propagation, IEE Proceedings H*, 132(1):63 –68.

[Orfanidis and Vail, 1986] Orfanidis, S. and Vail, L. (1986). Zero-tracking adaptive filters. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(6):1566 – 1572.

[Ouaaline and Radouane, 1998] Ouaaline, N. and Radouane, L. (1998). Pole-zero estimation of speech signal based on zero-tracking algorithm. *International Journal of Adaptive Control and Signal Processing*, 12(1):1–12.

[Padberg, 1999] Padberg, M. (1999). *Linear Optimization and Extensions*, volume 12. Springer.

[Pan, 1987] Pan, V. (1987). Sequential and parallel complexity of approximate evaluation of polynomial zeros. *Computers & Mathematics with Applications*, 14(8):591–622.

[Pan, 1996a] Pan, V. Y. (1996a). Optimal and nearly optimal algorithms for approximating polynomial zeros. *Computers & Mathematics with Applications*, 31(12):97–138.

[Pan, 1996b] Pan, V. Y. (1996b). Optimal and nearly optimal algorithms for approximating polynomial zeros. *Comput. Math. Appl.*, 31(12):97–138.

[Pan, 1997] Pan, V. Y. (1997). Solving a polynomial equation: some history and recent progress. *SIAM Rev.*, 39(2):187–220.

[Pan, 2001a] Pan, V. Y. (2001a). Univariate polynomials: Nearly optimal algorithms for numerical factorization and rootfinding. *J. Symbolic Computation*, 33:2002.

[Pan, 2001b] Pan, V. Y. (2001b). Univariate polynomials: Nearly optimal algorithms for numerical factorization and rootfinding. *J. Symbolic Computation*, 33:2002.

[Pan, 2012] Pan, V. Y. (2012). Root-refining for a polynomial equation. In *Computer Algebra in Scientific Computing*, pages 283–293. Springer.

[Parsons, 1987] Parsons, T. (1987). *Voice and speech processing*. MCGRAW HILL SERIES IN ELECTRICAL AND COMPUTER ENGINEERING. McGraw-Hill.

[Press et al., 1992] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in C*. Cambridge University Press, Cambridge, second edition. The art of scientific computing.

[Priestley, 1981] Priestley, M. (1981). *Spectral analysis and time series*. Number v. 1-2 in Probability and mathematical statistics. Academic Press.

[Rabiner, 1999] Rabiner, L., J. B. (1999). *Fundamentals of Speech Recognition*. Prentice-Hall, New Jersey.

[Ralston and Rabinowitz, 1978a] Ralston, A. and Rabinowitz, P. (1978a). *A first course in numerical analysis*. McGraw-Hill Book Co., New York, second edition. International Series in Pure and Applied Mathematics.

[Ralston and Rabinowitz, 1978b] Ralston, A. and Rabinowitz, P. (1978b). *A first course in numerical analysis*. International series in pure and applied mathematics. McGraw-Hill.

[Renegar, 1987] Renegar, J. (1987). On the worst-case arithmetic complexity of approximating zeros of polynomials. *J. Complexity*, 3(2):90–113.

[Riley, 1989] Riley, M. (1989). *Speech Time-Frequency Representation*. Kluwer International Series in Engineering and Computer Science. Springer.

[Rudin, 1987] Rudin, W. (1987). *Real and complex analysis*. McGraw-Hill Book Co., New York, third edition.

[Sakurai et al., 2003] Sakurai, T., Kravanja, P., Sugiura, H., and Van Barel, M. (2003). An error analysis of two related quadrature methods for computing zeros of analytic functions. *Journal of computational and applied mathematics*, 152(1):467–480.

[Schechter, 1996] Schechter, E. (1996). *Handbook of Analysis and Its Foundations*. Elsevier Science.

[Schmid and Barnard, 1995] Schmid, P. and Barnard, E. (Sept 1995). Robust, n-best formant tracking. In *Proc. EUROSPEECH?95, Madrid, Spain*, pages pp. 737–740.

[Schönhage, 1982] Schönhage, A. (1982). The fundamental theorem of algebra in terms of computational complexity. Technical report, Mathematisches Institut Universität Tübingen.

[Sitton et al., 2003] Sitton, G., Burrus, C., Fox, J., and Treitel, S. (2003). Factoring very-high-degree polynomials. *Signal Processing Magazine, IEEE*, 20(6):27 – 42.

[Smale, 1981] Smale, S. (1981). The fundamental theorem of algebra and complexity theory. *Bulletin of the American Mathematical Society*, 4(1):1–36.

[Smith et al., 1976] Smith, B. T., Boyle, J. M., Dongarra, J. J., Garbow, B. S., Ikebe, Y., Klema, V. C., and Moler, C. B. (1976). *Matrix eigensystem routines—EISPACK guide*. Springer-Verlag, Berlin, second edition. Lecture Notes in Computer Science, Vol. 6.

[Smith, 2003] Smith, S. (2003). *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Demystifying Technology Series. Newnes.

[Snell and Milinazzo, 1993] Snell, R. and Milinazzo, F. (1993). Formant location from lpc analysis data. *Speech and Audio Processing, IEEE Transactions on*, 1(2):129 –134.

