Aproximación de raíces de polinomios usando el índice de curvas planas

Juan Luis García Zapata

Departamento de Matemáticas
Universidad de Extremadura

Raíces de polinomios

Métodos para hallar las raíces:

- Iterativos (de aproximación)
- Geométricos (de localización)

Raíces de polinomios

Métodos para hallar las raíces:

- Iterativos (de aproximación) $\Longrightarrow$ Newton
- Geométricos (de localización) $\Longrightarrow$ Contour (integración)

El método de Newton es un bucle de estimación-corrección. Para un polinomio f, a partir de una estimación inicial x_0, se construye la secuencia :

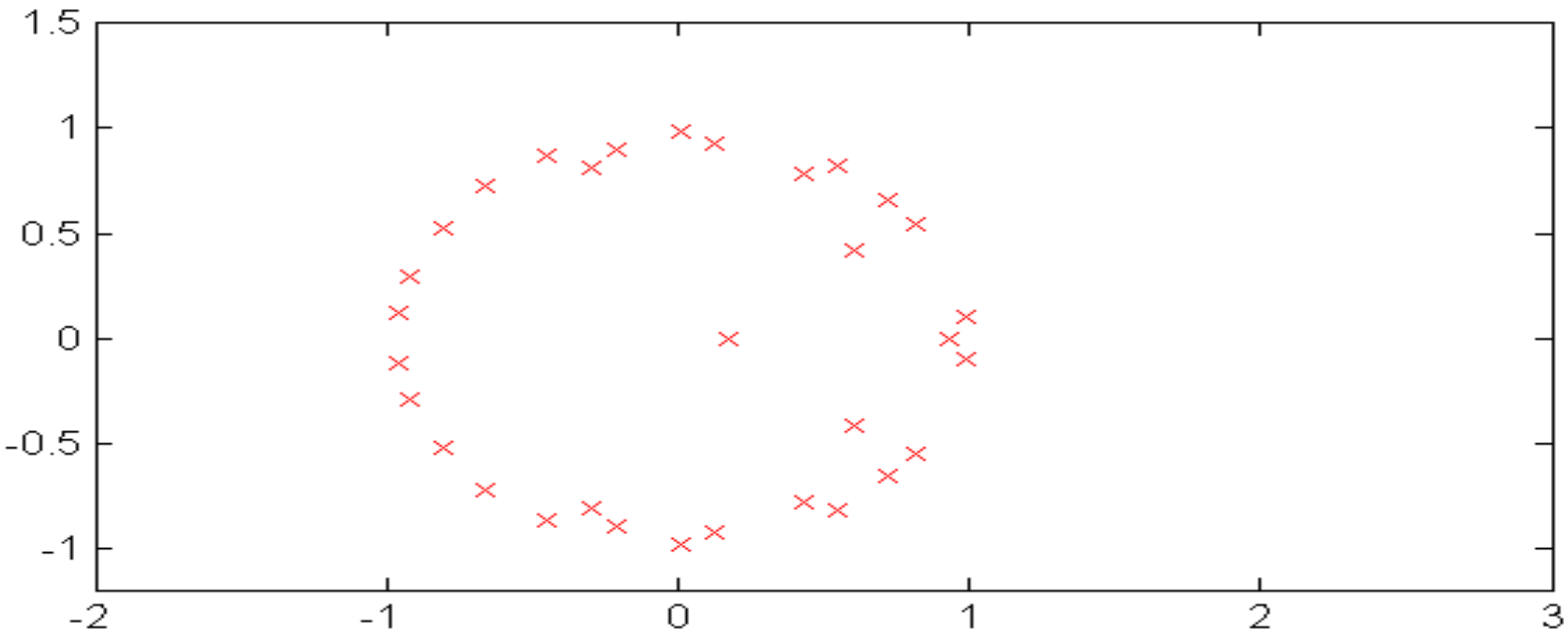
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

El método de Newton es un bucle de estimación-corrección. Para un polinomio f, a partir de una estimación inicial x_0, se construye la secuencia :

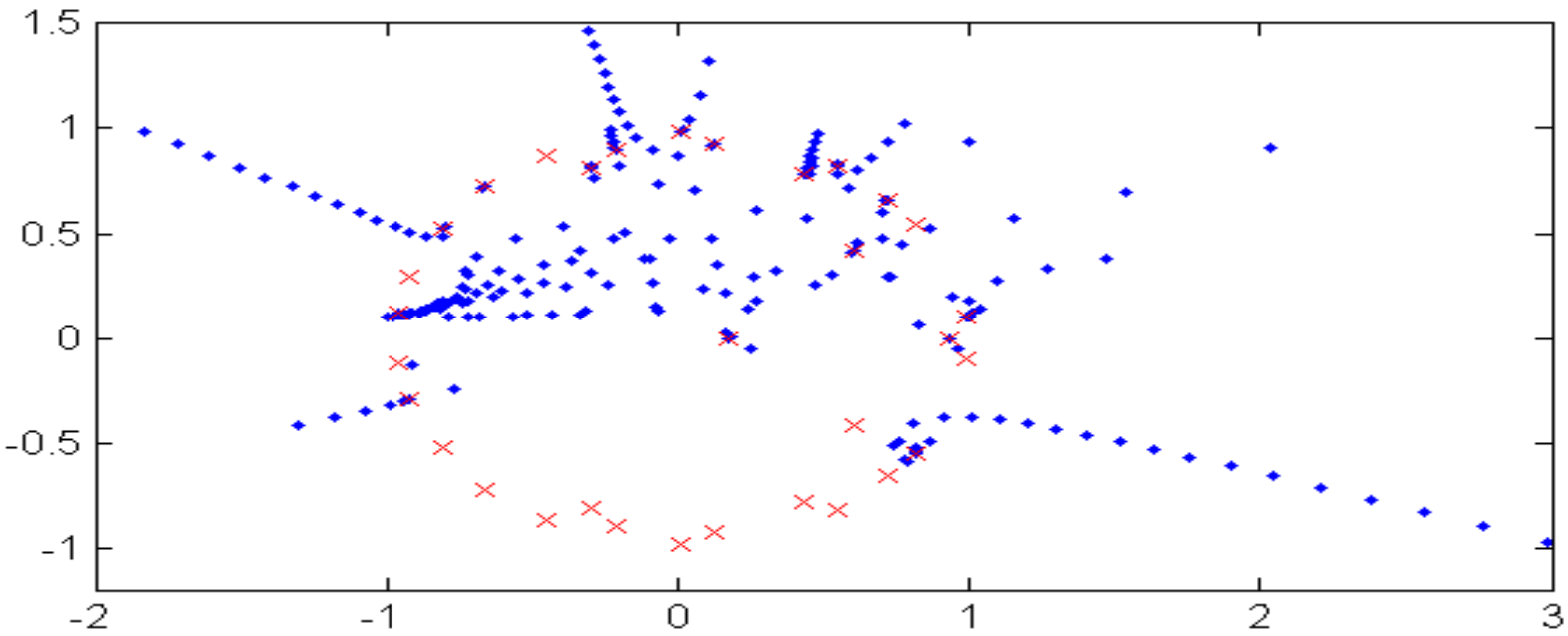
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- En ciertas condiciones, locales, se da la convergencia de x_n a una raíz.
- Dos estimaciones iniciales cercanas pueden converger a raíces muy distantes entre sí.

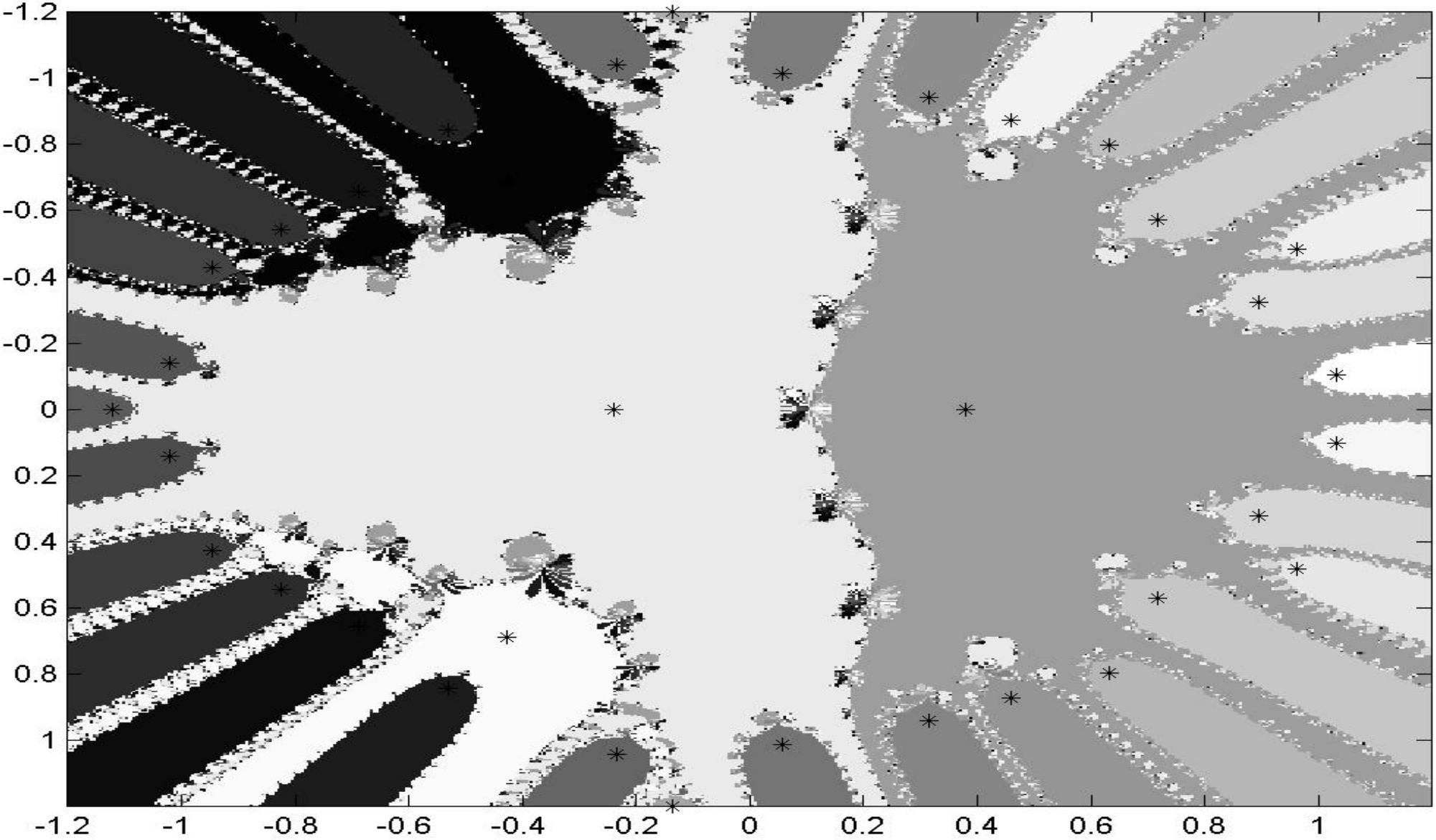
Método de Newton



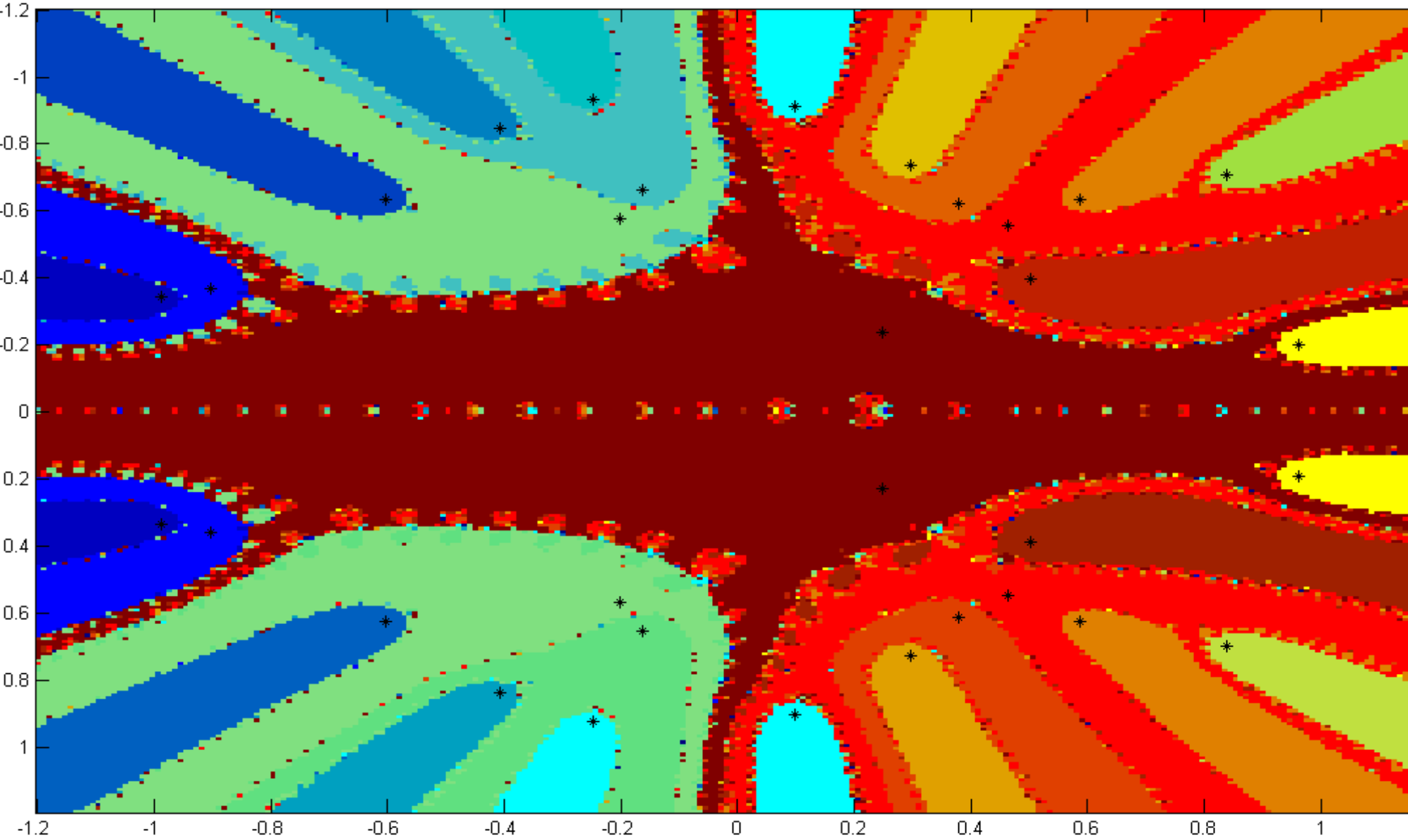
Método de Newton



Cuencas de atracción



Iterativos (Newton)



Problemas de los métodos iterativos

En general:

- Son inherentemente secuenciales (no paralelizables).
- No pueden restringir la búsqueda a una región predeterminada del plano complejo (impredicibilidad).
- El análisis de coste (y la determinación de los recursos necesarios) es difícil.

Problemas de los métodos iterativos

En general:

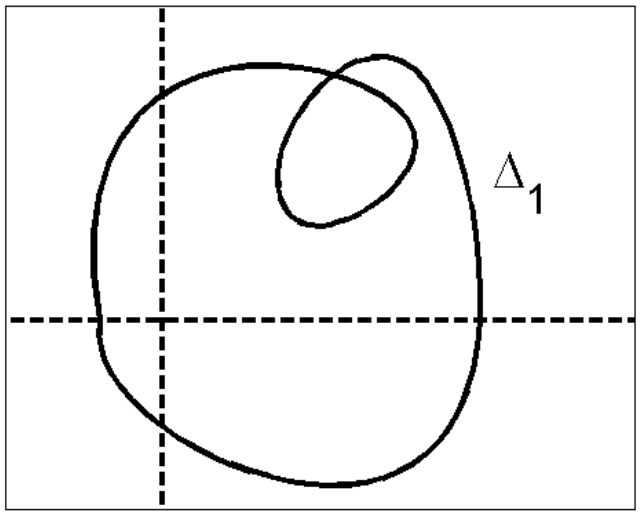
- Son inherentemente secuenciales (no paralelizables).
- No pueden restringir la búsqueda a una región predeterminada del plano complejo (impredicibilidad).
- El análisis de coste (y la determinación de los recursos necesarios) es difícil.

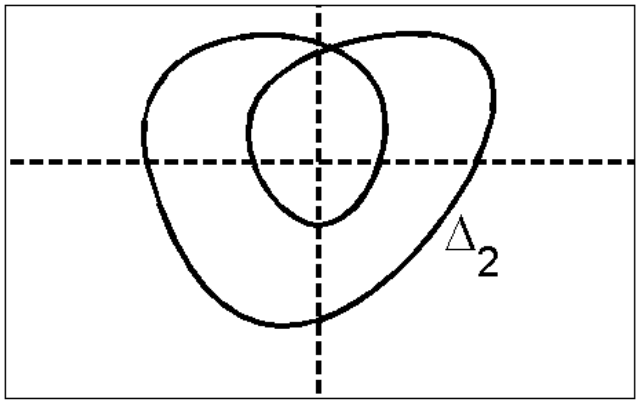
No pueden aplicarse con seguridad en polinomios de grado mayor de unas decenas

El **índice** de una curva plana cerrada $\Delta : [a, b] \to \mathbb{C}$ es el número de vueltas que da alrededor del origen.

El **índice** de una curva plana cerrada $\Delta : [a, b] \to \mathbb{C}$ es el número de vueltas que da alrededor del origen.

El índice (número de vueltas, *winding number*) de Δ_1 es 1, y el de Δ_2 es 2.





