

**UNIVERSIDAD DE PANAMÁ
FACULTAD DE CIENCIAS NATURALES Y EXACTAS
ESCUELA DE MATEMÁTICA
CENTRO REGIONAL UNIVERSITARIO DE VERAGUAS**

**SOLUCIÓN DE ECUACIONES ALGEBRAICAS
POR MÉTODOS ITERATIVOS, EMPLEANDO
COMPUTADORAS**

POR

**OLGA ESTER BATISTA GONZÁLEZ
Y
RAÚL ENRIQUE DUTARI DUTARI**

**TRABAJO DE GRADUACIÓN, PRESENTADO
A LA CONSIDERACIÓN DE LA ESCUELA,
PARA OPTAR POR EL TÍTULO DE
LICENCIADOS EN MATEMÁTICA.**

**SANTIAGO, REPÚBLICA DE PANAMÁ
1991**

DEDICATORIA

*Al ser supremo que nunca falla:
Dios, por haber dado la perfección a
toda su creación, en particular al hombre
cuya capacidad de conocer es
inagotable.*

*Con especial cariño a mis padres:
Juan y Eugenia, quienes depositaron su
confianza en mí, brindándome el apoyo
moral y económico durante mis años de
estudio para dar feliz término a una de
mis aspiraciones.*

*A mis hermanas: Omaira,
Manuela, y Judith. También a Raúl,
compañero desde mi primer año de
estudio universitario. Ellos supieron
ayudarme y comprenderme ante
cualquier situación difícil.*

OLGA

Al supremo creador, porque sin su apoyo es imposible lograr algo en esta vida.

A mis padres, Juan B. y María E., quienes con sus esfuerzos, apoyo y amor me alentaron para alcanzar la realización de uno de mis más preciados sueños.

A Vivian Vadina Alvarado, quién a través de la distancia y el tiempo, es la voz que, a lo lejos, me anima y guía a través de la soledad, la oscuridad y lo desconocido.

RAÚL

AGRADECIMIENTO

Expresamos nuestra profunda gratitud al profesor José del Carmen Ochoa, nuestro amigo y asesor, quién demostró siempre su confianza en nuestra capacidad para confeccionar este trabajo.

También nos sentimos profundamente agradecidos por la irrestricta colaboración que recibimos del profesor Amílcar Díaz T., quién nos facilitó la adquisición del software que necesitamos como entorno inicial para crear los programas de este trabajo.

Además, agradecemos profundamente a todas aquellas personas que, de una u otra forma, han contribuido a que este trabajo de graduación sea una realidad.

OLGA Y RAÚL

TABLA DE CONTENIDOS

DEDICATORIA.....	II
AGRADECIMIENTO.....	V
TABLA DE CONTENIDOS	VII
INTRODUCCIÓN.....	1
1. MÉTODOS ANALÍTICOS PARA ENCONTRAR LAS RAÍCES DE ECUACIONES ALGEBRAICAS.....	7
1.1. OBSERVACIONES PRELIMINARES.	8
1.2. ¿QUÉ SIGNIFICA RESOLVER UNA ECUACIÓN MEDIANTE FÓRMULAS O RADICALES?.....	9
1.3. SOLUCIÓN DE ECUACIONES LINEALES, CUADRÁTICAS, CÚBICAS Y CUÁRTICAS.	9
1.3.1. ECUACIÓN CUADRÁTICA.	10
1.3.1.1. DEDUCCIÓN DE LA FÓRMULA CUADRÁTICA.	11
1.3.2. ECUACIÓN CÚBICA.	12
1.3.2.1. DEDUCCIÓN DE LA FÓRMULA CÚBICA.	12
1.3.2.2. DISCUSIÓN DE LAS FÓRMULAS DE CARDANO.....	19
1.3.2.3. CASO IRREDUCIBLE: RESOLUCIÓN TRIGONOMÉTRICA.	21
1.3.3. ECUACIÓN CUÁRTICA.	23
1.3.3.1. DEDUCCIÓN DE LA FÓRMULA CUÁRTICA.	23

1.3.3.2.	ACLARACIÓN EN LA RESOLUCIÓN DE LA ECUACIÓN CUÁRTICA.....	28
1.4.	IMPOSIBILIDAD DE LA RESOLUCIÓN DE UNA ECUACIÓN DE GRADO MAYOR QUE CUATRO, POR RADICALES O FÓRMULAS.....	34
2.	MÉTODOS ITERATIVOS PARA ENCONTRAR LAS RAÍCES DE ECUACIONES ALGEBRAICAS.....	36
2.1.	OBSERVACIONES PRELIMINARES.	37
2.2.	MÉTODOS ITERATIVOS QUE NOS PERMITEN CALCULAR LAS RAÍCES REALES DE UNA ECUACIÓN ALGEBRAICA. ACOTACIÓN SUCESIVA DE LAS RAÍCES.....	39
2.2.1.	ACOTACIÓN DE RAÍCES.....	39
2.2.1.1.	MÉTODO DE LAGUERRE-THIBAUT.....	43
2.2.1.2.	MÉTODO DEL MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN.	44
2.2.1.3.	MÉTODO DE NEWTON.....	47
2.2.1.4.	MÉTODO ALTERNATIVO.	49
2.2.2.	SEPARACIÓN DE RAÍCES.	49
2.2.2.1.	TEOREMA DEL VALOR INTERMEDIO.....	54
2.2.2.1.1.	TEOREMA DE BISECCIÓN DE BOLZANO.....	58
2.2.2.1.2.	PROCEDIMIENTO DE BISECCIÓN.....	58