[Sommese and Wampler, 2005] Sommese, A. J. and Wampler, C. W. (2005). *The numerical solution of systems of polynomials: Arising in Engineering and Science*. World Scientific.

[Starer, 1990] Starer, D. (May 1990). *Algorithms for Polynomial-Based Signal Processing*. PhD thesis, Yale University.

[Stetter, 1996] Stetter, H. J. (1996). Analysis of zero clusters in multivariate polynomial systems. In *Proceedings of the 1996 international symposium on Symbolic and algebraic computation*, pages 127–136. ACM.

[Suzuki, 2001] Suzuki, Toshio; Suzuki, T. (2001). A globally convergent zero finding method. In *Proceedings of the Third World Congress of Nonlinear Analysts, Part 6 (Catania, 2000)*, volume 47, pages 3869–3875.

[Taylor, 1997] Taylor, J. (1997). *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. A series of books in physics. University Science Books.

[Toh and Trefethen, 1994] Toh, K.-C. and Trefethen, L. N. (1994). Pseudozeros of polynomials and pseudospectra of companion matrices. *Numerische Mathematik*, 68(3):403–425.

[Toselli and Widlund, 2005] Toselli, A. and Widlund, O. B. (2005). *Domain decomposition methods: algorithms and theory*, volume 34. Springer.

[Traub and Woźniakowski, 1979] Traub, J. F. and Woźniakowski, H. (1979). Convergence and complexity of Newton iteration for operator equations. *J. Assoc. Comput. Mach.*, 26(2):250–258.

[Trefethen, 2010] Trefethen, L. N. (2010). *The Princeton companion to mathematics*, chapter IV.21 Numerical Analysis, pages 604–615. Princeton University Press.

[Trefethen and Bau III, 1997] Trefethen, L. N. and Bau III, D. (1997). *Numerical linear algebra*. Number 50. Siam.

[Van Dooren, 1994] Van Dooren, P. (1994). Some numerical challenges in control theory. In *Linear algebra for control theory*, volume 62 of *IMA Vol. Math. Appl.*, pages 177–189. Springer, New York.

[Von Zur Gathen and Gerhard, 2013] Von Zur Gathen, J. and Gerhard, J. (2013). *Modern computer algebra*. Cambridge university press.

[W. Philips, 1992] W. Philips, G. D. J. (April 1992). Data compression of ecg's by high degree polynomial approximation. *IEEE Transactions on biomedical Engineering*, Vol. 39, No. 4:330–337.

[Welling and Ney, 1998] Welling, L. and Ney, H. (Jan. 1998.). Formant estimation for speech recognition. *IEEE Trans. On Speech and Audio Processing*, vol. 6, no. 1:36–48.

[Werner, 1982] Werner, W. (1982). On the simultaneous determination of polynomial roots. In Ansorge, R., Meis, T., and Törnig, W., editors, *Iterative Solution of Nonlinear Systems of Equations*, volume 953 of *Lecture Notes in Mathematics*, pages 188–202. Springer Berlin / Heidelberg. 10.1007/BFb0069383.

[Widrow and Stearns, 1985] Widrow, B. and Stearns, S. (1985). *Adaptive signal processing*. Prentice-Hall signal processing series. Prentice-Hall.

[Wiener, 1975] Wiener, N. (1975). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The M.I.T. Press, Cambridge, Massachussetts.

[Wilf, 1978] Wilf, H. S. (1978). A global bisection algorithm for computing the zeros of polynomials in the complex plane. *J. Assoc. Comput. Mach.*, 25(3):415–420.

[Wilkinson, 1964] Wilkinson, J. (1964). *Rounding errors in algebraic processes*. Prentice-Hall (Englewood Cliffs, NJ).

[Wilkinson, 1965] Wilkinson, J. H. (1965). *The algebraic eigenvalue problem*. Clarendon Press, Oxford.

[Williamson and Shmoys, 2011] Williamson, D. P. and Shmoys, D. B. (2011). *The design of approximation algorithms*. Cambridge University Press.

[Yakoubsohn, 2005] Yakoubsohn, J.-C. (2005). Numerical analysis of a bisection-exclusion method to find zeros of univariate analytic functions. *J. Complexity*, 21(5):652–690.

[Yap and Sagraloff, 2011] Yap, C. K. and Sagraloff, M. (2011). A simple but exact and efficient algorithm for complex root isolation. In *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, pages 353–360. ACM.

[Ying and Katz, 1988] Ying, X. and Katz, I. N. (1988). A reliable argument principle algorithm to find the number of zeros of an analytic function in a bounded domain. *Numer. Math.*, 53(1-2):143–163.

[Young and Gregory, 2012] Young, D. M. and Gregory, R. T. (2012). *A survey of numerical mathematics, volume I*. Courier Dover Publications.

[Zeng, 2004] Zeng, Z. (2004). Algorithm 835: Multroot—a matlab package for computing polynomial roots and multiplicities. *ACM Transactions on Mathematical Software (TOMS)*, 30(2):218–236.

[Zeng, 2005] Zeng, Z. (2005). Computing multiple roots of inexact polynomials. *Math. Comput.*, 74(250):250.