Principio del argumento (Cauchy)

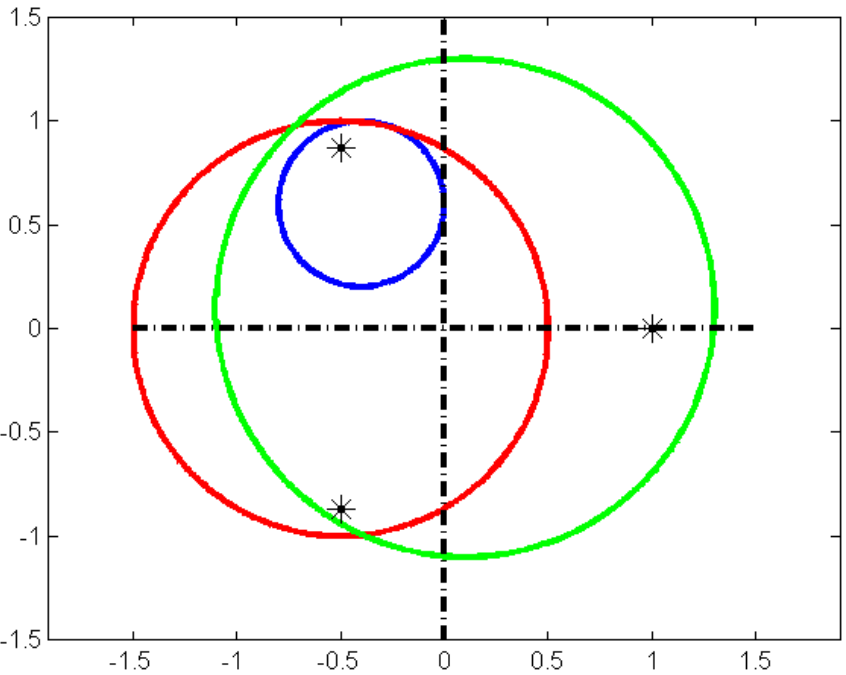
Dado un polinomio f, el número de raíces dentro de Γ es el índice de $f(\Gamma)$

Principio del argumento (Cauchy)

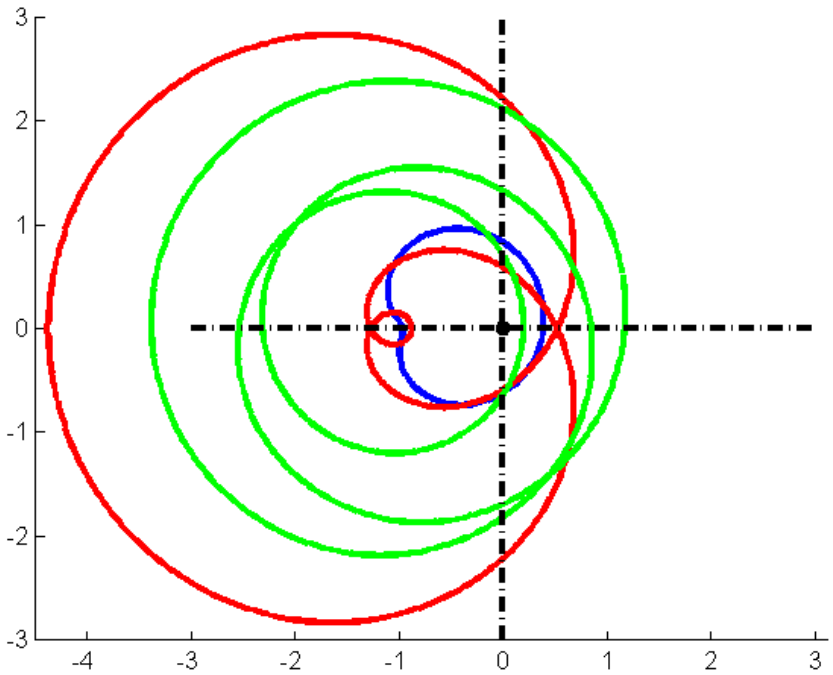
Dado un polinomio f, el número de raíces dentro de Γ es el índice de $f(\Gamma)$

Por ejemplo: $f(z) = z^3 - 1$

Contornos Γ



Curvas $\Delta = f(\Gamma)$

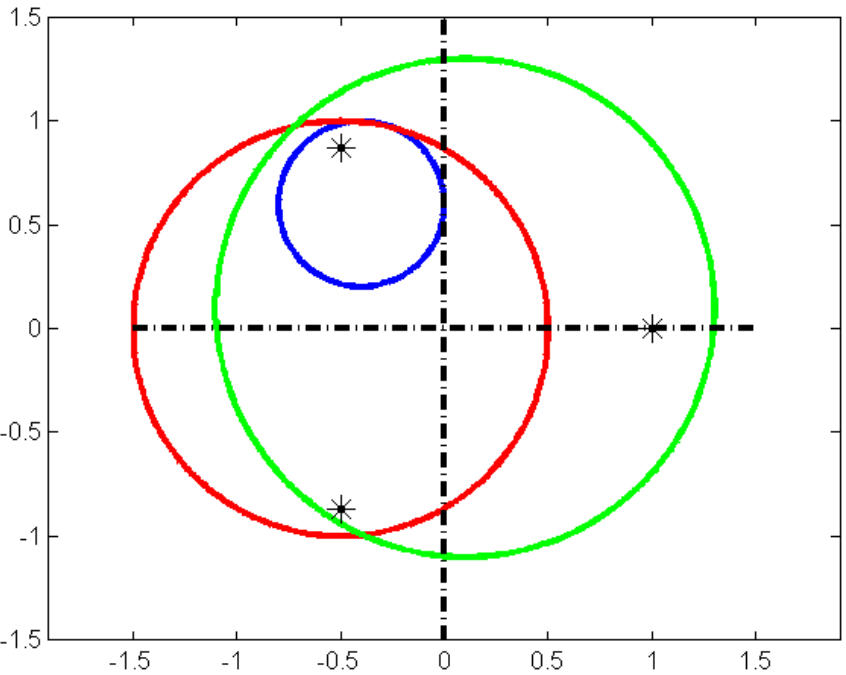


Principio del argumento (Cauchy)

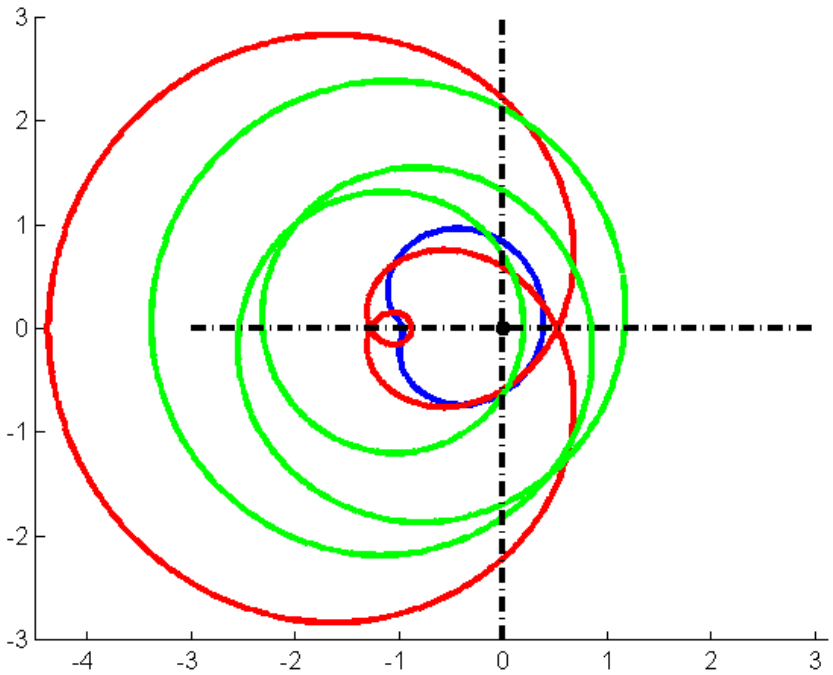
Dado un polinomio f, el número de raíces dentro de Γ es el índice de $f(\Gamma)$

Por ejemplo: $f(z) = z^3 - 1$

Contornos Γ

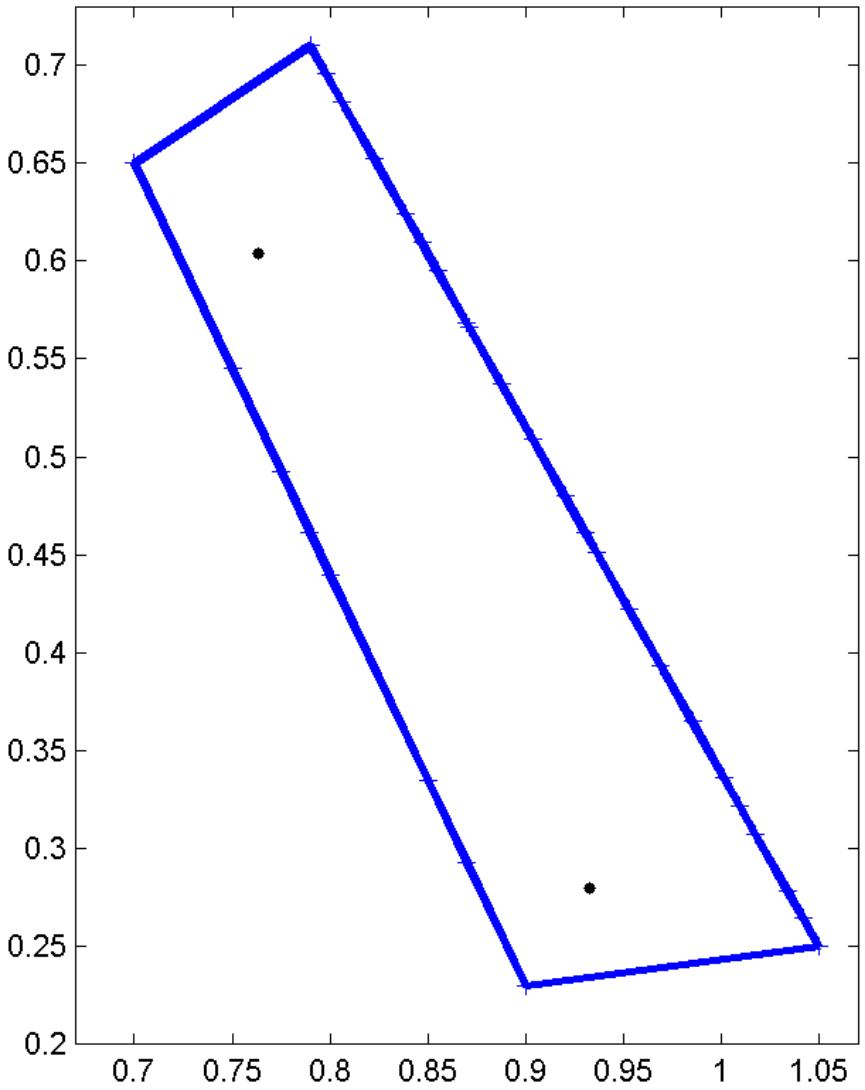


Curvas $\Delta = f(\Gamma)$

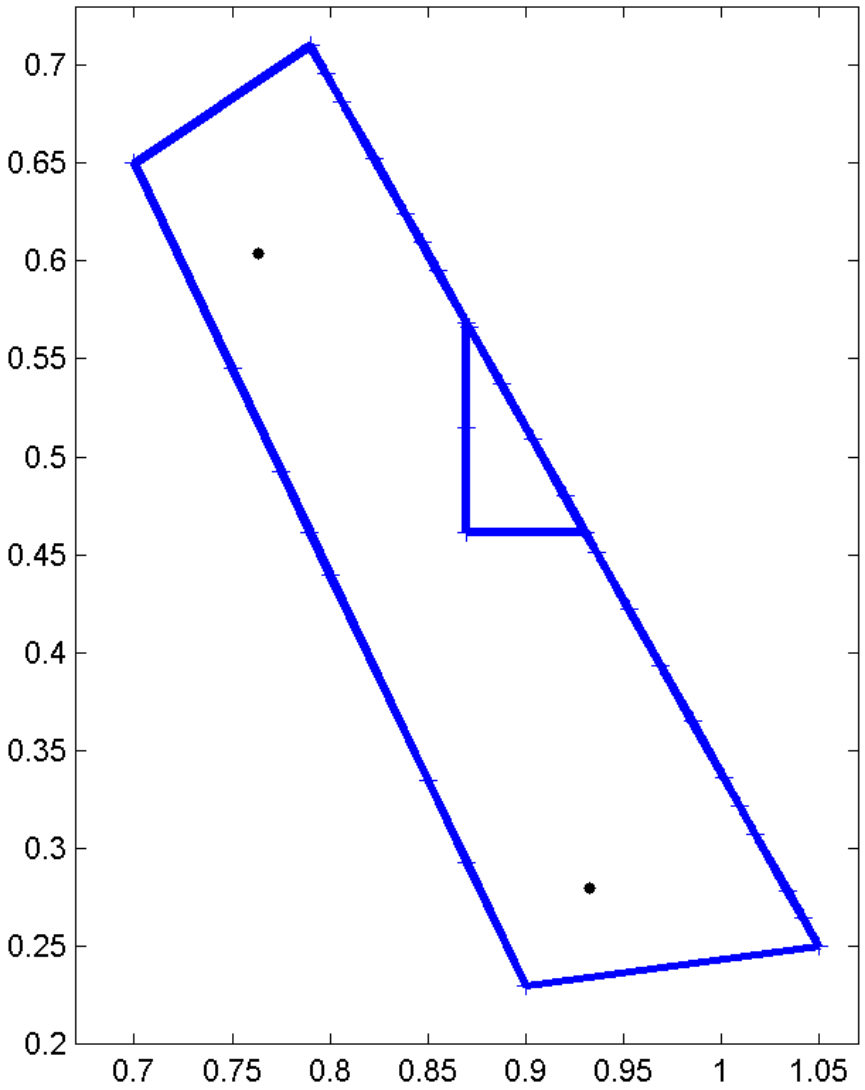


Así tenemos el número de raíces. Pero ¿dónde están?

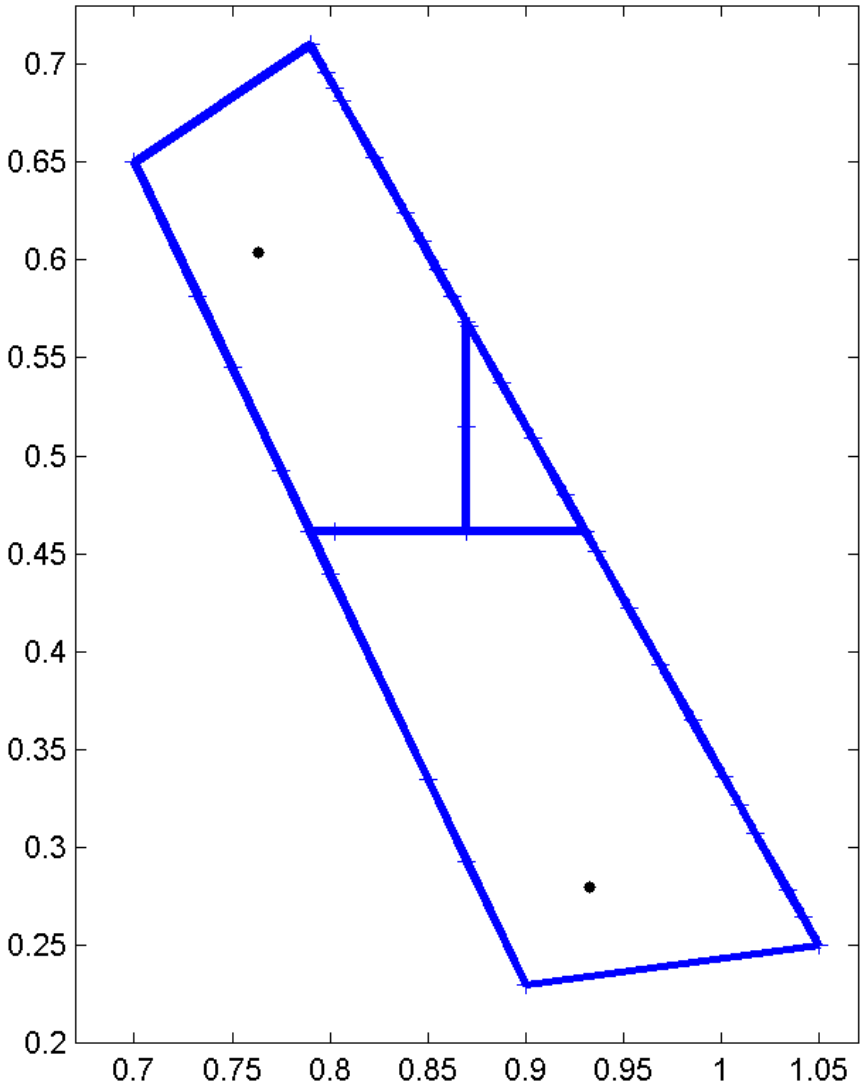
Ejemplo de localización, con descomposición recursiva:



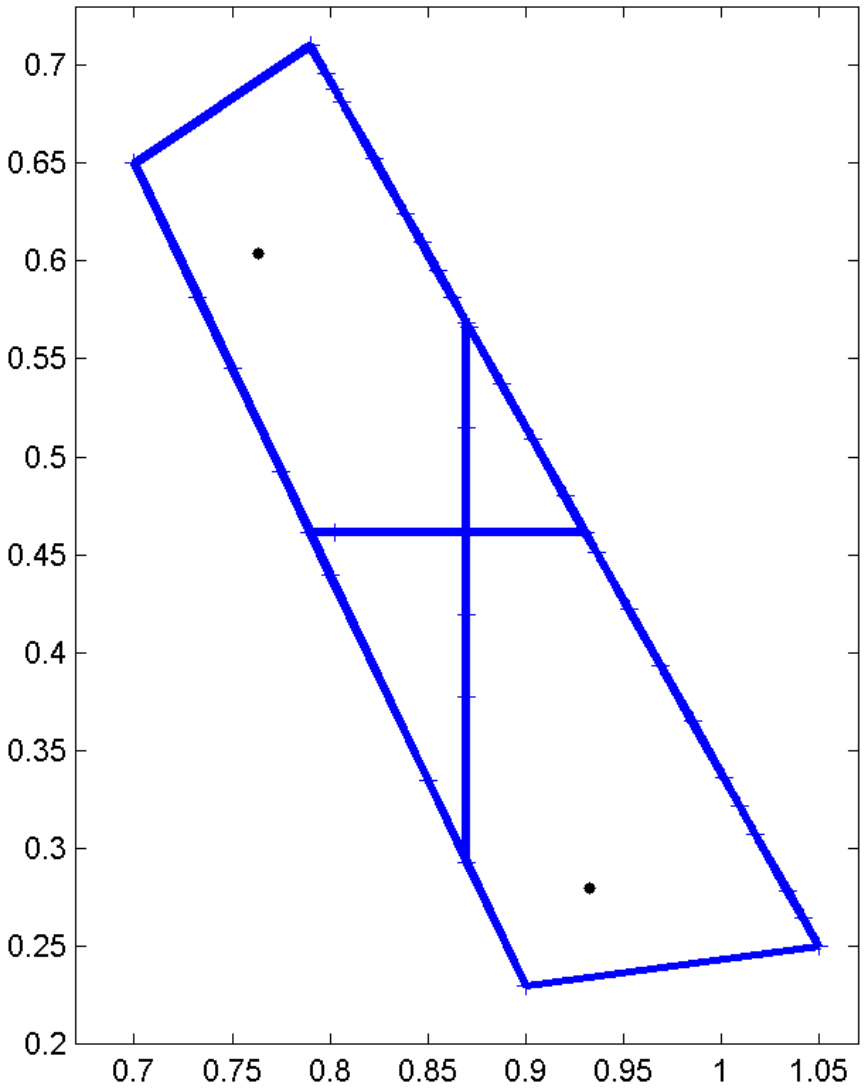
Ejemplo de localización, con descomposición recursiva:



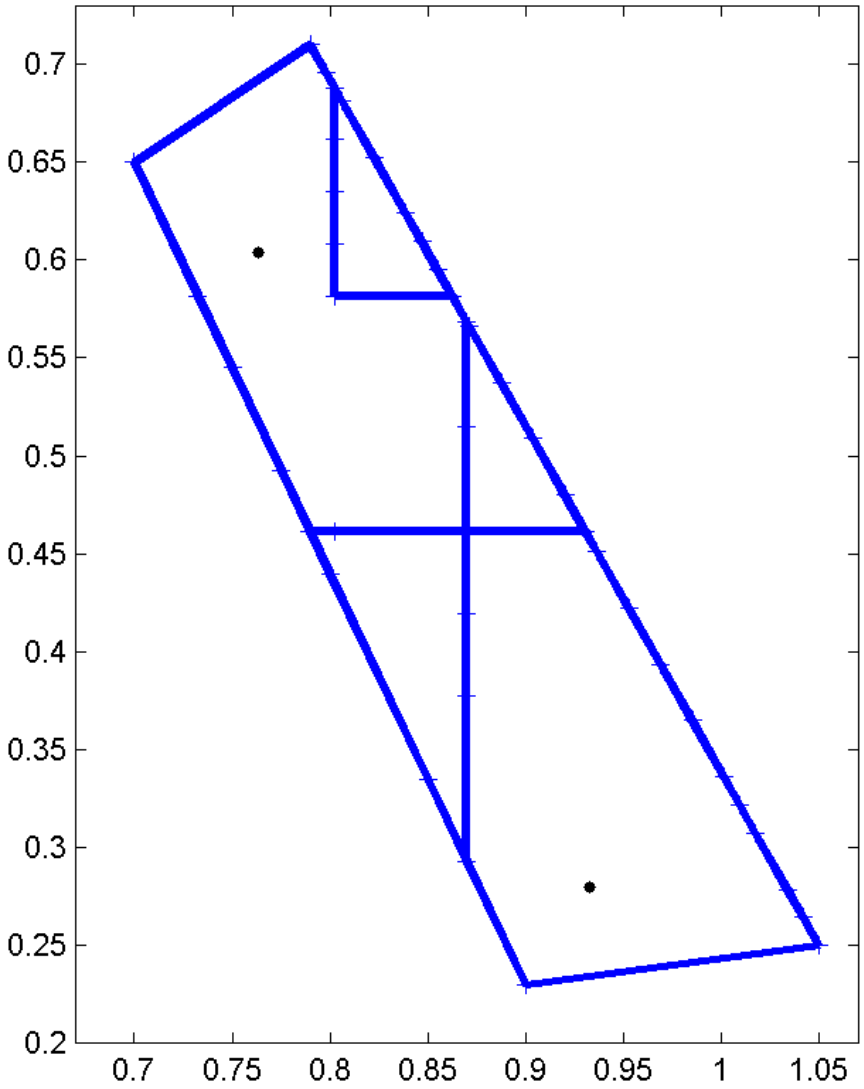
Ejemplo de localización, con descomposición recursiva:



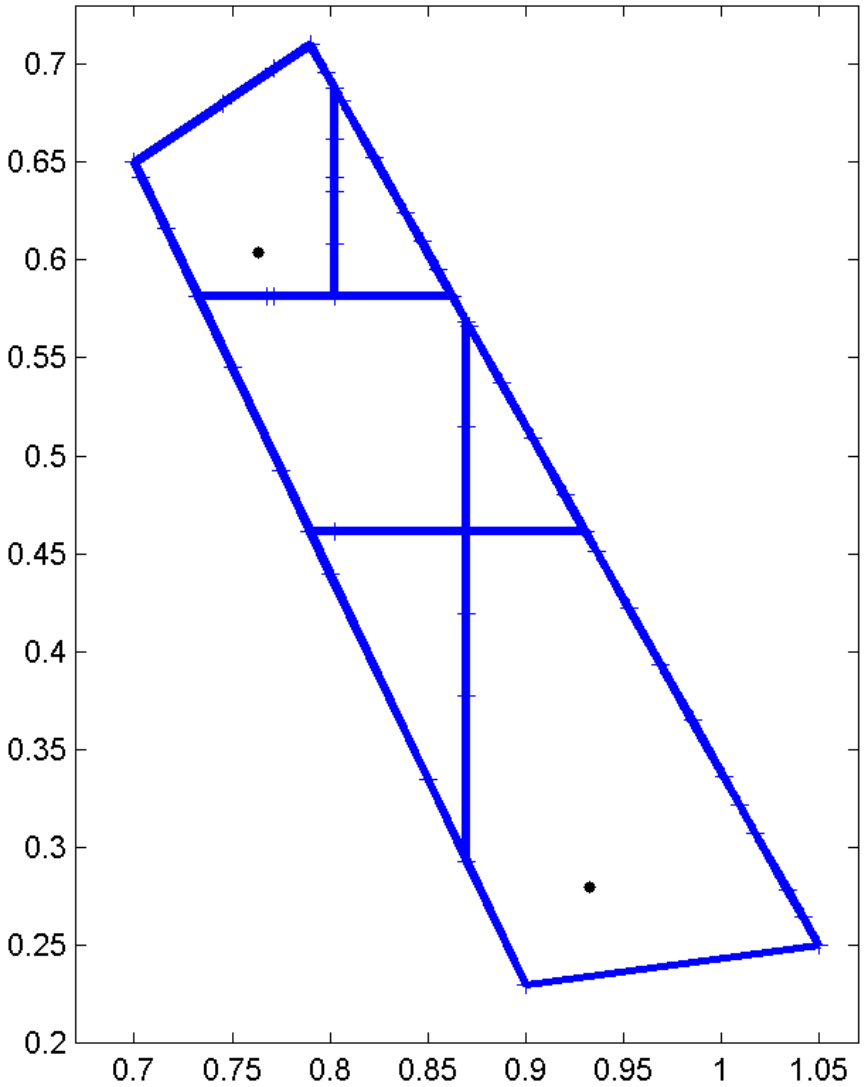
Ejemplo de localización, con descomposición recursiva:



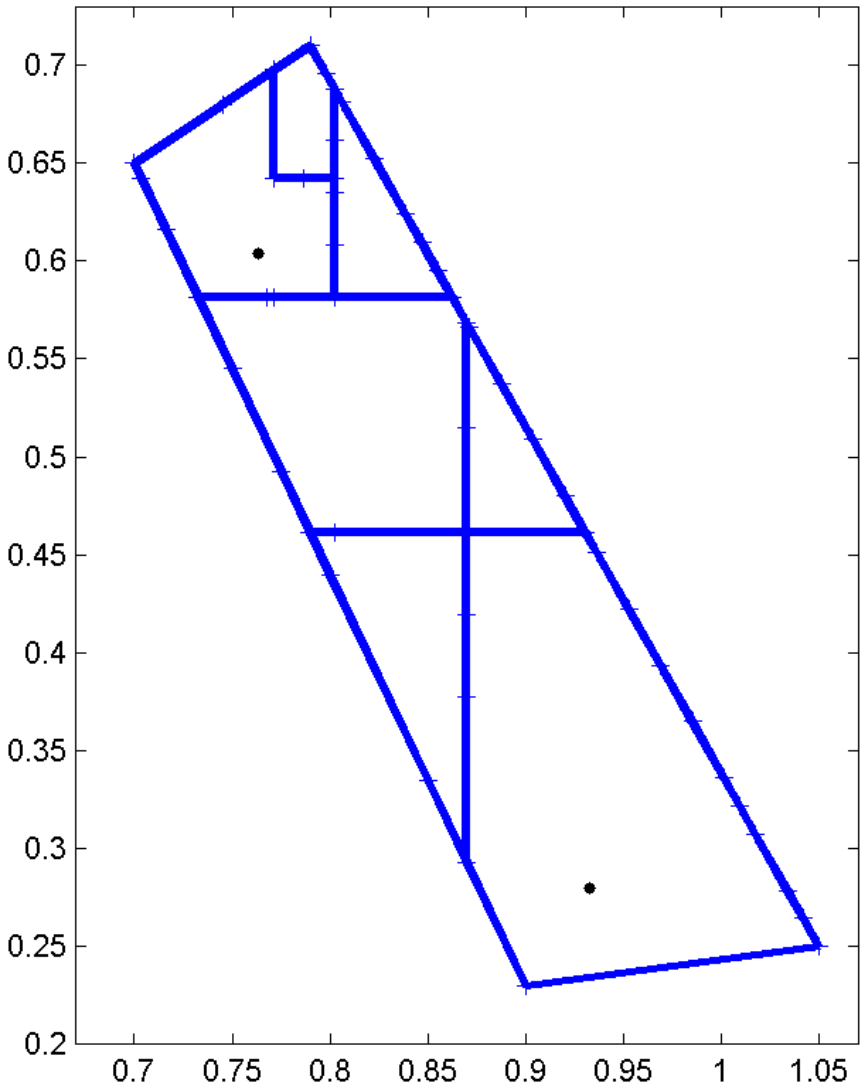
Ejemplo de localización, con descomposición recursiva:



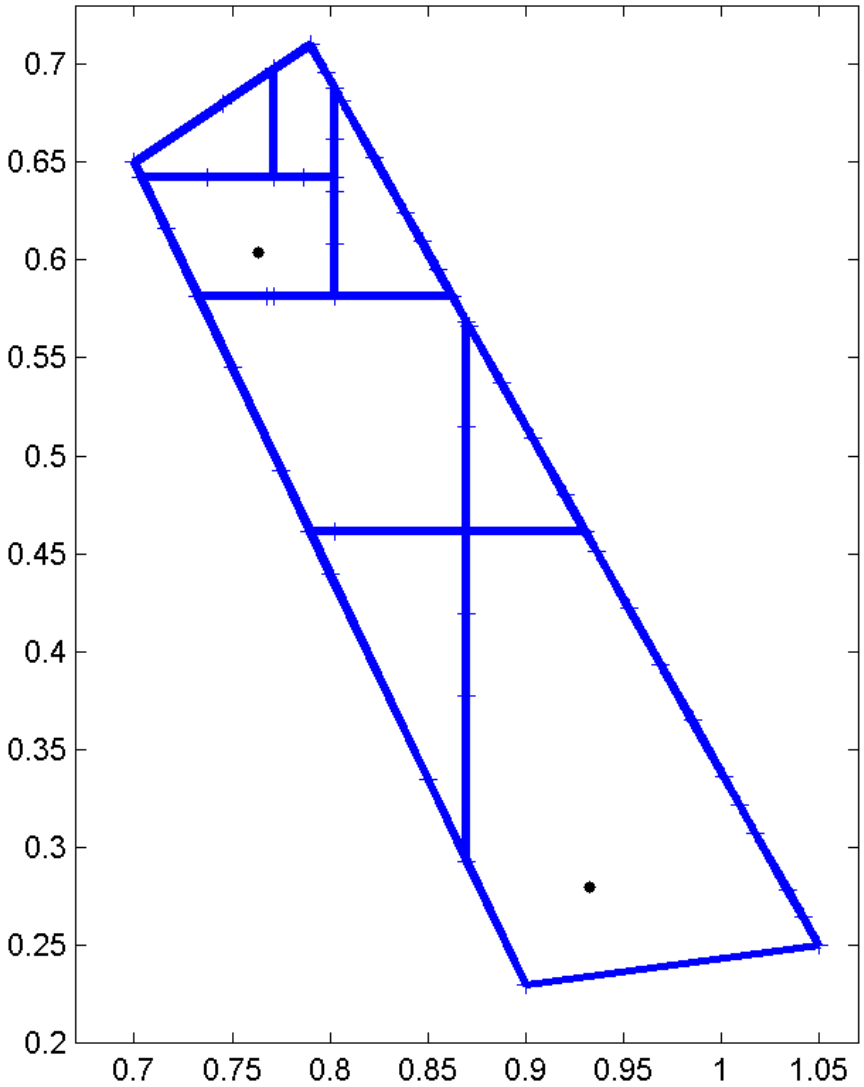
Ejemplo de localización, con descomposición recursiva:



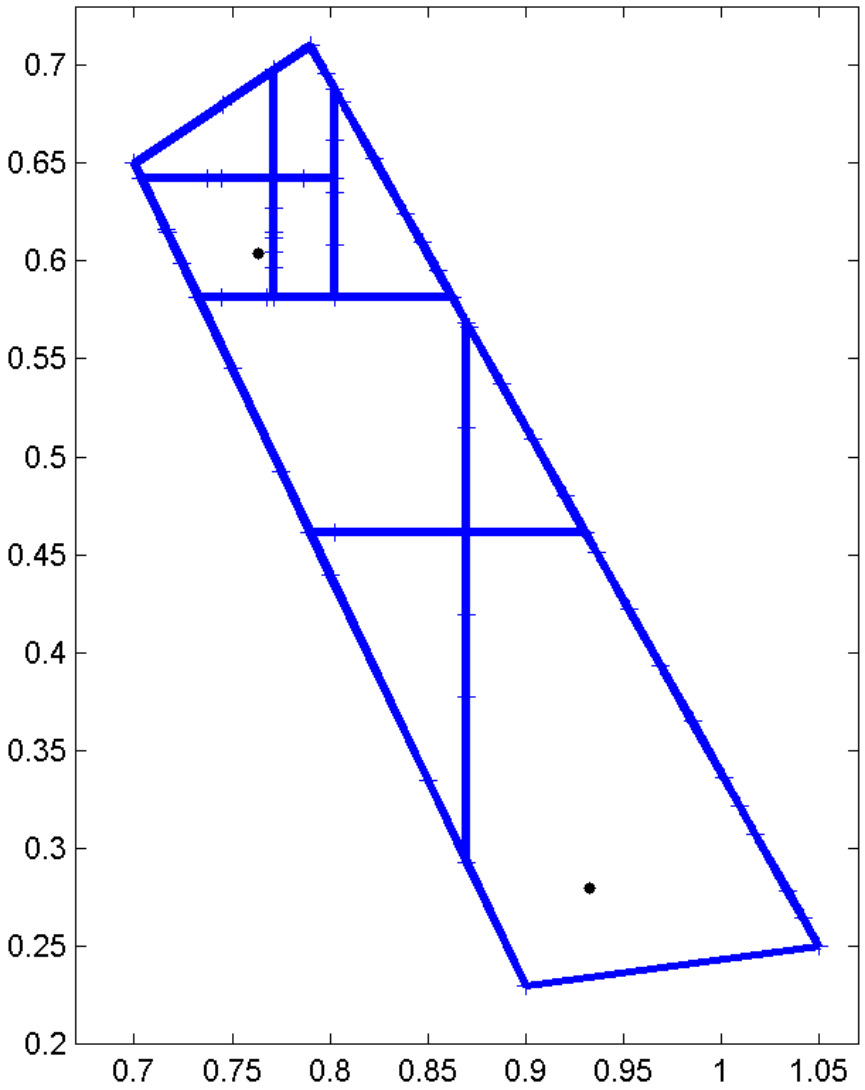
Ejemplo de localización, con descomposición recursiva:



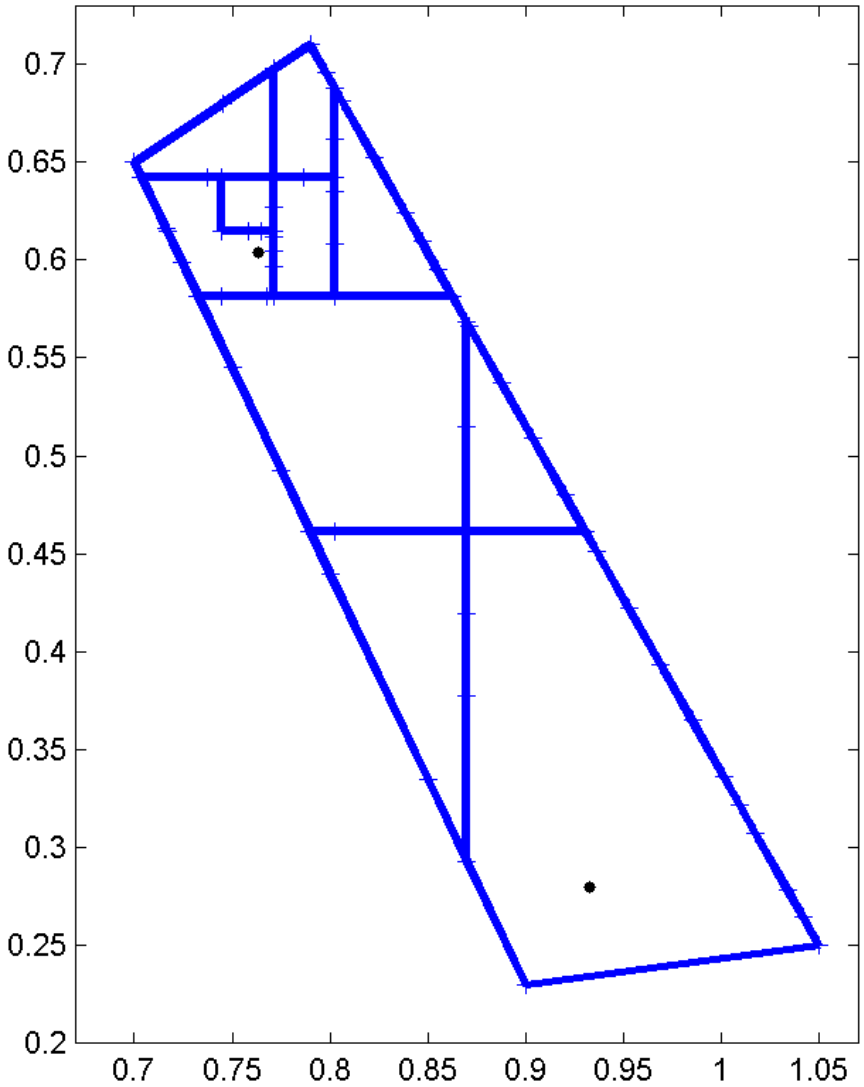
Ejemplo de localización, con descomposición recursiva:



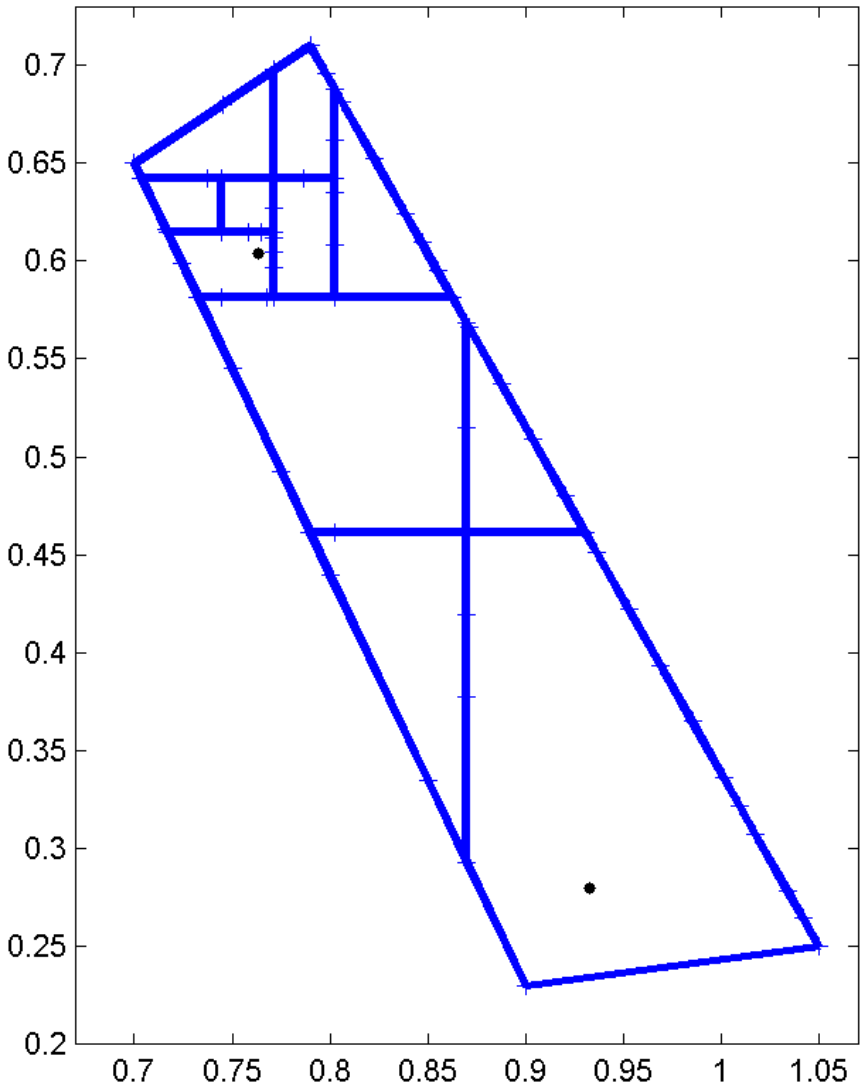
Ejemplo de localización, con descomposición recursiva:



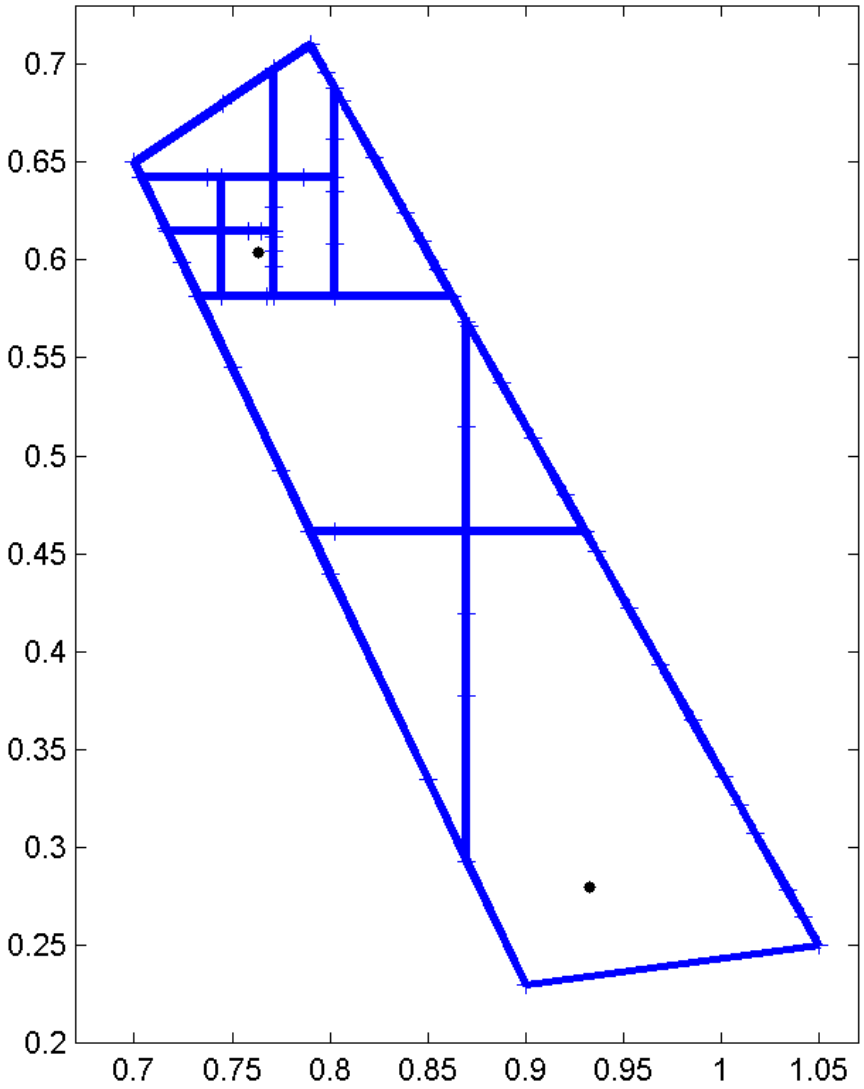
Ejemplo de localización, con descomposición recursiva:



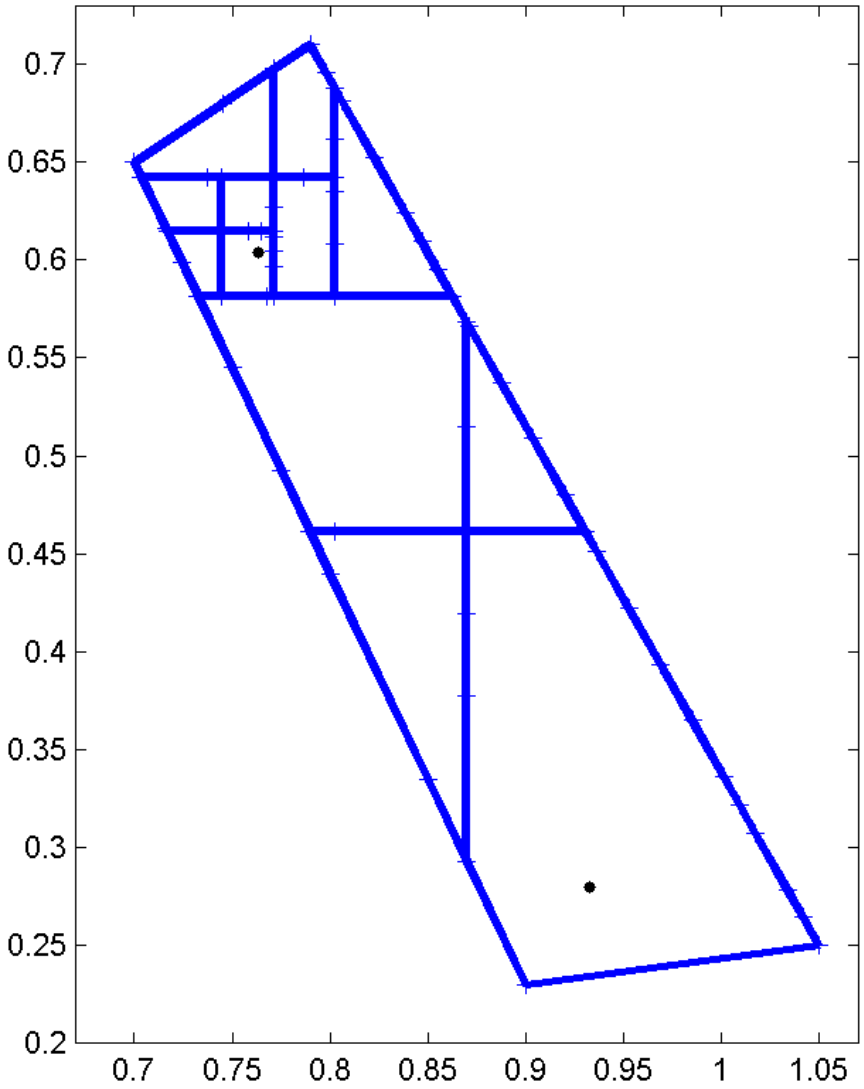
Ejemplo de localización, con descomposición recursiva:



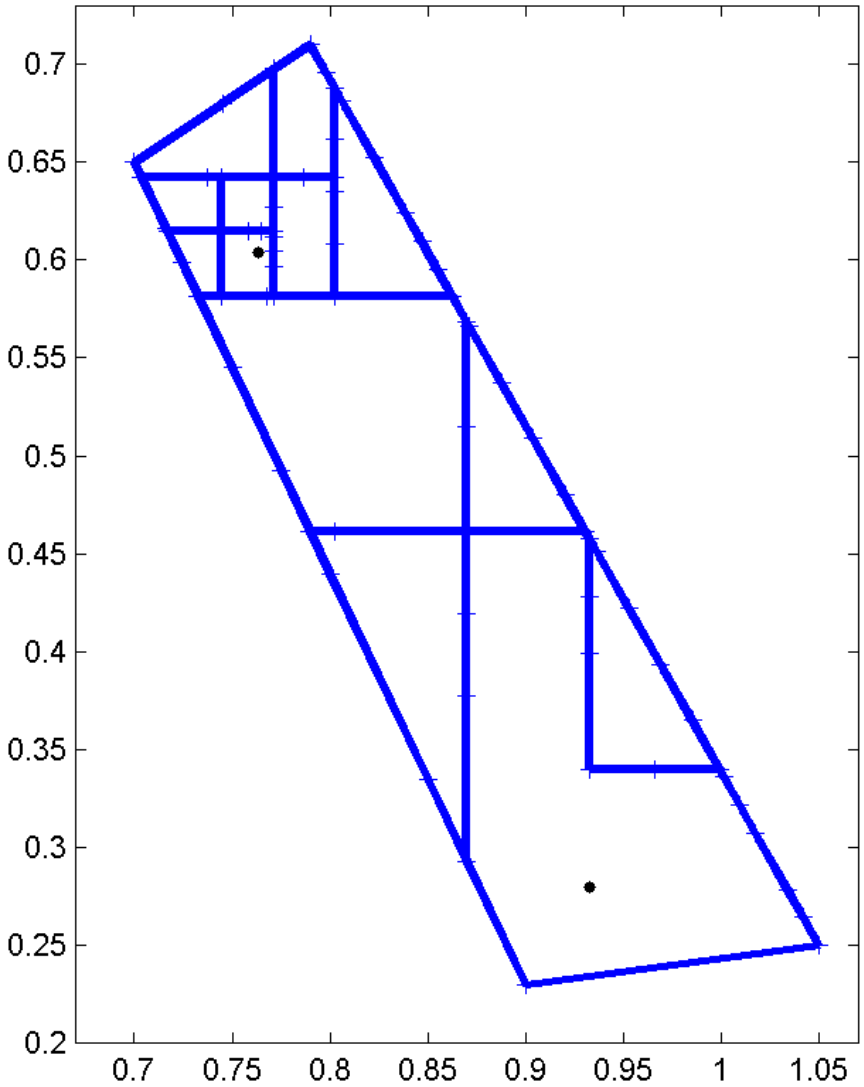
Ejemplo de localización, con descomposición recursiva:



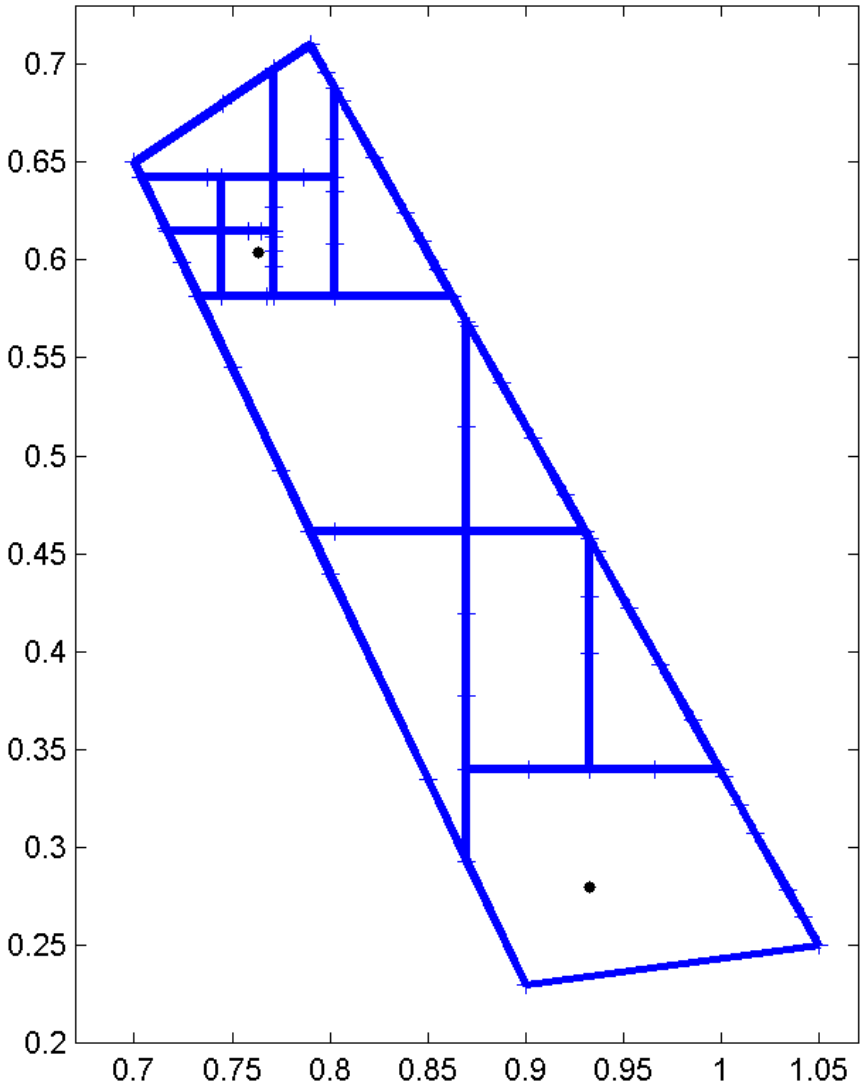
Ejemplo de localización, con descomposición recursiva:



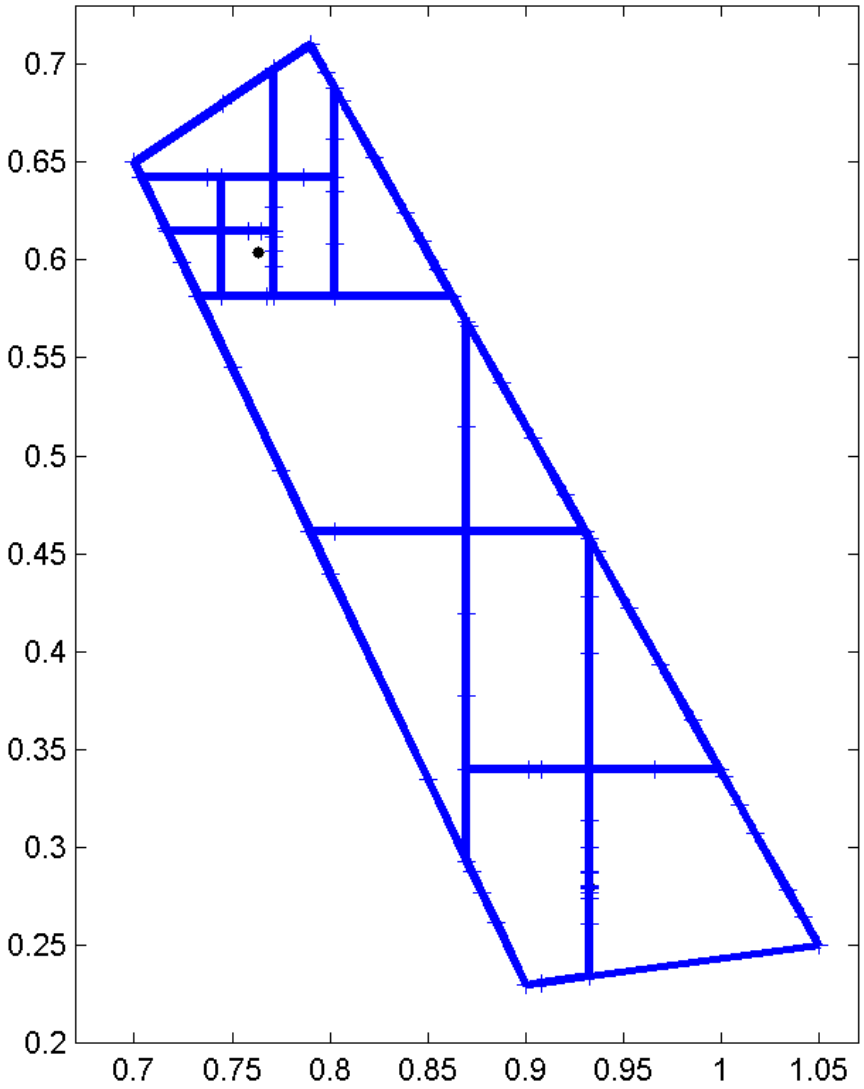
Ejemplo de localización, con descomposición recursiva:



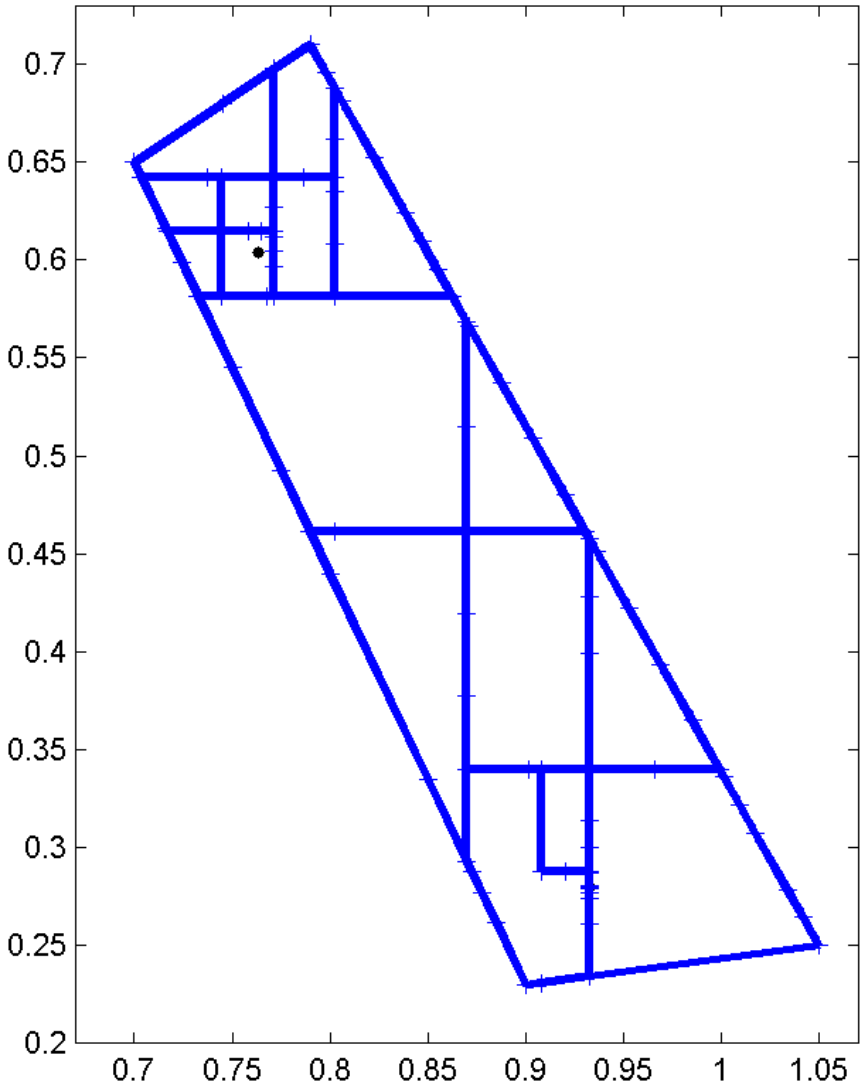
Ejemplo de localización, con descomposición recursiva:



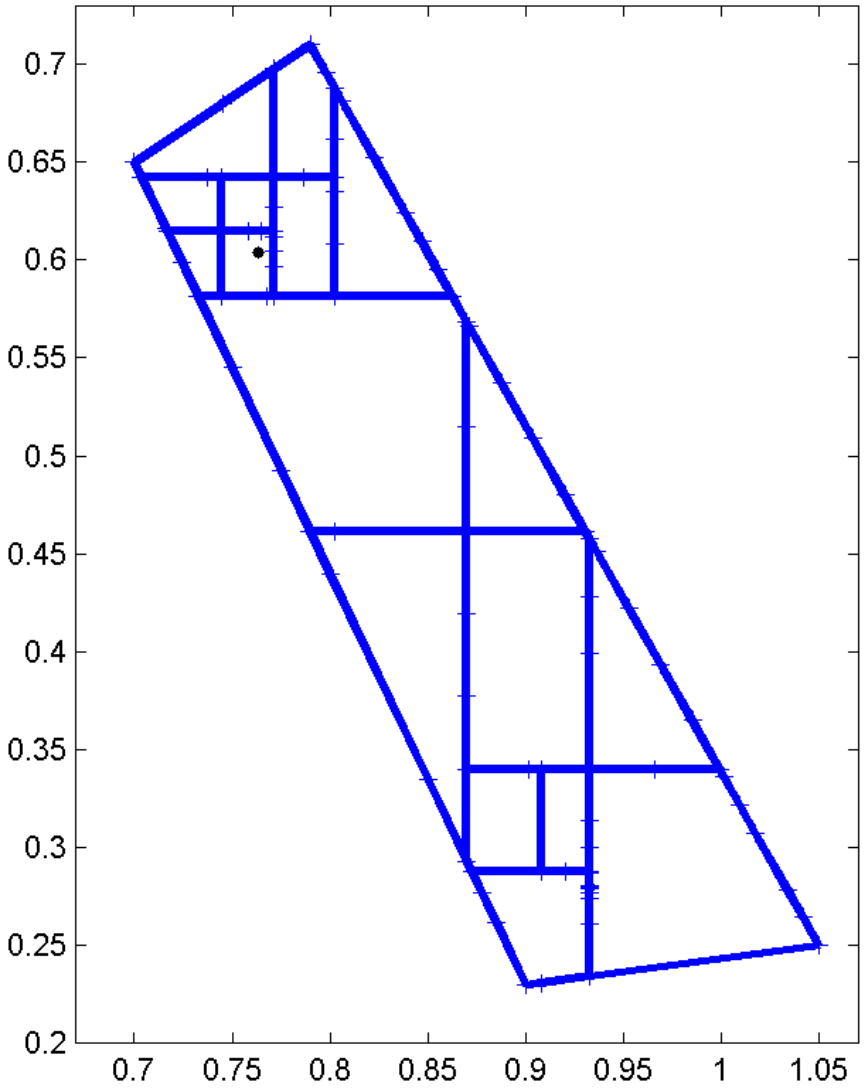
Ejemplo de localización, con descomposición recursiva:



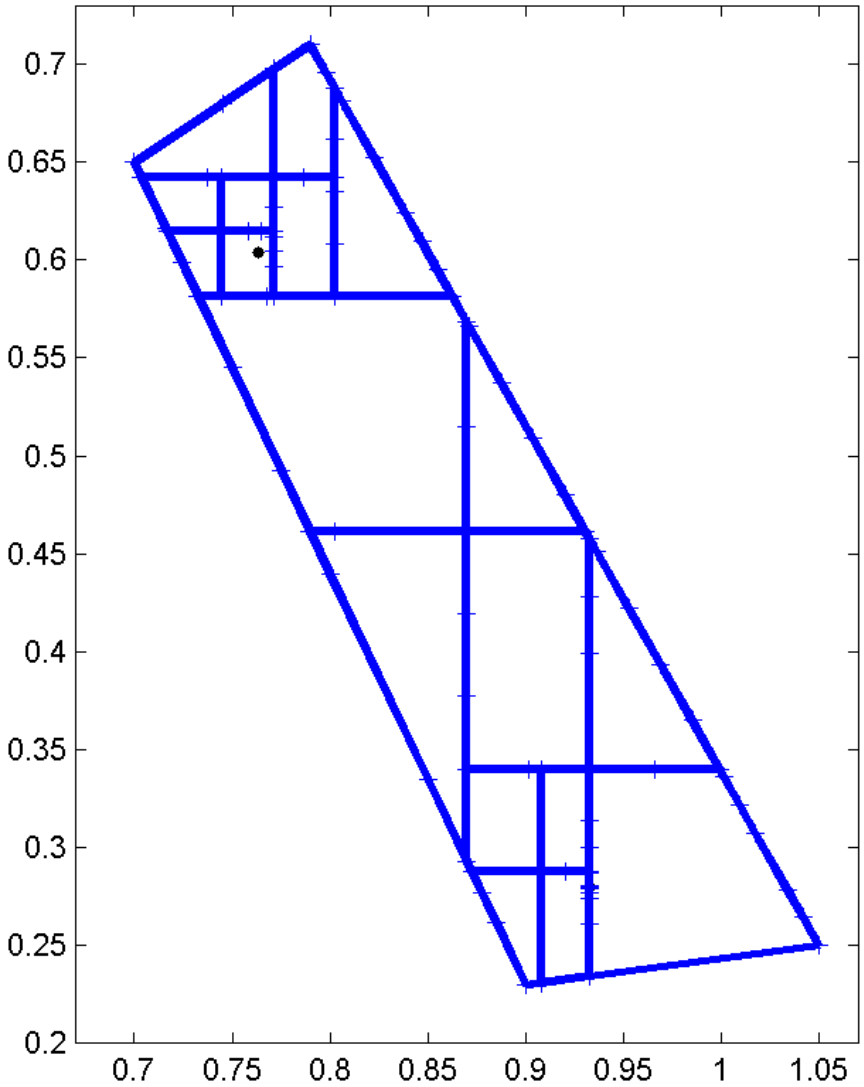
Ejemplo de localización, con descomposición recursiva:



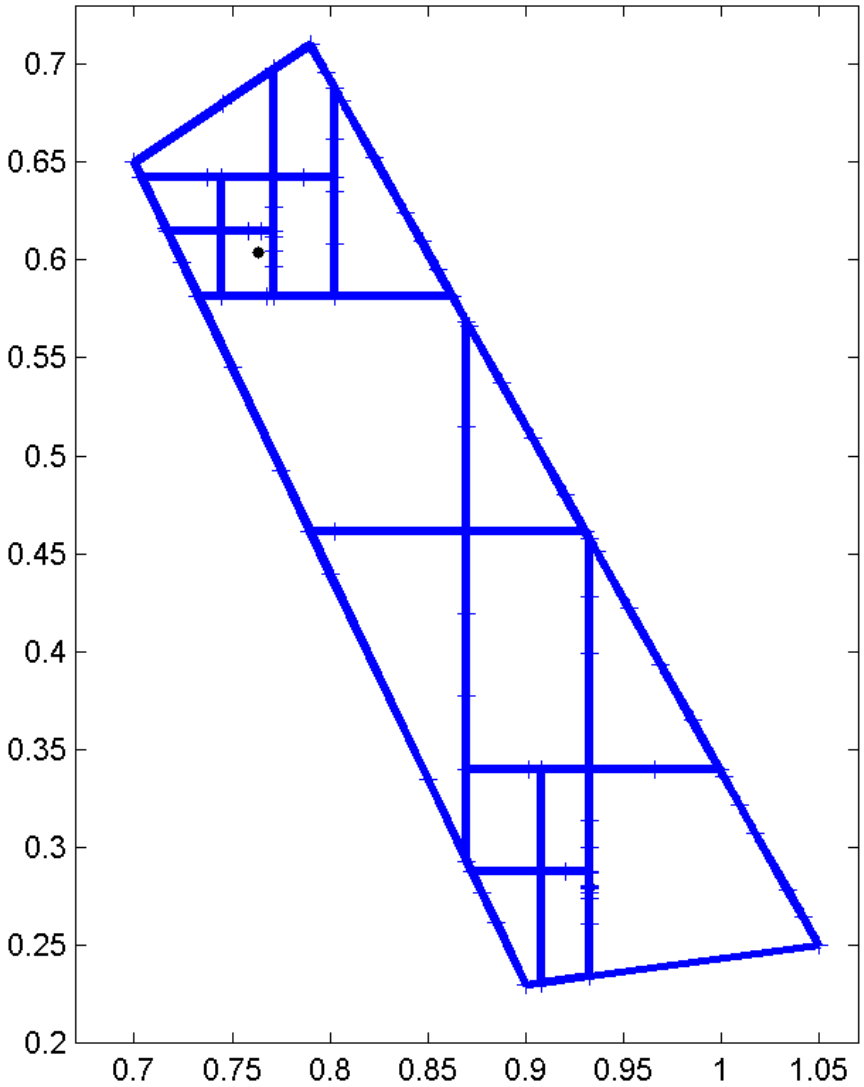
Ejemplo de localización, con descomposición recursiva:



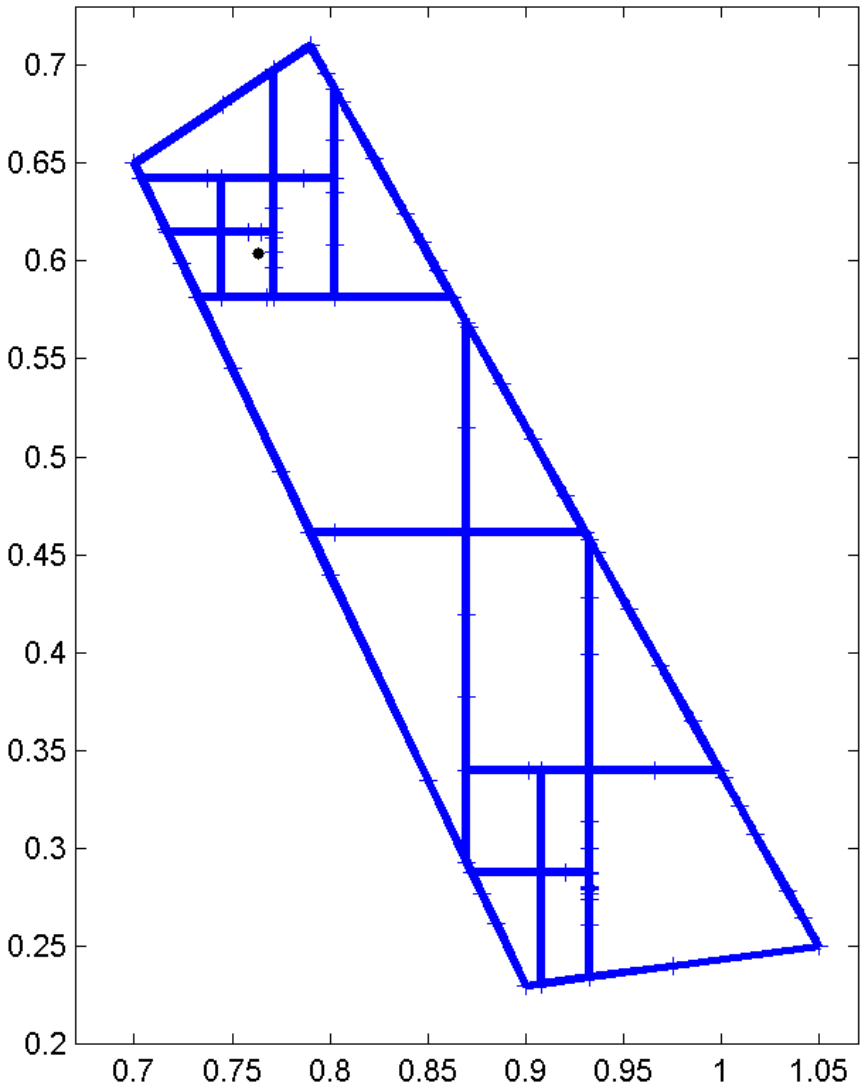
Ejemplo de localización, con descomposición recursiva:

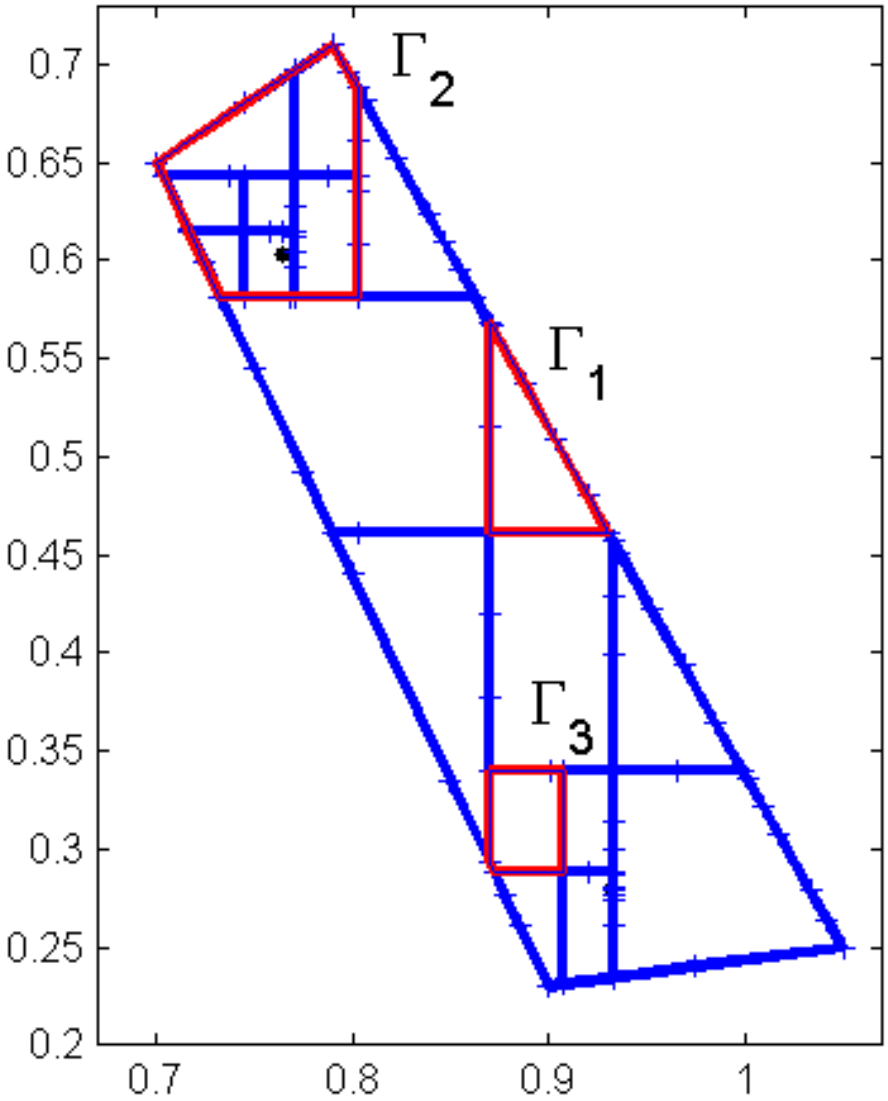


Ejemplo de localización, con descomposición recursiva:

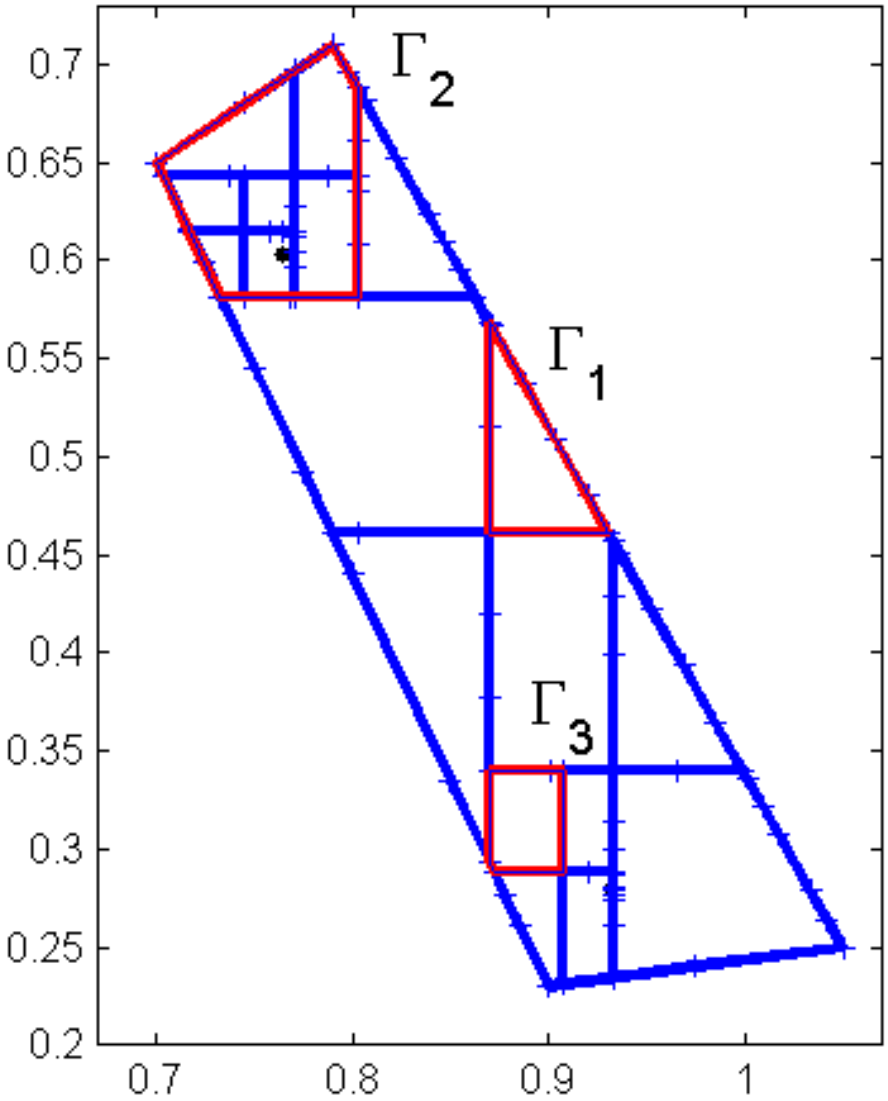


Ejemplo de localización, con descomposición recursiva:





Cada región está definida por su borde Γ.

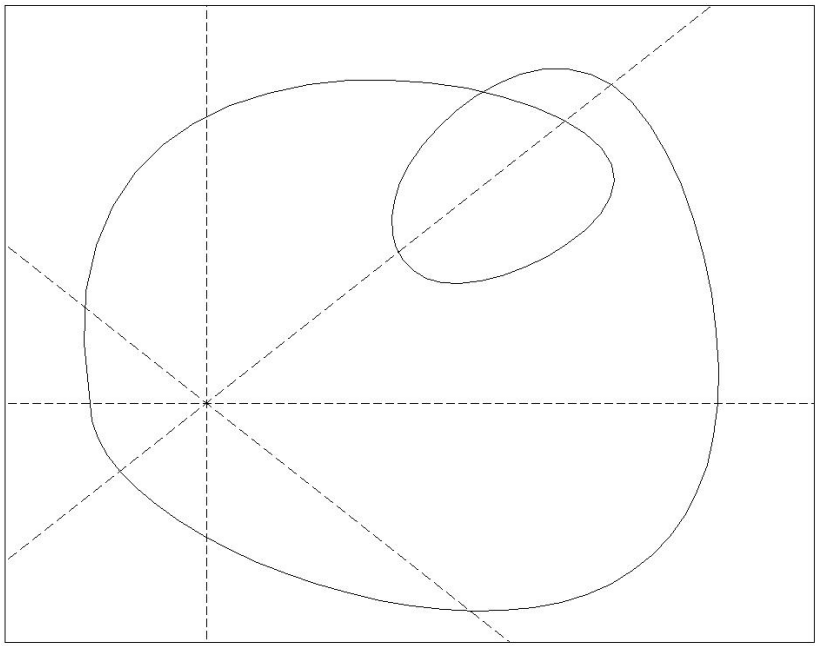


Cada región está definida por su borde Γ.

1. ¿Cómo calcular numéricamente el índice de cada $f(\Gamma)$?
2. ¿Cómo descomponer las regiones con índice no nulo?

Cálculo del índice

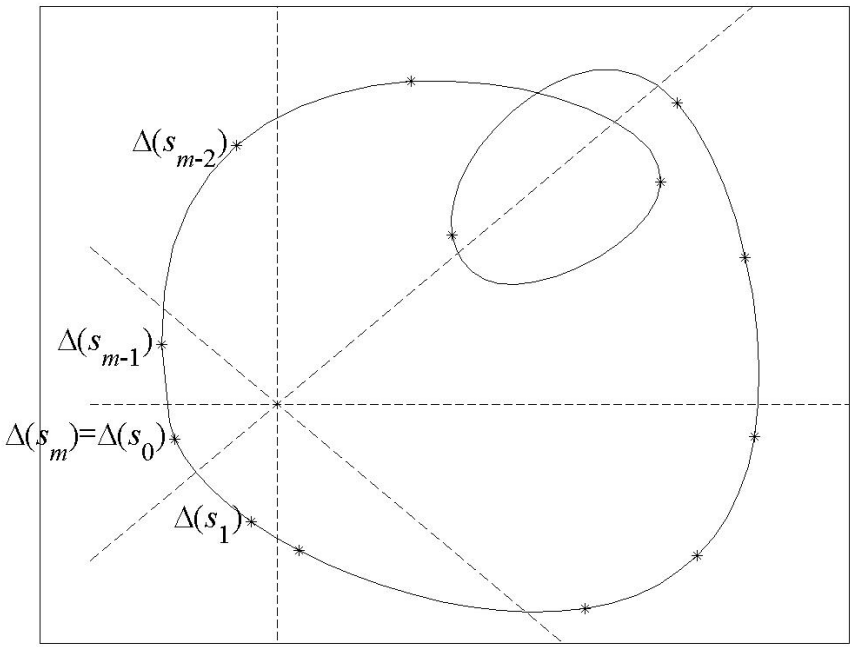
La curva Δ viene dada analíticamente como una función $\Delta : [a, b] \to \mathbb{C}$ de parámetro real.



Para calcular el índice de Δ, se divide el plano en ocho sectores, cada uno la mitad de un cuadrante.

Cálculo del índice

La curva Δ viene dada analíticamente como una función $\Delta : [a, b] \to \mathbb{C}$ de parámetro real.

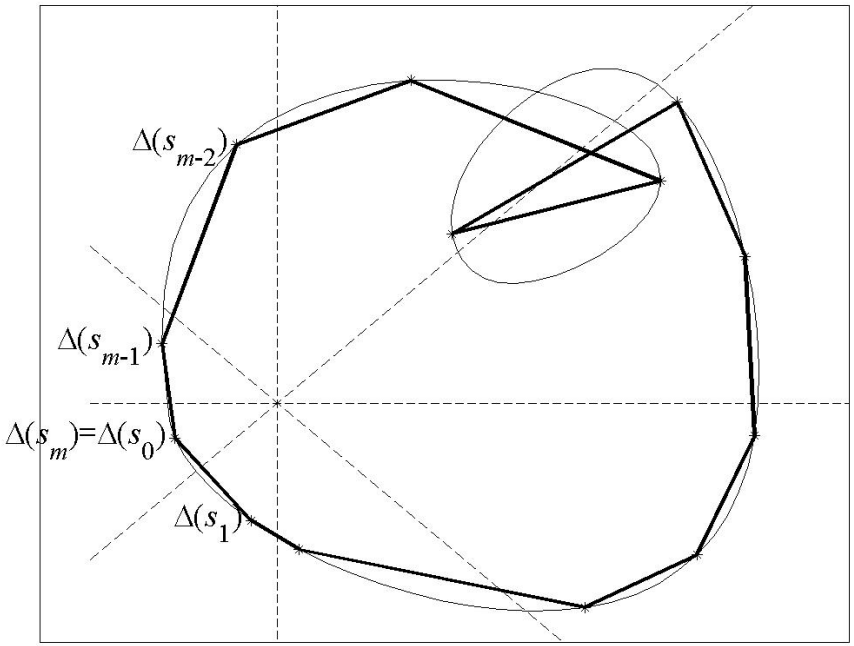


Para calcular el índice de Δ, se divide el plano en ocho sectores, cada uno la mitad de un cuadrante.