2.2.2.2.	MÉTODO DE ROLLE.....	60
2.2.2.3.	MÉTODO DE STURM.....	65
2.2.2.4.	MÉTODOS ALTERNATIVOS.....	75
2.2.3.	APROXIMACIÓN DE RAÍCES REALES.	77
2.2.3.1.	MÉTODO DE BOLZANO.	79
2.2.3.2.	MÉTODO DE RUFFINI-HORNER.....	85
2.2.3.3.	MÉTODO DE APROXIMACIONES SUCESIVAS (ITERACIÓN DE PUNTO FIJO).....	96
2.2.3.4.	MÉTODO DE NEWTON-RAPHSON-FOURIER.	113
2.2.3.5.	MÉTODOS ALTERNATIVOS.....	133
2.3.	COMPORTAMIENTO PRÁCTICO DE LOS MÉTODOS ITERATIVOS AL CALCULAR RAÍCES.....	139
3.	PROGRAMACIÓN Y COMPARACIÓN DE LOS MÉTODOS ITERATIVOS PARA ENCONTRAR LAS RAÍCES DE ECUACIONES ALGEBRAICAS.....	142
3.1.	OBSERVACIONES PRELIMINARES.	143
3.2.	VENTAJAS Y RESTRICCIONES QUE PRESENTA LA APLICACIÓN DEL USO DE COMPUTADORAS EN LA SOLUCIÓN DE NUESTRO PROBLEMA.....	145
3.3.	ALGUNOS ASPECTOS IMPORTANTES ACERCA DE LA TEORÍA DEL ERROR.....	148

3.4.	ALGUNAS LIMITACIONES QUE SE PRESENTAN AL TRATAR DE DETERMINAR EL MEJOR MÉTODO.	151
3.5.	CONVENCIONES EN CUANTO A LA IMPLEMENTACIÓN COMPUTACIONAL DE LOS MÉTODOS DESARROLLADOS.	153
3.6.	DESARROLLO DE LOS MÉTODOS SEGÚN EL PLAN TRAZADO.	154
3.6.1.	MÉTODOS QUE ACOTAN RAÍCES REALES.	155
3.6.1.1.	MÉTODO DE LAGUERRE-THIBAUT.	156
3.6.1.2.	MÉTODO DEL MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN.	163
3.6.1.3.	MÉTODO DE NEWTON.	170
3.6.1.4.	MÉTODO MIXTO DE ACOTACIÓN: MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN-NEWTON.	178
3.6.2.	MÉTODOS QUE SEPARAN RAÍCES REALES.	185
3.6.2.1.	MÉTODO DE STURM.	187
3.6.2.2.	MÉTODO MIXTO DE SEPARACIÓN: MÉTODO MIXTO DE ACOTACIÓN-STURM.	206
3.6.3.	MÉTODOS QUE APROXIMAN RAÍCES REALES.	225
3.6.3.1.	MÉTODO DE BISECCIÓN DE BOLZANO.	229
3.6.3.2.	MÉTODO DE NEWTON-RAPHSON-FOURIER.	245

3.6.3.3.	MÉTODO MIXTO 1 DE APROXIMACIÓN: MÉTODO MIXTO DE SEPARACIÓN-BOLZANO.	263
3.6.3.4.	MÉTODO MIXTO 2 DE APROXIMACIÓN: MÉTODO MIXTO DE SEPARACIÓN-NEWTON-RAPHSON-FOURIER.	284
3.7.	VENTAJAS, DESVENTAJAS Y LIMITACIONES DE LOS MÉTODOS DESARROLLADOS, EN LA SOLUCIÓN DE PROBLEMAS PARTICULARES.	306
3.8.	COMENTARIOS FINALES.	309
	CONCLUSIONES.....	314
	RECOMENDACIONES	317
	ANEXOS	320
	ANEXO #1: APÉNDICE DE TEORÍA DE ECUACIONES.....	321
	ANEXO #2: APÉNDICE DE CÁLCULO.....	331
	ANEXO #3: LISTA DE SÍMBOLOS	339
	BIBLIOGRAFÍA	343

“La popularidad y crecimiento explosivo del análisis numérico en la época actual demuestra una vez más que las aplicaciones prácticas siguen siendo la fuente principal de inspiración para la creatividad matemática. Cuando quiera que surge en este campo una idea nueva, se encontrará que generalmente obedece a una necesidad de la época. La computadora electrónica es un buen ejemplo: fue la respuesta a la necesidad inaplazable de cálculos más rápidos. A su vez la aparición de tales máquinas ha hecho posible satisfacer las demandas de las aplicaciones actuales, desarrollando en muchos casos métodos numéricos más perfeccionados. Esta es la genealogía del análisis numérico moderno; es el aspecto numérico del amplio campo del análisis aplicado. ...”

FRANCIS SCHEID

INTRODUCCIÓN

A lo largo de nuestra vida universitaria, la resolución de ecuaciones algebraicas se nos ha presentado como un problema constante, que —en el caso general— pocas veces ha tenido una solución simple, y, sin embargo, es un paso importantísimo en la solución de una cantidad innumerable de problemas matemáticos, físicos y de otras áreas.

Además, cuando recibimos nuestros cursos de programación I y II; nos encontramos con que ésta es un área donde no existe un método definido para la resolución de las mencionadas ecuaciones, ya que cada uno de ellos presenta deficiencias y limitaciones que mediatizan su acción, con todo y que ellas, vistas como grupo de funciones, han recibido un estudio especial por parte de los matemáticos. Esto ocurre hoy día, en detrimento de la gran cantidad de métodos que existen actualmente para resolverlas. Cada uno de ellos presenta características favorables y desfavorables para su empleo en la solución de determinadas ecuaciones. Estos hechos, aunados a nuestra inclinación hacia la programación con computadoras, son los que nos han motivado a escoger para este trabajo un tema relacionado con el problema de obtener raíces de ecuaciones.

Este trabajo se concretará al desarrollo e implementación de métodos iterativos para la obtención de raíces de ecuaciones algebraicas en general; serán aplicados únicamente a ecuaciones polinomiales; y, finalmente, su aplicación práctica será automatizada a través del empleo de programas para computadoras.

Mientras nos dedicábamos a la recopilación de bibliografía para confeccionar el trabajo, nos enteramos de que ésta se encuentra relativamente dispersa, ya que la mayoría de las obras consultadas tratan el problema global en forma fraccionaria, sin tratar de resolverlo satisfactoriamente, de modo que el estudioso del tema se encuentra inicialmente enfrentado a una marejada de

métodos, que no sabe cuando debe o no aplicar. Esta deficiencia hemos tratado de salvarla, optando por estudiar el problema desde sus orígenes, y realizando las construcciones en forma tan detallada y precisa como nos ha sido posible.

Precisamente el origen acerca del tema de nuestro trabajo de graduación, así como los conocimientos que previamente teníamos acerca del problema, han marcado la tónica con respecto al desenvolvimiento del trabajo. El mismo, ha sido planteado de modo que se presenta como una recopilación constructiva de algunos de los métodos y técnicas más conocidos y prácticos, que existen para resolver dichas ecuaciones, así como algunos programas que hemos confeccionado con el fin de automatizar, hasta donde nos ha sido posible, la solución de nuestro problema.

El primer capítulo de nuestro trabajo es de tipo introductorio y presupondrá conocimientos de la teoría de ecuaciones algebraicas. En él, analizaremos el problema planteado desde el punto de vista algebraico, resolviendo las ecuaciones cuadrática, cúbica y cuártica. Luego, al considerar el hecho probado dentro de las aplicaciones de la “teoría de Galois” consistente en que las ecuaciones de grado mayor que el cuarto, no son resolubles por radicales; dejaremos al descubierto que este camino no brinda una solución —ni factible ni práctica— para nuestro problema —contrario a lo que comúnmente se cree—.

Este capítulo tendrá un carácter netamente expositivo e introductorio al tema en cuestión, al considerar aspectos que actualmente son problemas resueltos de la matemática.

En cambio, el segundo capítulo presenta un desenvolvimiento poco conocido —desde el punto de vista práctico— para resolver el problema. En él, desarrollaremos algunos de los métodos iterativos más conocidos en nuestro

medio, para la solución de ecuaciones algebraicas; concentrándose este estudio sobre la obtención de raíces reales de polinomios reales. Este capítulo presupone conocimientos básicos acerca de cálculo diferencial, así como los prerrequisitos establecidos para el primer capítulo.

La construcción teórica de los métodos la realizaremos empleando desarrollos matemáticos constructivos (geométricos y analíticos), basados fundamentalmente en hechos conocidos del análisis matemático, complementadas con teoremas relativamente poco conocidos de la teoría de ecuaciones algebraicas. Además, al final del capítulo discutiremos si los métodos iterativos previamente tratados, en verdad representan, por sí solos, la solución práctica al problema planteado.

Hemos tratado de realizar los desarrollos tan detalladamente como nos fue posible, con el objeto de hacer accesible al máximo el fundamento teórico de los métodos desarrollados. Además, con el objeto de facilitar la comprensión de los temas, hemos construido tres apéndices que complementan este trabajo con los resultados más importantes que han sido empleados de la teoría de ecuaciones, y el cálculo diferencial; además, añadimos la lista de símbolos más usualmente empleados.

Finalmente, en el tercer capítulo, empezaremos justificando el empleo de las computadoras, en nuestro esfuerzo por resolver el problema central de este trabajo, destacando sus características positivas y negativas al respecto, así como sus limitaciones. Luego, basándonos en esa discusión, replantearemos el problema inicial, para tratar de automatizar las soluciones aproximadas que hemos obtenido en teoría (tales como: el procedimiento a seguir para desarrollar un programa de computadora y aspectos relativos a la teoría del error aplicados a la computación).

Para tal efecto, desarrollaremos programas en el lenguaje de programación BASIC de algunos de los métodos antes desarrollados. Posteriormente, estos métodos se compararán entre sí bajo diversos criterios, de modo que sean evidentes las situaciones en que se puedan emplear eficientemente. Basándonos en estos resultados, obtendremos combinaciones de dichos métodos, que sean mejores que los desarrollados individualmente.

Luego, dejaremos claramente establecidas las ventajas, desventajas y limitaciones que caracterizan a los programas desarrollados. Finalizaremos el capítulo realizando una breve exposición introductoria acerca de la obtención de las raíces imaginarias de polinomios complejos, mencionando algunos de los métodos equivalentes a los desarrollados a lo largo del segundo capítulo, con el objeto de motivar a los lectores de este trabajo para que realicen estudios similares en dicha área.

El tercer capítulo presupone que el lector tiene conocimientos previos del lenguaje de programación BASIC, así como de las técnicas que se deben implementar para elaborar programas de computadoras. El contenido del trabajo no contempla estos tópicos, ya que hubiera aumentado exorbitantemente el volumen del mismo; sin embargo, en estos casos, remitiremos al lector a grupos de referencias bibliográficas seleccionados, a medida que desarrollamos los temas, de modo que él puede, si lo desea, documentarse al respecto.

Todas las tablas que muestran las ejecuciones de los programas han sido confeccionadas por los autores, basados exclusivamente en los resultados de las ejecuciones de los programas, y, por tanto, la información que aparece en ellas —así como en el resto del trabajo— es de nuestra absoluta responsabilidad, salvo que se indique explícitamente lo contrario.

Esperamos que nuestro trabajo sirva para que los lectores, en general, conozcan algunos de los métodos y técnicas que pueden emplear para resolver ecuaciones polinomiales, principalmente las que no poseen una factorización evidente.

1. **MÉTODOS ANALÍTICOS PARA ENCONTRAR
LAS RAÍCES DE ECUACIONES ALGEBRAICAS.**

1.1. OBSERVACIONES PRELIMINARES.

A manera de introducción al tema, podemos decir que el problema de encontrar las raíces de una ecuación algebraica de grado arbitrario, ha llamado la atención de muchas generaciones de matemáticos, entre los que se destacan: Newton, Bernoulli, Fourier, Laguerre, e incluso en la actualidad, es raro el año en que los eruditos en la materia no hacen contribuciones importantes en esta área.

Ha sido esencial en el desarrollo de las matemáticas y en la solución de muchos problemas prácticos, dar respuesta a este dilema, ya que ha conducido al hombre a extender sus sistemas de numeración paulatinamente, partiendo desde los naturales y llegando a los complejos y a otros sistemas numéricos más generales.