La curva se muestrea en valores del parámetro tales que los puntos correspondientes están conectados (es decir, en el mismo sector o adyacente).

Cálculo del índice

La curva Δ viene dada analíticamente como una función $\Delta : [a, b] \to \mathbb{C}$ de parámetro real.



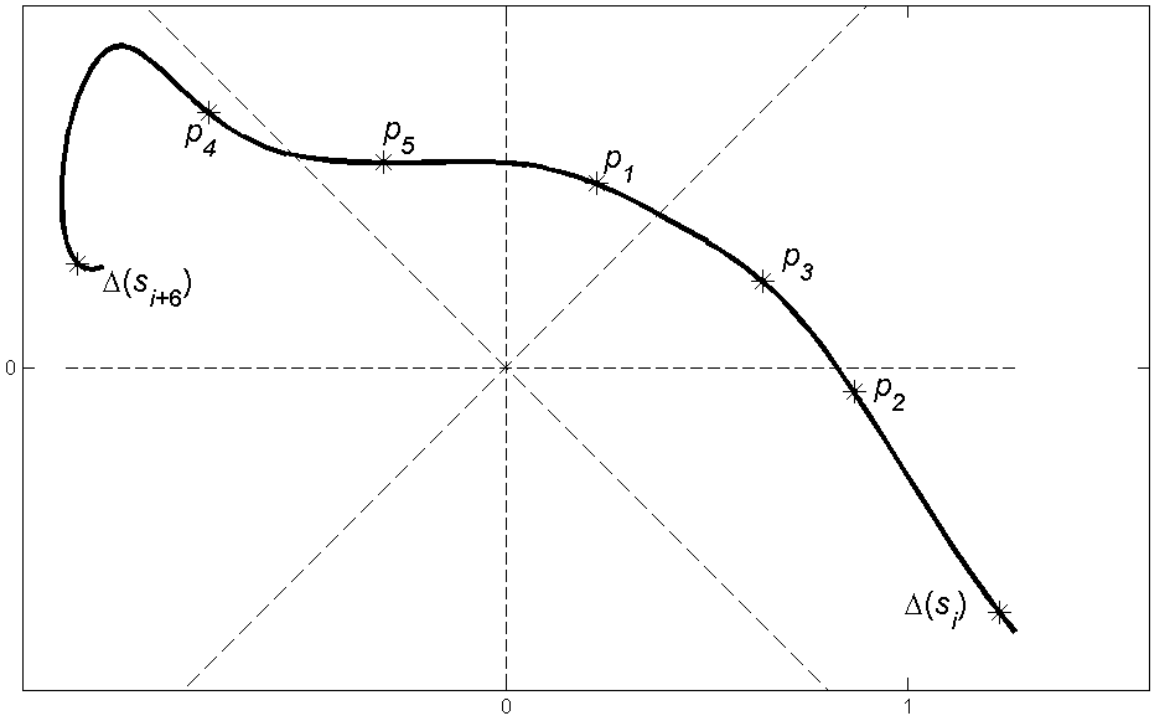
Para calcular el índice de Δ, se divide el plano en ocho sectores, cada uno la mitad de un cuadrante.

La curva se muestrea en valores del parámetro tales que los puntos correspondientes están conectados (es decir, en el mismo sector o adyacente).

De la poligonal $\widetilde{\Delta}$, el índice es el número de cruces por el eje OX positivo [Henrici, 74].

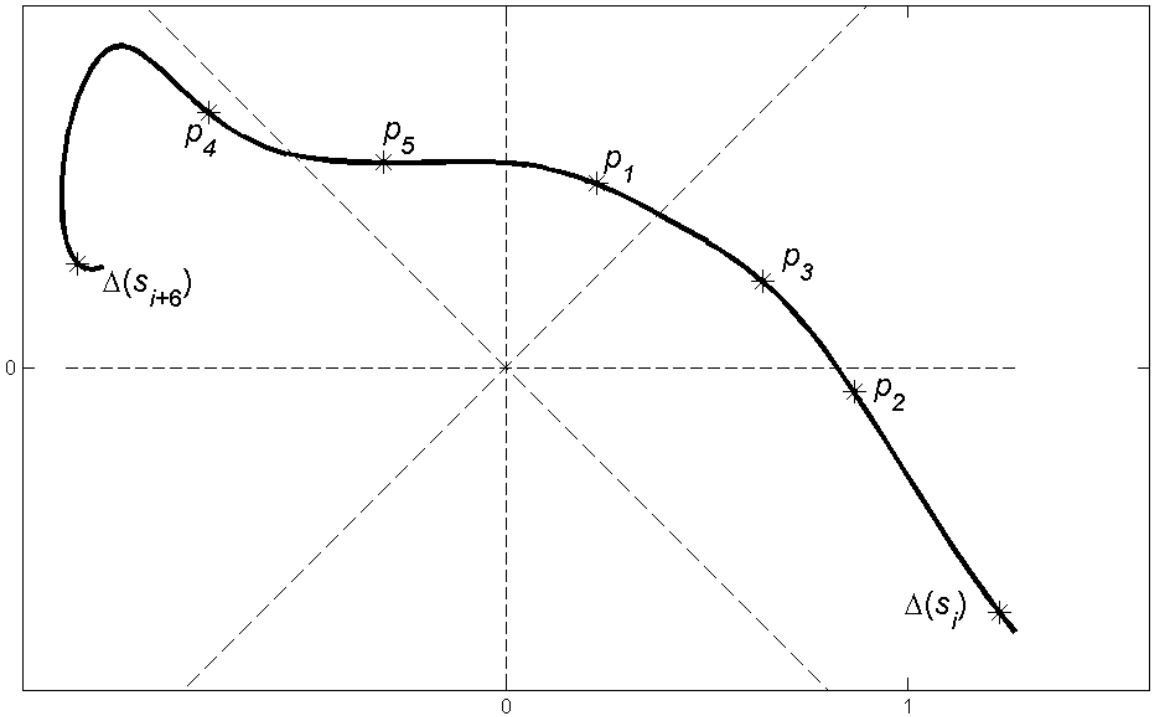
¿Cómo hallar un muestreo conectado $S = (a = s_0, s_1, \ldots, s_n = b)$?

Si un muestreo no está conectado, insertamos valores intermedios progresivamente hasta que lo esté. Procedimiento de inserción [Ying and Katz, 88].



¿Cómo hallar un muestreo conectado $S = (a = s_0, s_1, \ldots, s_n = b)$?

Si un muestreo no está conectado, insertamos valores intermedios progresivamente hasta que lo esté. Procedimiento de inserción [Ying and Katz, 88].



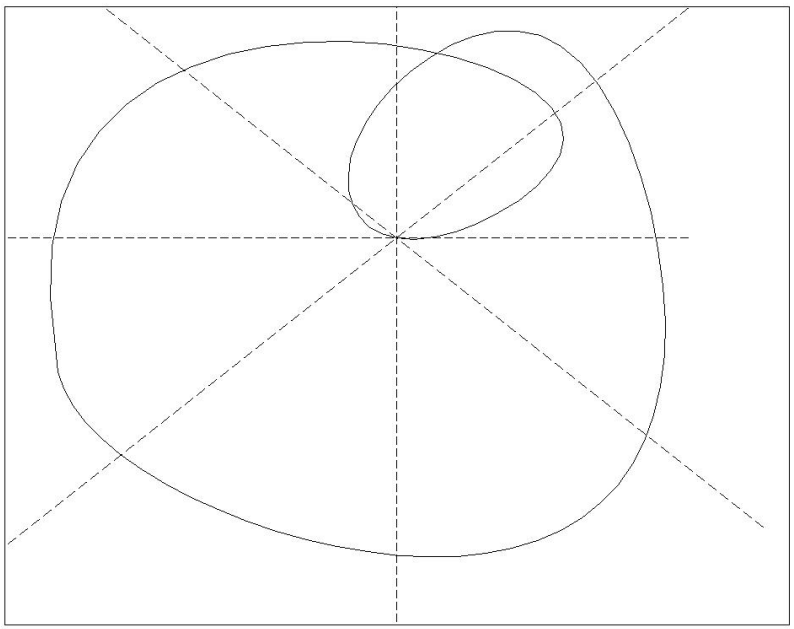
Este procedimiento tiene dos problemas:

Primer problema de YK: Bucle infinito

En curvas singulares (es decir, que pasan por el origen) el procedimiento de inserción no acaba. Estas curvas no tienen muestreo conexo.

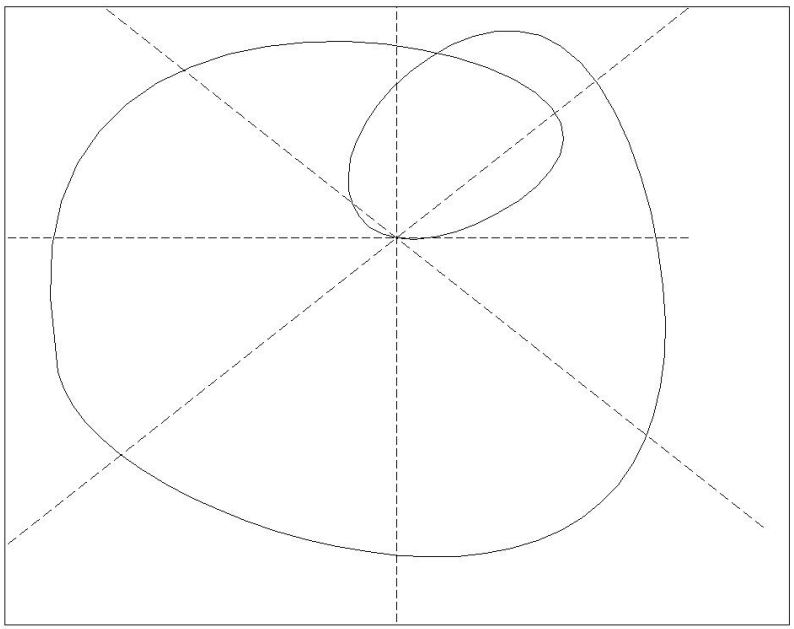
Primer problema de YK: Bucle infinito

En curvas singulares (es decir, que pasan por el origen) el procedimiento de inserción no acaba. Estas curvas no tienen muestreo conexo.



Primer problema de YK: Bucle infinito

En curvas singulares (es decir, que pasan por el origen) el procedimiento de inserción no acaba. Estas curvas no tienen muestreo conexo.

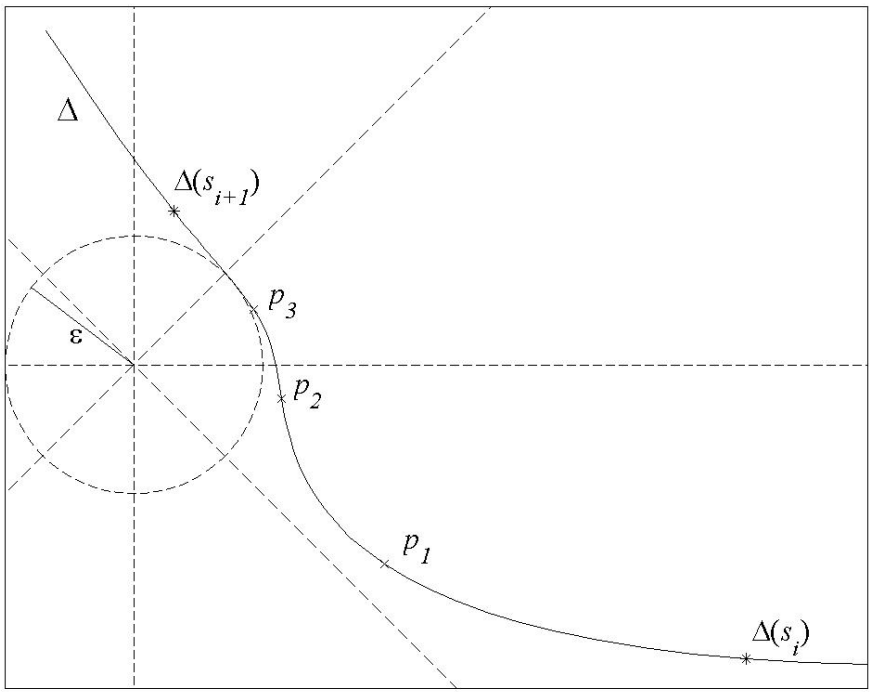


El índice está definido solo para curvas a una distancia $\varepsilon > 0$ del origen.

Una curva ε-*singular* es la que está a distancia mayor de ε del origen.

Primer problema de YK: Bucle infinito

En curvas singulares (es decir, que pasan por el origen) el procedimiento de inserción no acaba. Estas curvas no tienen muestreo conexo.

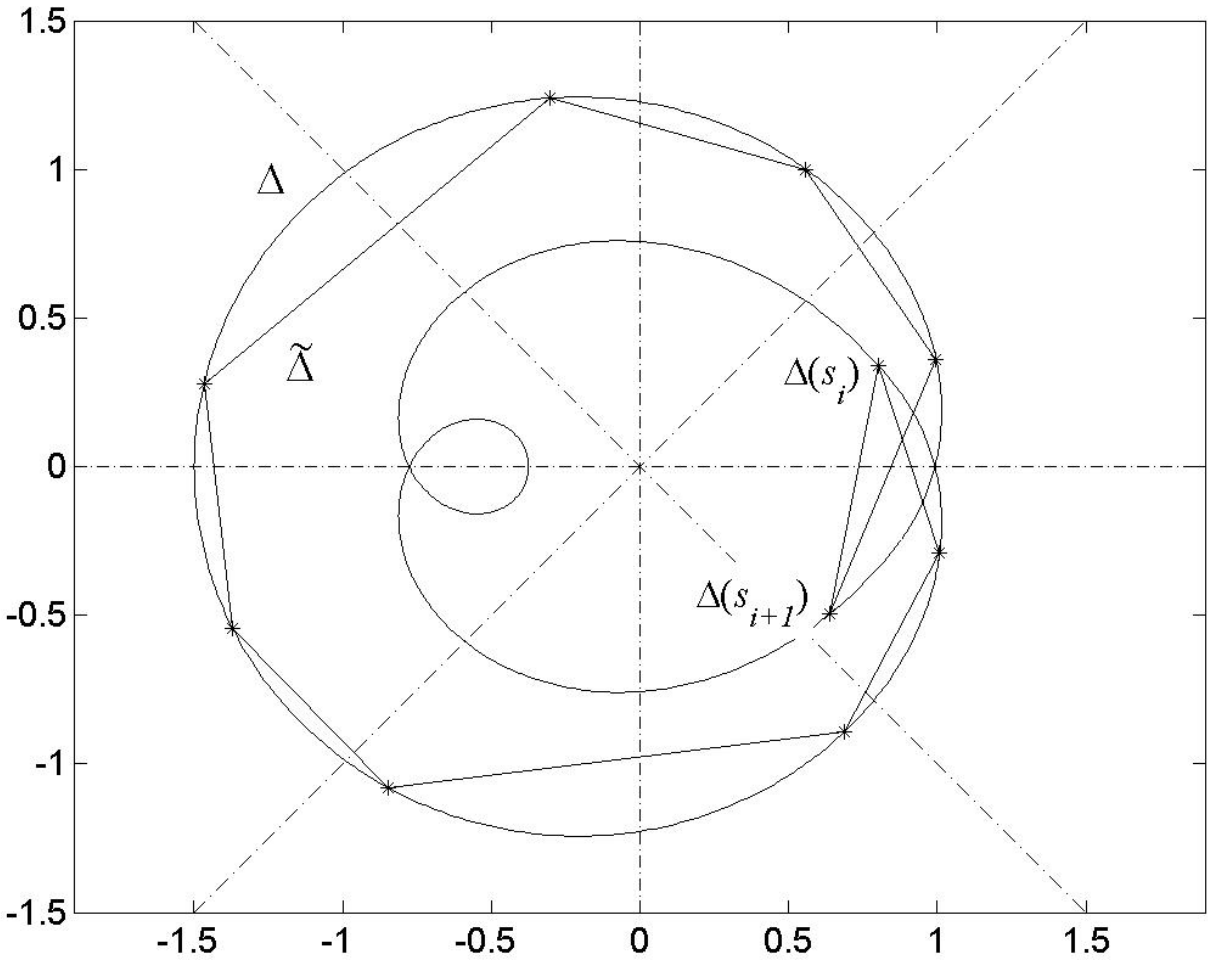


El índice está definido solo para curvas a una distancia $\varepsilon > 0$ del origen.

Una curva ε-*singular* es la que está a distancia mayor de ε del origen.

Segundo problema de YK: Giros Perdidos

Aunque termine, la poligonal producida no da el índice si hay *giros perdidos*.



Para evitar estos inconvenientes, definimos el Procedimiento de Inserción con Control de la Singularidad (PICS).

Para evitar estos inconvenientes, definimos el Procedimiento de Inserción con Control de la Singularidad (PICS).

Una curva $\Delta : [a, b] \to \mathbb{C}$ es *Lipschitziana de constante* L si verifica $|\Delta(s) - \Delta(t)| \leq L|s - t|$.

Para evitar estos inconvenientes, definimos el Procedimiento de Inserción con Control de la Singularidad (PICS).

Una curva $\Delta : [a,b] \to \mathbb{C}$ es *Lipschitziana de constante* L si verifica $|\Delta(s) - \Delta(t)| \leq L|s-t|$.

PICS utiliza los siguiente asertos sobre los valores de un muestreo $S = (a, \ldots, s_i, s_{i+1}, \ldots, b)$, y una curva $\Delta = f(\Gamma)$.

- El aserto $p(s_i)$ es "los puntos $\Delta(s_i)$ y $\Delta(s_{i+1})$ están en sectores no conectados".
- El aserto $q(s_i)$ "los valores s_i y s_{i+1} en la secuencia S verifican:

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

Se tiene:

$$\neg p(s_i) \text{ para todo } i \implies \text{el muestreo es conexo}$$
$$\neg q(s_i) \text{ para todo } i \implies \text{no hay giros perdidos en } \Delta = f(\Gamma)$$
$$\neg p(s_i) \text{ y } \neg q(s_i) \text{ para todo } i \implies \text{cruces de } \widetilde{\Delta} = \mathrm{Ind}(\Delta)$$

Se tiene:

$$\neg p(s_i) \text{ para todo } i \implies \text{el muestreo es conexo}$$
$$\neg q(s_i) \text{ para todo } i \implies \text{no hay giros perdidos en } \Delta = f(\Gamma)$$
$$\neg p(s_i) \text{ y } \neg q(s_i) \text{ para todo } i \implies \text{cruces de } \widetilde{\Delta} = \mathrm{Ind}(\Delta)$$

- El aserto $r(s_i, Q)$ (para un valor $Q > 0$) es "los valores s_i y s_{i+1} verifican $|s_{i+1} - s_i| \leq Q$".

Se tiene:

$$\neg p(s_i) \text{ para todo } i \implies \text{el muestreo es conexo}$$
$$\neg q(s_i) \text{ para todo } i \implies \text{no hay giros perdidos en } \Delta = f(\Gamma)$$
$$\neg p(s_i) \text{ y } \neg q(s_i) \text{ para todo } i \implies \text{cruces de } \widetilde{\Delta} = \mathrm{Ind}(\Delta)$$

- El aserto $r(s_i, Q)$ (para un valor $Q > 0$) es "los valores s_i y s_{i+1} verifican $|s_{i+1} - s_i| \leq Q$".

PICS

Mientras($p(s_i)$ o $q(s_i)$ para algún i)
($\text{Insertar } \frac{s_i + s_{i+1}}{2} \text{ en } S \text{ entre } s_i \text{ y } s_{i+1}$;
Si $r(s_i, Q)$ entonces salir con error)

$(p(s_i)$ o $q(s_i))$ y $r(s_i, Q)$ para algún $i \implies$ hay una raíz cercana a Γ

Se tiene:

$$\neg p(s_i) \text{ para todo } i \implies \text{el muestreo es conexo}$$
$$\neg q(s_i) \text{ para todo } i \implies \text{no hay giros perdidos en } \Delta = f(\Gamma)$$
$$\neg p(s_i) \text{ y } \neg q(s_i) \text{ para todo } i \implies \text{cruces de } \widetilde{\Delta} = \mathrm{Ind}(\Delta)$$

- El aserto $r(s_i, Q)$ (para un valor $Q > 0$) es "los valores s_i y s_{i+1} verifican $|s_{i+1} - s_i| \leq Q$".