Además, este problema ha llevado al desarrollo de una rama muy antigua de la matemática llamada “teoría de ecuaciones”.

El tema a tratar lo centraremos, en general, en la obtención de raíces de ecuaciones algebraicas con ceros únicamente reales.

Así, haciendo memoria, diremos que se acostumbra llamar “ecuación algebraica” a toda ecuación obtenida al igualar a cero un polinomio de grado n en una variable x , es decir, $P_n(x)$ será una ecuación algebraica de grado $n > 0$, si y sólo si:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0, \quad (1.1)$$

donde $a_n \neq 0$ y los $a_i \in R$, con $i = 0, 1, \dots, n \in N$.

Se llama “raíz de una ecuación” algebraica a todo valor numérico de la variable x que verifique la igualdad **(1.1)**.

En este primer capítulo, en particular, abordaremos el desarrollo histórico de la obtención de raíces, a través de fórmulas que nos permiten resolver cualquier ecuación algebraica de grado menor o igual que cuatro, en el campo complejo.

1.2. ¿QUÉ SIGNIFICA RESOLVER UNA ECUACIÓN MEDIANTE FÓRMULAS O RADICALES?

Resolver una ecuación algebraica a través de fórmulas no es más que expresar las raíces de ésta, ya sean reales o complejas, como una combinación finita de operaciones racionales y de cálculo de diversas cantidades subradicales (en términos de los coeficientes de la ecuación). Estas operaciones pueden ser facilitadas con la aplicación de resultados obtenidos dentro de la teoría de ecuaciones.

De los resultados obtenidos en esta área de la matemática, emplearemos un grupo de proposiciones, cuyos enunciados estarán contenidos en el apéndice respectivo a teoría de ecuaciones de este trabajo. Para mayor información de dichos resultados (incluyendo algunas de las demostraciones) el lector puede consultar la bibliografía indicada en dicha sección.

1.3. SOLUCIÓN DE ECUACIONES LINEALES, CUADRÁTICAS, CÚBICAS Y CUÁRTICAS.

Es evidente que tratar de resolver una ecuación de grado cero no tiene sentido.

Encontrar la raíz de una ecuación lineal no ofrece dificultad y es ampliamente conocido que:

$$a_1x + a_0 = 0$$

tiene como raíz única a:

$$x = -\frac{a_0}{a_1}$$

$a_1 \neq 0$ por definición.

Las raíces de las ecuaciones cuadráticas, cúbicas y cuárticas se obtienen empleando procedimientos algebraicos que desarrollamos en los siguientes puntos de este capítulo.

1.3.1. ECUACIÓN CUADRÁTICA.

La ecuación:

$$a_2x^2 + a_1x + a_0 = 0 \tag{1.2}$$

se denomina cuadrática, donde los $a_i \in R$ y $a_2 \neq 0$, ésta puede ser resuelta como sigue:

1.3.1.1. DEDUCCIÓN DE LA FÓRMULA CUADRÁTICA.

La expresión analítica que resuelve cualquier ecuación cuadrática se obtiene completando el trinomio cuadrado perfecto en la igualdad **(1.2)** de la siguiente forma:

$$\begin{aligned}
 a_2x^2 + a_1x + a_0 &= a_2 \left[x^2 + \frac{a_1}{a_2}x + \frac{a_0}{a_2} \right] = 0 \\
 &= a_2 \left[\left(x^2 + \frac{a_1}{a_2}x + \left\{ \frac{a_1}{2a_2} \right\}^2 \right) + \left(\frac{a_0}{a_2} - \left\{ \frac{a_1}{2a_2} \right\}^2 \right) \right] = 0 \\
 &= a_2 \left[\left(x + \frac{a_1}{2a_2} \right)^2 + \frac{a_1^2 - 4a_2a_0}{4a_2^2} \right] = 0 \\
 &\Leftrightarrow \left(x + \frac{a_1}{2a_2} \right)^2 = \frac{a_1^2 - 4a_2a_0}{4a_2^2}
 \end{aligned}$$

Despejando x y admitiendo soluciones complejas para esta igualdad:

$$\Rightarrow x = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2a_0}}{2a_2} \quad (1.3)$$

lo que equivale a:

$$\begin{cases} x_1 = \frac{-a_1 + \sqrt{a_1^2 - 4a_2a_0}}{2a_2} \\ x_2 = \frac{-a_1 - \sqrt{a_1^2 - 4a_2a_0}}{2a_2} \end{cases}$$

que serán las raíces de la igualdad **(1.2)** en el campo complejo. La ecuación **(1.3)** se denomina universalmente la fórmula cuadrática.

La cantidad:

$$\Delta = a_1^2 - 4a_2a_0 \quad (1.4)$$

es llamada discriminante de la ecuación **(1.2)** y, determina la naturaleza de sus raíces (si éstas son reales y distintas, reales e iguales, o complejas conjugadas). Para un estudio más profundo de la expresión **(1.4)** (comprensible perfectamente para estudiantes del nivel de bachillerato), refiérase a **SAGASTUME BERRA y FERNÁNDEZ [29]**, página **408** y siguientes.

1.3.2. ECUACIÓN CÚBICA.

La ecuación:

$$a_3x^3 + a_2x^2 + a_1x + a_0 = 0 \quad (1.5)$$

se denomina cúbica, donde los $a_i \in R$ y $a_3 \neq 0$, ésta puede ser resuelta como sigue:

1.3.2.1. DEDUCCIÓN DE LA FÓRMULA CÚBICA.

La expresión analítica que nos da la solución de cualquier ecuación cúbica la deduciremos así:

En la igualdad **(1.5)**, dividiendo toda la expresión por a_3 , y empleando la traslación:

$$x = y - \frac{a_2}{3a_3}$$

que por conveniencia fue escogida para solucionar una ecuación cúbica. Los detalles para elegir esta traslación se localizan en el apéndice de teoría de ecuaciones (ver teorema número 1). Así, tenemos que:

$$\begin{aligned} \Rightarrow a_3 x^3 + a_2 x^2 + a_1 x + a_0 &= x^3 + \frac{a_2}{a_3} x^2 + \frac{a_1}{a_3} x + \frac{a_0}{a_3} \\ \Rightarrow \left(y - \frac{a_2}{3a_3} \right)^3 + \frac{a_2}{a_3} \left(y - \frac{a_2}{3a_3} \right)^2 + \frac{a_1}{a_3} \left(y - \frac{a_2}{3a_3} \right) + \frac{a_0}{a_3} &= 0 \\ \Rightarrow y^3 + \left(\frac{a_2}{a_3} - \frac{a_2}{a_3} \right) y^2 + \left(\frac{1}{3} \left[\frac{a_2}{a_3} \right]^2 - \frac{2}{3} \left[\frac{a_2}{a_3} \right]^2 + \frac{a_1}{a_3} \right) y + \\ &\quad \left(\frac{1}{9} \left[\frac{a_2}{a_3} \right]^3 - \left[\frac{a_2}{3a_3} \right]^3 - \frac{a_1 a_2}{3a_3^2} + \frac{a_0}{a_3} \right) = 0 \\ \Rightarrow y^3 + \left(\frac{a_1}{a_3} - \frac{1}{3} \left[\frac{a_2}{a_3} \right]^2 \right) y + \left(\frac{2}{27} \left[\frac{a_2}{a_3} \right]^3 - \frac{a_1 a_2}{3a_3^2} + \frac{a_0}{a_3} \right) &= 0 \end{aligned} \quad (1.6)$$

Se denomina cúbica reducida esta última.

Denotemos por $p \wedge q$:

$$\begin{cases} p = \frac{a_1}{a_3} - \frac{1}{3} \left[\frac{a_2}{a_3} \right]^2 \\ q = \frac{2}{27} \left[\frac{a_2}{a_3} \right]^3 - \frac{a_1 a_2}{3a_3^2} + \frac{a_0}{a_3} \end{cases}$$

En consecuencia, la igualdad **(1.6)** se reduce a:

$$y^3 + py + q = 0 \quad (1.7)$$

Introduzcamos dos nuevas variables en el problema, con el objeto de originar una nueva ecuación que nos permita relacionar los valores que conocemos, y reducir el orden de la ecuación (es una substitución artificiosa), de este modo, hagamos:

$$y = u + v, \quad (1.8)$$

con $u, v \in C$, como variables desconocidas.

Elevando al cubo la ecuación **(1.8)** tenemos:

$$y^3 = u^3 + v^3 + 3uv(u + v)$$

reemplazando $(u + v)$ por su valor dado en la igualdad **(1.8)**:

$$\Rightarrow y^3 = u^3 + v^3 + 3uvy$$

$$\Rightarrow y^3 - u^3 - v^3 - 3uvy = 0 \quad (1.9)$$

Al igualar los coeficientes de las expresiones **(1.7)** y **(1.9)** se obtiene el sistema de ecuaciones:

$$\begin{cases} p = -3uv \\ q = -u^3 - v^3 \end{cases} \quad (1.10)$$

que expresado en otra forma es:

$$\begin{cases} -\frac{p^3}{27} = u^3 v^3 \\ -q = u^3 + v^3 \end{cases} \quad (1.11)$$

Tomando en cuenta las relaciones que existen entre los coeficientes y las raíces de una ecuación cuadrática, tenemos que $u^3 \wedge v^3$ son las raíces de una ecuación de este tipo (ver el teorema número **2** del apéndice de teoría de ecuaciones, es decir, las fórmulas de Viète). De este modo podemos establecer que:

$$(t - u^3)(t - v^3) = 0 \quad (1.12)$$

$$\Rightarrow t^2 - (u^3 + v^3)t + u^3 v^3 = 0$$

y al comparar con el sistema de ecuaciones **(1.11)**; a la última ecuación, ésta puede expresarse como:

$$t^2 + qt - \frac{p^3}{27} = 0$$

que se denomina resolvente de la ecuación cúbica. Aplicando la fórmula cuadrática **(1.3)** a la última igualdad:

$$\Rightarrow t = \frac{-q \pm \sqrt{q^2 + \frac{4p^3}{27}}}{2}$$

$$\Rightarrow t = -\frac{q}{2} \pm \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}$$

por la condición **(1.12)**, la última ecuación se subdivide en:

$$\begin{cases} u^3 = -\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \\ v^3 = -\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \end{cases} \quad (1.13)$$

El valor de y se obtiene al reemplazar el sistema de ecuaciones **(1.13)** en **(1.8)** y está dado por:

$$y = u + v = \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}}$$

Como cada una de las cantidades del segundo miembro del sistema de ecuaciones **(1.13)** tiene tres raíces cúbicas (en el campo complejo), parece que la variable y admite nueve posibles valores; sin embargo, esto no es así en la realidad. Lo que sucede es que al pasar del sistema de ecuaciones **(1.10)** al **(1.11)** se han introducido soluciones extrañas al mismo.

Tenemos que las raíces cúbicas de $u \wedge v$, por el teorema de Moivre (ver apéndice de cálculo, teorema número **2**), al ser expresadas en su forma cartesiana, están dadas por: $u, \omega u, \omega^2 u; \wedge v, \omega v, \omega^2 v$; respectivamente, donde $\omega, \omega^2 \wedge \omega^3 = 1$ son las raíces cúbicas complejas de la unidad, expresadas por:

$$\begin{cases} \omega = \frac{-1 + i\sqrt{3}}{2} \\ \omega^2 = \frac{-1 - i\sqrt{3}}{2} \\ \omega^3 = \omega\omega^2 = 1 \end{cases}$$