PICS

Mientras($p(s_i)$ o $q(s_i)$ para algún i)
($\text{Insertar } \frac{s_i + s_{i+1}}{2} \text{ en } S \text{ entre } s_i \text{ y } s_{i+1}$;
Si $r(s_i, Q)$ entonces salir con error)

($p(s_i)$ o $q(s_i)$) y $r(s_i, Q)$ para algún $i \implies$ hay una raíz cercana a Γ

Para un polinomio f de raíces z_i, definimos el número de condición:

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

Para un polinomio f de raíces z_i, definimos el número de condición:

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

Teorema [García Zapata and Díaz Martín, 2014]: Si $\Gamma : [a, b] \to \mathbb{C}$ es Lipschitziana, entonces PICS para la curva $\Delta = f(\Gamma)$:

- Concluye en menos de $\lfloor \frac{b-a}{Q} + 1 \rfloor$ iteraciones.
- Si acaba normalmente da correctamente el número de raíces de f dentro de Γ.
- Si acaba con error, entonces $\kappa_f(\Gamma) \geq \frac{2-\sqrt{2}}{4Q}$.

Para un polinomio f de raíces z_i, definimos el número de condición:

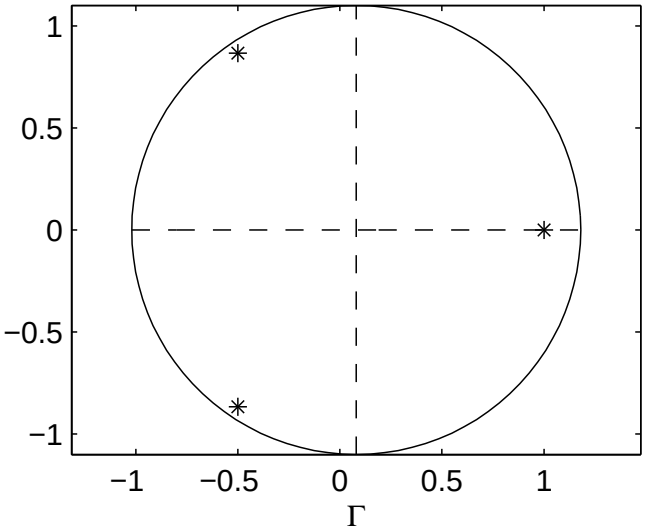
$$\kappa_f(\Gamma) = \sum_{i=1}^{n} \frac{1}{\mathrm{d}(z_i, \Gamma)}.$$

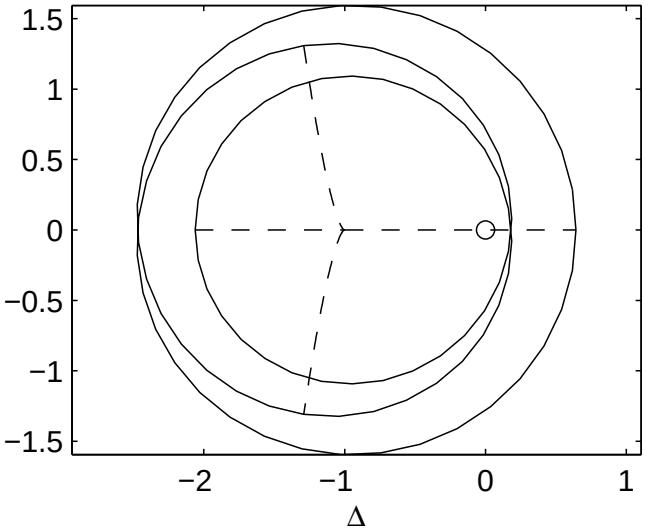
Teorema [García Zapata and Díaz Martín, 2014]: Si $\Gamma : [a, b] \to \mathbb{C}$ es Lipschitziana, entonces PICS para la curva $\Delta = f(\Gamma)$:

- Concluye en menos de $\lfloor \frac{b-a}{Q} + 1 \rfloor$ iteraciones.
- Si acaba normalmente da correctamente el número de raíces de f dentro de Γ.
- Si acaba con error, entonces $\kappa_f(\Gamma) \geq \frac{2-\sqrt{2}}{4Q}$.

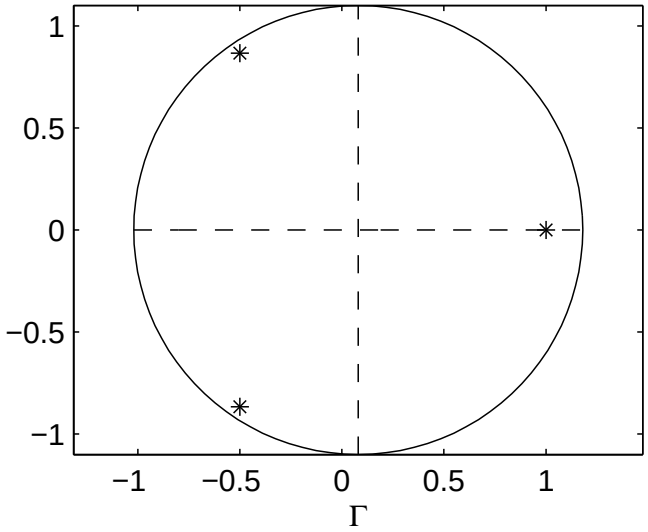
Esa cota en error implica que hay una raíz a menos de $\frac{4nQ}{2-\sqrt{2}}$ de Γ.

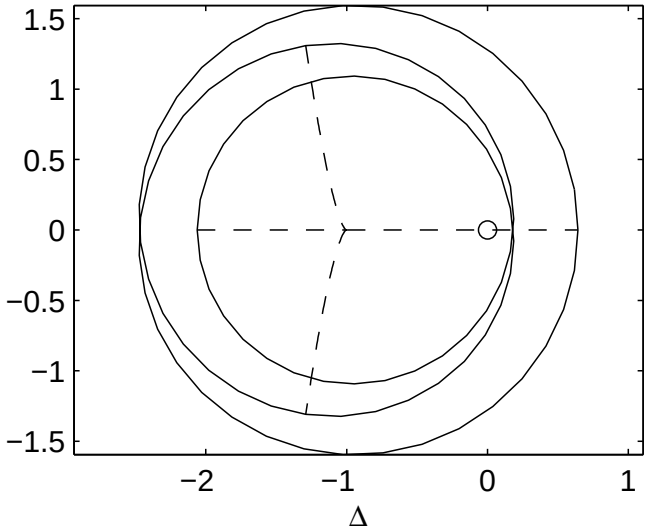
La salida con error no calcula el índice, pero da información sobre la singularidad de Δ. Esto será crucial en la descomposición recursiva:

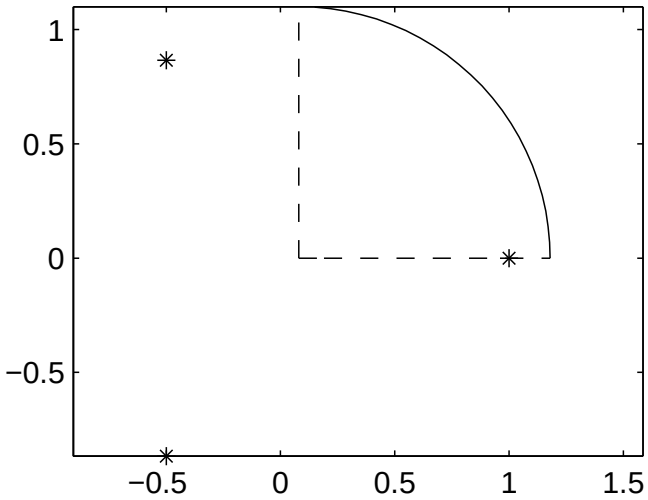


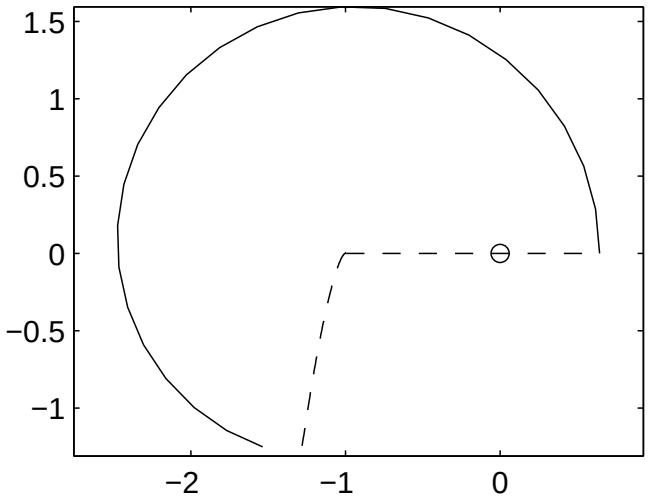


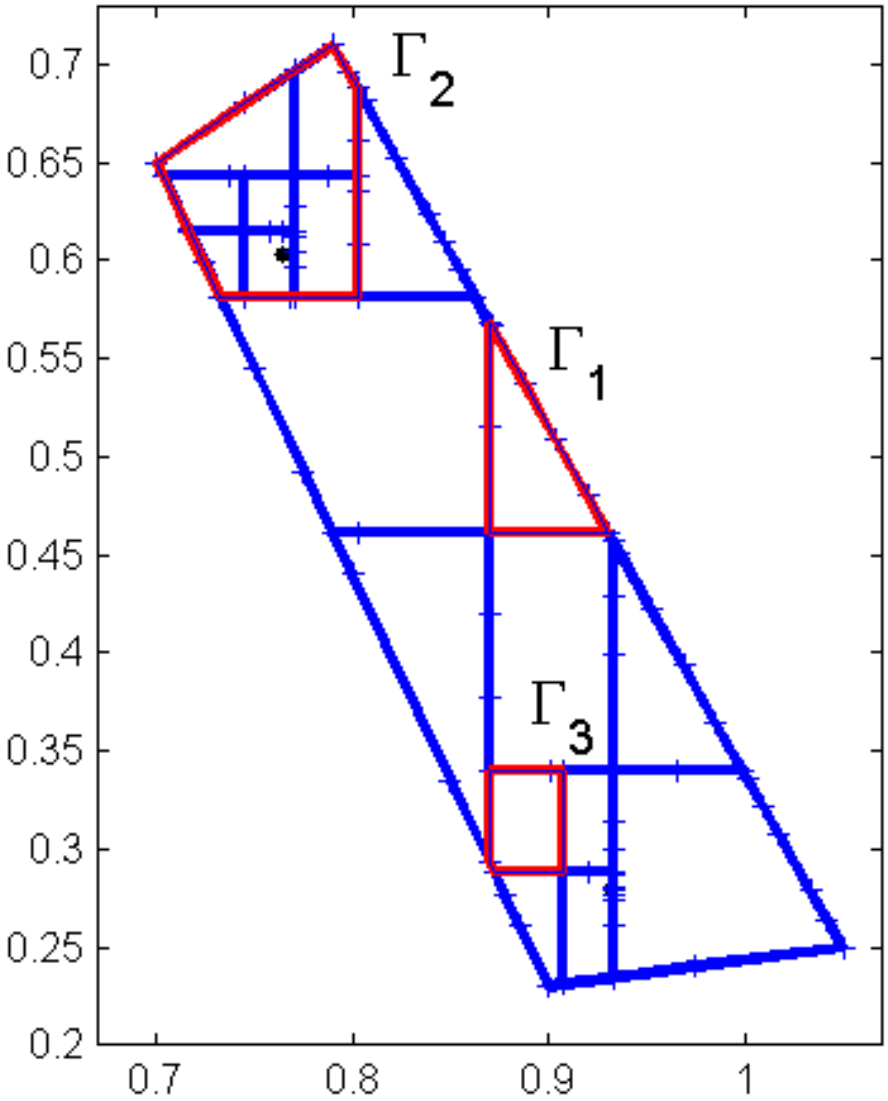
La salida con error no calcula el índice, pero da información sobre la singularidad de Δ. Esto será crucial en la descomposición recursiva:











Cada región está definida por su borde Γ.

1. ¿Cómo calcular numéricamente el índice de cada $f(\Gamma)$?
2. ¿Cómo descomponer las regiones con índice no nulo?

Procedimiento Recursivo de División (PRec)

Halla las raíces de f contenidas en una región convexa P_I. Las listas Π y N van a contener las raíces que se vayan hallando, y sus multiplicidades, con una precisión A.

Procedimiento Recursivo de División (PRec)

Halla las raíces de f contenidas en una región convexa P_I. Las listas Π y N van a contener las raíces que se vayan hallando, y sus multiplicidades, con una precisión A.

PRec (sin gestión de errores)

PRec(P, A) $\{$

Si iTest$(P) = 0$, retornar Π y N vacías. [Salida 1]

Si $\mathrm{dm}(P) < A$ 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]

$\}$

Procedimiento Recursivo de División (PRec)

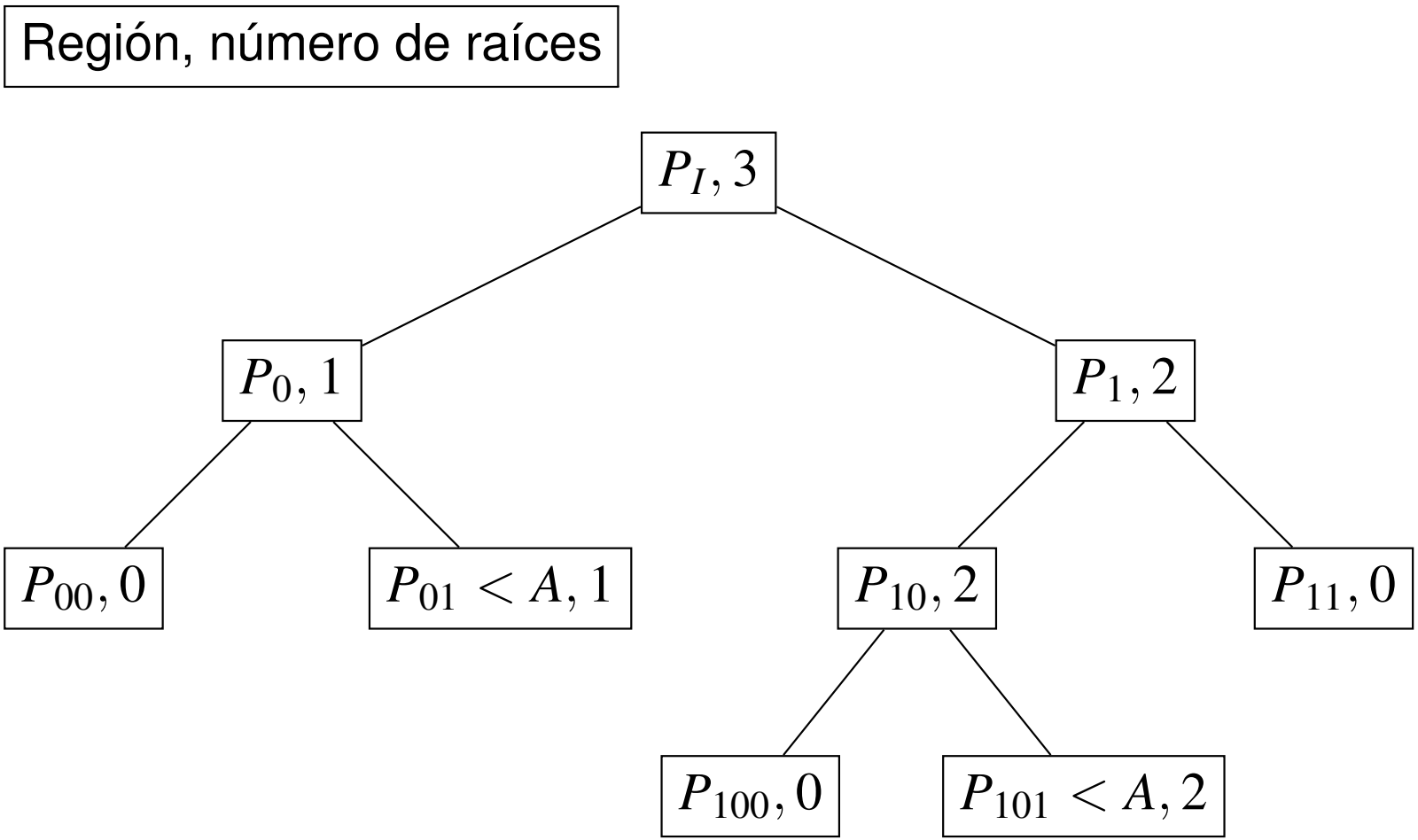
Halla las raíces de f contenidas en una región convexa P_I. Las listas Π y N van a contener las raíces que se vayan hallando, y sus multiplicidades, con una precisión A.

PRec (sin gestión de errores)

PRec(P, A) $\{$
- Si iTest$(P) = 0$, retornar Π y N vacías. [Salida 1]
- Si $\mathrm{dm}(P) < A$ 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]

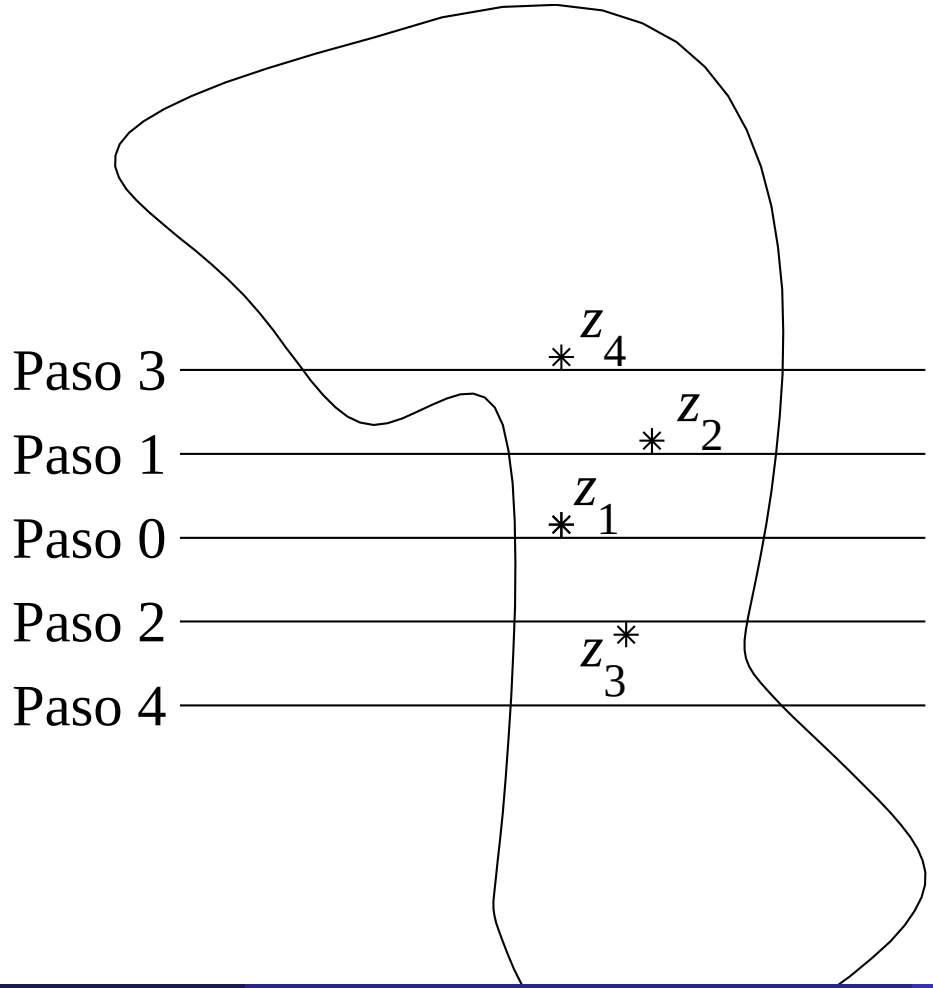
$\}$

Ejemplo de ejecución



Evitando singularidades

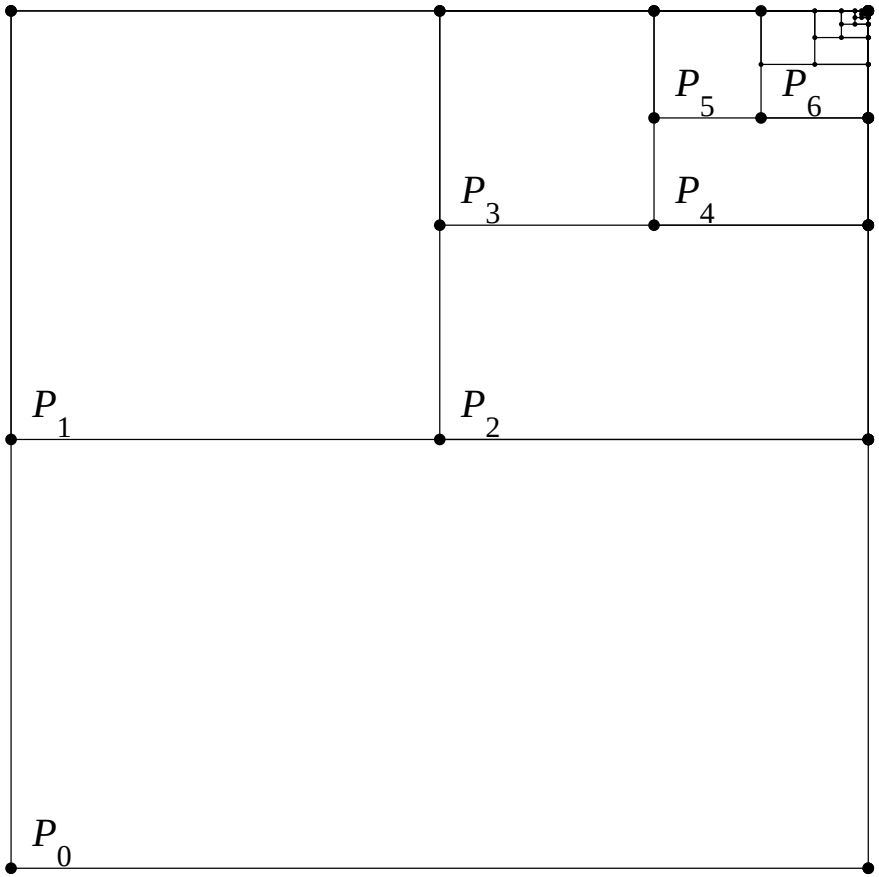
Cortes Iterados con Desplazamiento (CID): Si iTest vuelve con error, se desplaza el corte para evitar la raíz cercana.