Para suprimir las soluciones extrañas, probaremos los posibles pares de valores que surgen en **(1.13)** de modo que encontremos las parejas de $u \wedge v$ que satisfagan la condición **(1.10)**, es decir:

$$\begin{aligned} 3uv &= -p; p \in R \\ \Rightarrow uv &= -\frac{p}{3}; \end{aligned}$$

luego

$$(\omega u)(\omega^2 v) = \omega^3 uv = -\frac{P}{3}$$

$$(\omega^2 u)(\omega v) = \omega^3 uv = -\frac{P}{3}$$

$$u(\omega v) = \omega uv = -\omega \frac{P}{3} \neq -\frac{P}{3}$$

$$u(\omega^2 v) = \omega^2 uv = -\omega^2 \frac{P}{3} \neq -\frac{P}{3}$$

$$(\omega u)v = \omega uv = -\omega \frac{P}{3} \neq -\frac{P}{3}$$

$$(\omega u)(\omega v) = \omega^2 uv = -\omega^2 \frac{P}{3} \neq -\frac{P}{3}$$

$$(\omega^2 u)v = \omega^2 uv = -\omega^2 \frac{P}{3} \neq -\frac{P}{3}$$

$$(\omega^2 u)(\omega^2 v) = \omega^4 uv = \omega uv = -\omega \frac{P}{3} \neq -\frac{P}{3}$$

Si $u \wedge v$ son los valores de cualquier par de raíces cúbicas que satisfacen esta condición, entonces los otros dos pares están dados por: $\omega u, \omega^2 v; \wedge \omega^2 u, \omega v$, pues ellos, como se acaba de probar, son los únicos que verifican la condición requerida.

En conclusión, las raíces cúbicas de la ecuación **(1.7)** son:

$$\begin{cases} y_1 = u + v \\ y_2 = \omega u + \omega^2 v \\ y_3 = \omega^2 u + \omega v \end{cases}$$

Como en la ecuación **(1.5)** hicimos:

$$x = y - \frac{a_2}{3a_3}$$

entonces, las soluciones de esta ecuación estarán expresadas por:

$$\begin{cases} x_1 = u + v - \frac{a_2}{3a_3} \\ x_2 = \omega u + \omega^2 v - \frac{a_2}{3a_3} \\ x_3 = \omega^2 u + \omega v - \frac{a_2}{3a_3} \end{cases} \quad (1.14)$$

1.3.2.2. DISCUSIÓN DE LAS FÓRMULAS DE CARDANO.

La naturaleza de las raíces de la ecuación cúbica reducida **(1.7)** depende del signo de la cantidad subradical en las expresiones **(1.13)**, o sea de:

$$\Delta = \frac{q^2}{4} + \frac{p^3}{27}$$

El estudio de la naturaleza de dichas raíces no es del conocimiento general, por tanto, lo realizaremos a continuación.

Así, se presentan tres situaciones en las raíces de la ecuación **(1.7)**, según Δ sea mayor, menor o igual a cero.

CASO 1: $\Delta > 0$.

Entonces $u^3 \wedge v^3$ son reales; representemos por $u \wedge v$ sus raíces cúbicas aritméticas, entonces las raíces de la ecuación **(1.7)** son:

$$\begin{cases} y_1 = u + v \\ y_2 = -\frac{u + v}{2} + \frac{\iota(u - v)\sqrt{3}}{2} \\ y_3 = -\frac{u + v}{2} - \frac{\iota(u - v)\sqrt{3}}{2} \end{cases}$$

CASO 2: $\Delta = 0$.

Entonces $u^3 = v^3$; en consecuencia:

$$u = v = -\frac{q}{2}$$

luego las raíces de la ecuación **(1.7)** son:

$$\begin{cases} y_1 = u + v = u + u = 2u \\ y_2 = u(\omega + \omega^2) = -u \\ y_3 = u(\omega + \omega^2) = -u \end{cases}$$

Sólo si $p = q = 0$, la ecuación **(1.7)** tiene una raíz triple $y_1 = y_2 = y_3 = 0$.

CASO 3: $\Delta < 0$.

Entonces las expresiones $u^3 \wedge v^3$ son imaginarias:

$$\begin{cases} u^3 = -\frac{q}{2} - \iota\sqrt{-\Delta} \\ v^3 = -\frac{q}{2} + \iota\sqrt{-\Delta} \end{cases} \quad (1.15)$$

Sea $u = \alpha + \iota\beta$ cualquiera de las raíces cúbicas de u ; $\Rightarrow v = \alpha - \iota\beta$ (v es la conjugada de u). Luego, las tres raíces que resultan son reales y distintas:

$$\begin{cases} y_1 = (\alpha + \iota\beta) + (\alpha - \iota\beta) = 2\alpha \\ y_2 = \omega(\alpha + \iota\beta) + \omega^2(\alpha - \iota\beta) = -\alpha - \beta\sqrt{3} \\ y_3 = \omega^2(\alpha + \iota\beta) + \omega(\alpha - \iota\beta) = -\alpha + \beta\sqrt{3} \end{cases}$$

Este caso es denominado irreducible, dentro de las fórmulas de Cardano; porque los radicales y radicandos complejos que intervienen en ellas no pueden reemplazarse por otros reales; y, por tanto, el cálculo de estas raíces involucra el trabajo trigonométrico. Para demostrar la última afirmación; referimos al lector a **SAGASTUME BERRA y FERNÁNDEZ [29]**, página 416.

1.3.2.3. CASO IRREDUCIBLE: RESOLUCIÓN TRIGONOMÉTRICA.

Expresemos a $u^3 \wedge v^3$ en forma trigonométrica, donde $u^3 \wedge v^3$ están dadas por el sistema de ecuaciones (1.15), luego:

$$|u^3| = \sqrt{\left(-\frac{q}{2}\right)^2 - \Delta} = |v^3|$$

donde $\Delta < 0$ y:

$$\begin{cases} \arg(u^3) = \tan^{-1}\left(\frac{\frac{\sqrt{-\Delta}}{2}}{-\frac{q}{2}}\right) = \tan^{-1}\left(-\frac{2\sqrt{-\Delta}}{q}\right) \\ \arg(v^3) = \tan^{-1}\left(\frac{-\frac{\sqrt{-\Delta}}{2}}{-\frac{q}{2}}\right) = \tan^{-1}\left(\frac{2\sqrt{-\Delta}}{q}\right) \end{cases}$$

En consecuencia, podemos expresar a $u^3 \wedge v^3$ trigonómicamente, de la siguiente forma:

$$\begin{cases} u^3 = \sqrt{\left(-\frac{q}{2}\right)^2 - \Delta} \cdot \text{cis}\left(\tan^{-1}\left(-\frac{2\sqrt{-\Delta}}{q}\right)\right) \\ v^3 = \sqrt{\left(-\frac{q}{2}\right)^2 - \Delta} \cdot \text{cis}\left(\tan^{-1}\left(\frac{2\sqrt{-\Delta}}{q}\right)\right) \end{cases}$$

Por el teorema de Moivre (ver apéndice de cálculo, teorema número **2**), tenemos que las raíces cúbicas de $u \wedge v$ son:

$$\begin{cases} u = \sqrt[6]{\left(-\frac{q}{2}\right)^2 - \Delta} \cdot \text{cis}\left(\frac{1}{3}\left[\tan^{-1}\left(-\frac{2\sqrt{-\Delta}}{q}\right) + 2\pi k\right]\right) \\ v = \sqrt[6]{\left(-\frac{q}{2}\right)^2 - \Delta} \cdot \text{cis}\left(\frac{1}{3}\left[\tan^{-1}\left(\frac{2\sqrt{-\Delta}}{q}\right) + 2\pi k\right]\right) \end{cases}$$

para $k = 0, 1, 2$.

Tomando $u \wedge v$ de modo que satisfagan la condición **(1.10)**, se obtienen los valores de y que satisfacen la ecuación **(1.7)**, a través de los que obtendremos las soluciones de la ecuación cúbica **(1.5)**.

1.3.3. ECUACIÓN CUÁRTICA.

La ecuación:

$$a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = 0 \quad (1.16)$$

se denomina cuártica, donde los $a_i \in R$ y $a_4 \neq 0$, ésta puede ser resuelta como sigue:

1.3.3.1. DEDUCCIÓN DE LA FÓRMULA CUÁRTICA.

La expresión analítica que nos da la solución de la ecuación cuártica **(1.16)**, la deduciremos como sigue:

Dividiendo toda la ecuación por a_4 se tiene que:

$$x^4 + \frac{a_3}{a_4}x^3 + \frac{a_2}{a_4}x^2 + \frac{a_1}{a_4}x + \frac{a_0}{a_4} = 0 \quad (1.17)$$

Haciendo:

$$b = \frac{a_3}{a_4}, c = \frac{a_2}{a_4}, d = \frac{a_1}{a_4}, e = \frac{a_0}{a_4}$$

obtenemos:

$$x^4 + bx^3 + cx^2 + dx + e = 0$$

trasladando los tres últimos términos al miembro derecho:

$$x^4 + bx^3 = -cx^2 - dx - e$$

Agregando el término $\frac{1}{4}b^2x^2$ en ambos miembros, y completando el

cuadrado perfecto en miembro izquierdo:

$$\left[x^2 + \frac{1}{4}bx \right]^2 = \left[\frac{1}{4}b^2 - x \right] x^2 - dx - e$$

para obtener un trinomio cuadrado perfecto en el miembro izquierdo, sumamos a cada miembro:

$$\left[x^2 + \frac{1}{2}bx \right] y + \frac{1}{4}y^2$$

se tiene que:

$$\left[x^2 + \frac{1}{2}bx \right]^2 + \left[x^2 + \frac{1}{2}bx \right]y + \frac{1}{4}y^2 = \left[\frac{1}{4}b^2 - c \right]x^2 + \frac{1}{4}y^2 - dx - e + \left[x^2 + \frac{1}{2}bx \right]y$$

de donde:

$$\left(\left[x^2 + \frac{1}{2}bx \right] + \frac{1}{2}y \right)^2 = \left[\frac{1}{4}b^2 - c + y \right]x^2 + \left[\frac{1}{2}by - d \right]x - e + \frac{1}{4}y^2 \quad (1.18)$$

El miembro izquierdo de la ecuación **(1.18)** es un cuadrado perfecto, y, en consecuencia, el de la derecha también debe serlo. Luego, el discriminante de la ecuación cuadrática del miembro derecho es cero; es decir:

$$\Delta = \left[\frac{1}{2}by - d \right]^2 - 4 \left[\frac{1}{4}b^2 - c + y \right] \left[\frac{1}{4}y^2 - e \right] = 0$$

Desarrollando y simplificando términos semejantes, tenemos:

$$y^3 - cy^2 + (db - 4e)y - b^2e + 4ce - d^2 = 0 \quad (1.19)$$

denominada resolvente de la ecuación de cuarto grado.

Como ya tratamos la resolución de la ecuación cúbica, podemos encontrar una raíz de **(1.19)**. Sea $y = y_0$ una raíz encontrada a dicha ecuación, su representación está dada por:

$$y_0 = u + v + \frac{c}{3} \quad (1.20)$$

Lo antes mencionado se justifica por el sistema de ecuaciones **(1.14)**, que expresa las raíces de una ecuación cúbica cualquiera. Los valores $u \wedge v$ en y_0 se obtienen al aplicar el sistema de ecuaciones **(1.13)** a la ecuación **(1.19)**, donde:

$$\begin{cases} p = db - 4e - \frac{1}{3}c^2 \\ q = \frac{c}{3} \left(8e + bd - \frac{2c^2}{9} \right) - d^2 - b^2e \end{cases} \quad (1.21)$$

Al reemplazar en la ecuación **(1.18)** el valor de y_0 , el miembro de la derecha se convierte en el cuadrado perfecto de una función lineal en x ; representemos a esta función como $(mx + n)$. Esto es, **(1.18)** puede expresarse por:

$$\left(\left[x^2 + \frac{1}{2}bx \right] + \frac{1}{2}y_0 \right)^2 = (mx + n)^2 \quad (1.22)$$

de donde:

$$\begin{cases} \left[x^2 + \frac{1}{2}bx \right] + \frac{1}{2}y_0 = mx + n \\ \left[x^2 + \frac{1}{2}bx \right] + \frac{1}{2}y_0 = -mx - n \end{cases}$$

de modo que sólo bastaría resolver estas dos ecuaciones de segundo grado para hallar las cuatro raíces de la ecuación **(1.16)**.