La tolerancia μ es el máximo desplazamiento alcanzado

Disminuyendo el diámetro

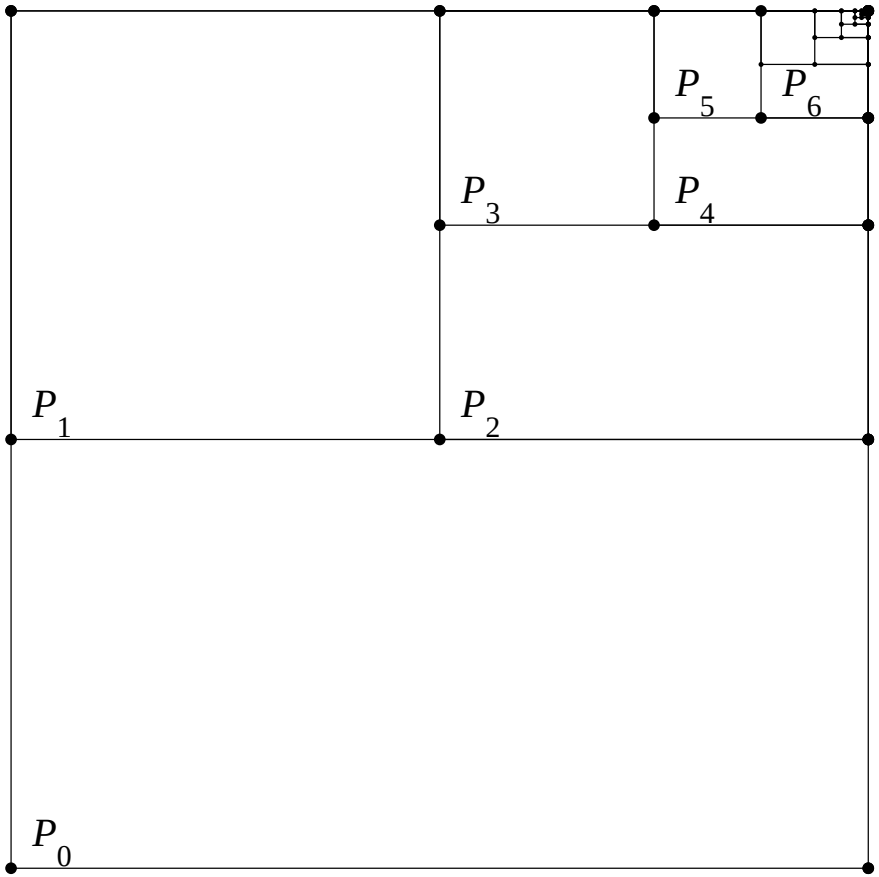
Alternando cortes H y V con desplazamiento, se dispara el aspecto:



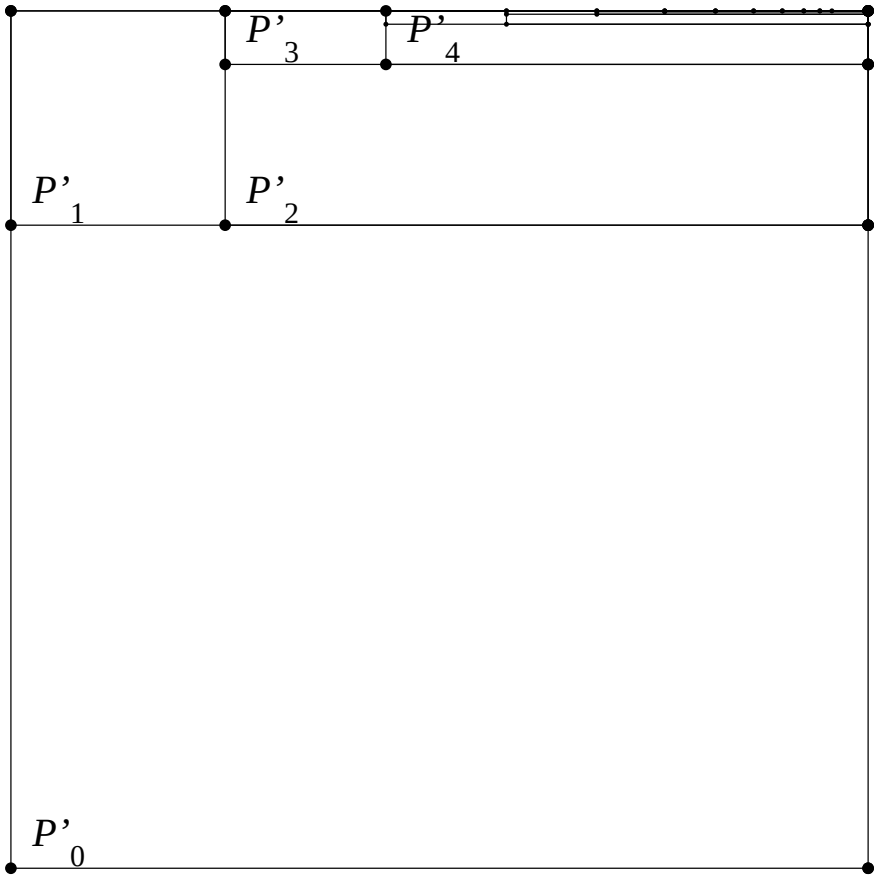
Sin desplazamiento

Disminuyendo el diámetro

Alternando cortes H y V con desplazamiento, se dispara el aspecto:



Sin desplazamiento

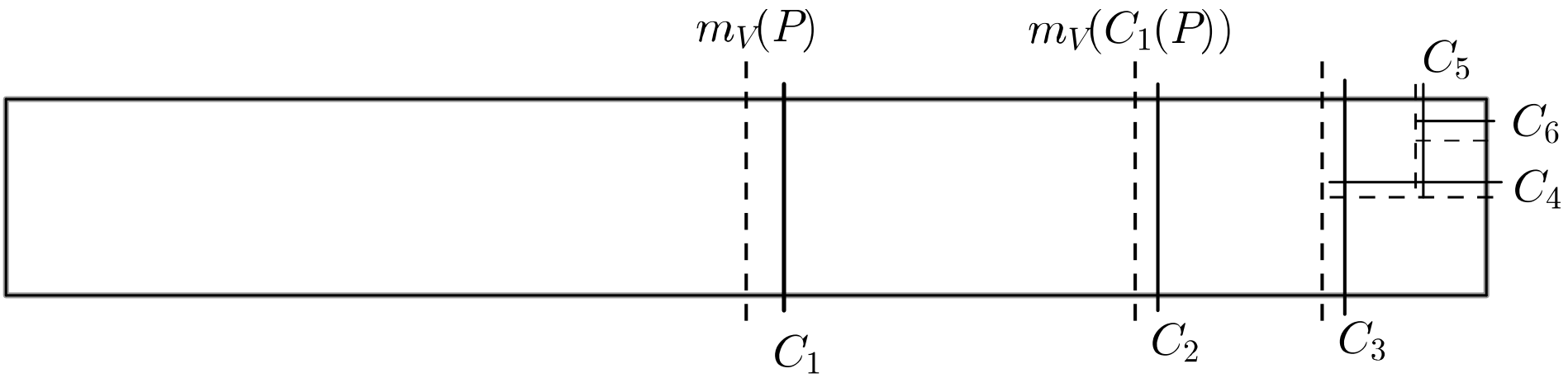


Con desplazamiento ($\mu = 1/2$)

Disminuyendo el diámetro

El aspecto $\mathrm{asp}(P)$ en función del ancho y alto de P es $\dfrac{\text{máx}(\text{ancho,alto})}{\text{mín}(\text{ancho,alto})}$.

Con cortes a lo largo del eje menor, se mantiene acotado el aspecto:



Teorema [García Zapata and Díaz Martín, Rev.]: Para un polinomio f y región P_I sin raíces cerca del borde, PRec(P_I, A) alcanza una profundidad de recursión como mucho de

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

y acaba correctamente.

Teorema [García Zapata and Díaz Martín, Rev.]: Para un polinomio f y región P_I sin raíces cerca del borde, PRec(P_I, A) alcanza una profundidad de recursión como mucho de

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

y acaba correctamente.

El resultado tiene demostración complicada por la interrelación de Q (en el aserto $r(s_i, Q)$ de PICS) con la tolerancia μ de los cortes producidos por CID.

Teorema [García Zapata and Díaz Martín, Rev.]: Para un polinomio f y región P_I sin raíces cerca del borde, PRec(P_I, A) alcanza una profundidad de recursión como mucho de

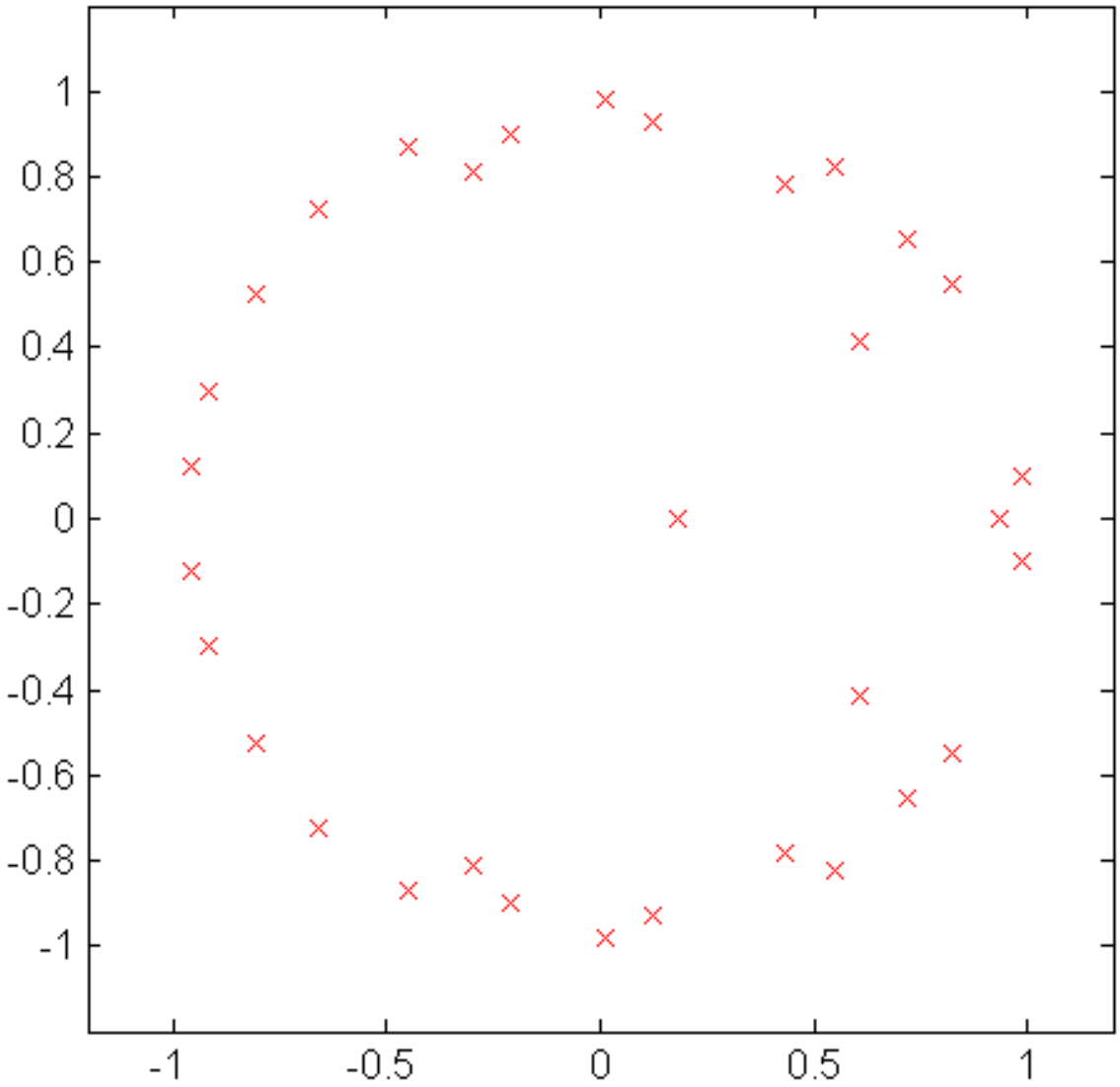
$$l_{\text{máx}} = \left\lceil \frac{6 - \lg_2 3}{(2 - \lg_2 3)^2} \lg_2 \frac{\text{dm}(P_I)}{A} + \frac{\lg_2(\text{asp}(P_I))}{2 - \lg_2 3} + 2 \right\rceil .$$

y acaba correctamente.

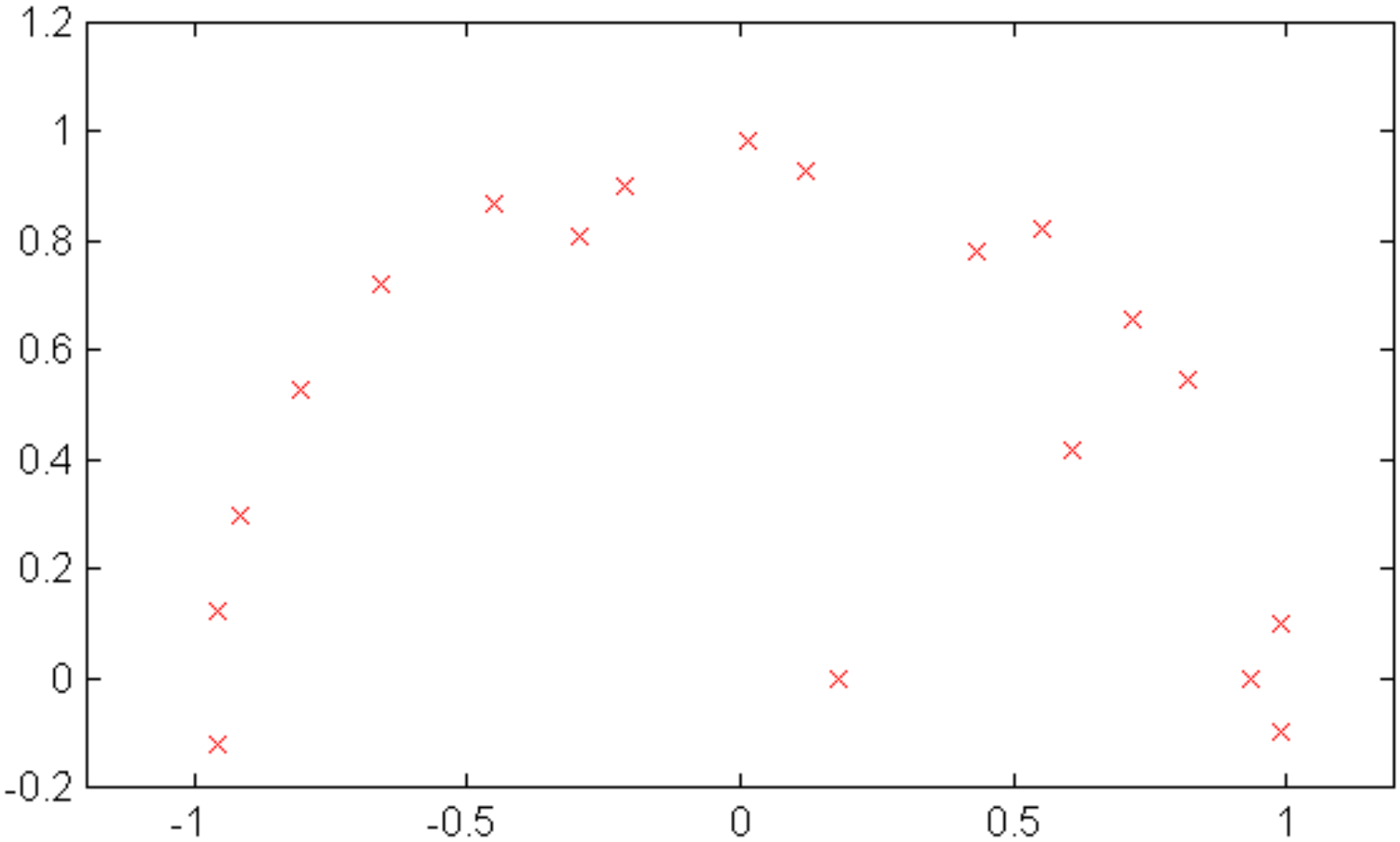
El resultado tiene demostración complicada por la interrelación de Q (en el aserto $r(s_i, Q)$ de PICS) con la tolerancia μ de los cortes producidos por CID.

Con esto se consigue el subobjetivo 2 “Descomposición recursiva”.

Método Contour (Círculo unidad)



Método Contour (Corona)



Hitos obtenidos:

- Cálculo del índice con ε-singularidad, y número de condición.
- Descomposición recursiva con efectividad demostrada, y coste.
- Implementación y biblioteca de pruebas.

Hitos obtenidos:

- Cálculo del índice con ε-singularidad, y número de condición.
- Descomposición recursiva con efectividad demostrada, y coste.
- Implementación y biblioteca de pruebas.

Trabajo futuro:

- Implementación paralela.
- Aplicaciones.

Referencias

[García Zapata and Díaz Martín, 2012] *A geometrical root finding method for polynomials, with complexity analysis.* Journal of Complexity, 28:320-345.

[García Zapata and Díaz Martín, 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.

[García Zapata and Díaz Martín, Rev.] *A Geometrical Root Finding Method for Polynomials.*

Muchas gracias