En un apartado posterior de este capítulo, demostraremos que los valores de $m \wedge n$ que satisfacen la ecuación **(1.22)** están dados por:

$$\begin{cases} m = \sqrt{\frac{1}{4}b^2 - c + y_0} \\ n = \sqrt{\frac{1}{4}y_0^2 - e} \end{cases}$$

donde y_0 queda determinado por las ecuaciones **(1.20)**, **(1.13)** y **(1.21)**.

$$\text{En } \left[x^2 + \frac{1}{2}bx \right] + \frac{1}{2}y_0 = mx + n; 1 \text{ se tiene:}$$

$$\Rightarrow x^2 + \left(\frac{1}{2}b - m \right)x + \frac{1}{2}y_0 - n = 0 \quad (1.23)$$

$$\text{En } \left[x^2 + \frac{1}{2}bx \right] + \frac{1}{2}y_0 = -mx - n; 2 \text{ se tiene:}$$

$$\Rightarrow x^2 + \left(\frac{1}{2}b + m \right)x + \frac{1}{2}y_0 + n = 0 \quad (1.24)$$

Ahora, usando la fórmula cuadrática **(1.3)** para resolver las ecuaciones **(1.23)** y **(1.24)**, encontramos que:

$$\begin{cases} x_{1,2} = -\frac{\left(\frac{1}{2}b-m\right) \pm \sqrt{\left(\frac{1}{2}b-m\right)^2 - 4\left(\frac{1}{2}y_0-n\right)}}{2} \\ x_{3,4} = -\frac{\left(\frac{1}{2}b+m\right) \pm \sqrt{\left(\frac{1}{2}b+m\right)^2 - 4\left(\frac{1}{2}y_0+n\right)}}{2} \end{cases}$$

que son las cuatro raíces de la ecuación **(1.16)**.

1.3.3.2. ACLARACIÓN EN LA RESOLUCIÓN DE LA ECUACIÓN CUÁRTICA.

En la resolución de la ecuación cuártica, fue necesario encontrar una raíz de la ecuación cúbica **(1.19)** para reemplazarla en la ecuación **(1.18)**. Obtuvimos que el miembro derecho de **(1.18)** se convierte en un cuadrado perfecto. Para percatarnos de que esto realmente sucede, tomemos la raíz reemplazada **(1.20)**:

$$y_0 = u + v + \frac{c}{3} \quad (1.20)$$

donde $u \wedge v$ se pueden definir despejándolas en el sistema de ecuaciones **(1.13)**; con $p \wedge q$ dados por el sistema **(1.21)**. Además, la resolución de la ecuación cúbica impone como condición extra el sistema **(1.10)**. Considerando lo antes expuesto como hipótesis, tomemos la ecuación **(1.18)**, es decir:

$$\left(\left[x^2 + \frac{1}{2}bx \right] + \frac{1}{2}y \right)^2 = \left[\frac{1}{4}b^2 - c + y \right] x^2 + \left[\frac{1}{2}by - d \right] x - e + \frac{1}{4}y^2 \quad (1.18)$$

Como el miembro de la izquierda es un cuadrado perfecto, en consecuencia, el de la derecha también debe serlo; y podrá representarse en la forma:

$$(mx + n)^2 = m^2 x^2 + 2mx + n^2$$

donde:

$$\begin{cases} m = \sqrt{\frac{1}{4}b^2 - c + y} \\ n = \sqrt{\frac{1}{4}y_0^2 - e} \end{cases} \quad (1.25)$$

Observemos que en **(1.18)**:

$$2mn = \left(\frac{1}{2}by - d \right) \quad (1.26)$$

$$\Leftrightarrow 4m^2 n^2 = \left(\frac{1}{2}by - d \right)^2 \quad (1.27)$$

Si al reemplazar el sistema de ecuaciones **(1.25)** en **(1.27)**, obtenemos una igualdad, tomando en cuenta las hipótesis antes establecidas, probaremos que en realidad el miembro derecho de la ecuación **(1.18)** es un cuadrado perfecto.

Luego, realizando lo antes dicho, tenemos:

$$(2mn)^2 = 4\left(\frac{1}{4}b^2 - c + y\right)\left(\frac{1}{4}y^2 - e\right)$$

Realizando los productos indicados, y reemplazando el valor de y dado por la ecuación **(1.20)**:

$$\begin{aligned}\Rightarrow (2mn)^2 &= \frac{1}{4}b^2\left(u + v + \frac{c}{3}\right)^2 - c\left(u + v + \frac{c}{3}\right)^2 + \\ &\quad \left(u + v + \frac{c}{3}\right)^3 - 4e\left(u + v + \frac{c}{3}\right) - b^2e + 4ce\end{aligned}$$

Tratando de obtener un trinomio cuadrado perfecto, no alteraremos el primer término del miembro derecho de esta última expresión; mientras que factorizamos una parte adecuada de la expresión restante, resultando:

$$\begin{aligned}\Rightarrow (2mn)^2 &= \frac{1}{4}b^2\left(u + v + \frac{c}{3}\right)^2 + 4ce - b^2e + \\ &\quad -\left(u + v + \frac{c}{3}\right)\left(c\left[u + v + \frac{c}{3}\right] - \left[u + v + \frac{c}{3}\right]^2 + 4e\right) \\ \Rightarrow (2mn)^2 &= \frac{1}{4}b^2\left(u + v + \frac{c}{3}\right)^2 + 4ce - b^2e + \\ &\quad -\left(u + v + \frac{c}{3}\right)\left(\frac{c}{3}\{u + v\} - \{u + v\}^2 + \frac{2c^2}{9} + 4e\right)\end{aligned}\tag{1.28}$$

En lo sucesivo y hasta que así lo indiquemos, las operaciones siguientes las realizaremos, por conveniencia, partiendo de **(1.28)**, donde:

$$\mu = \frac{c}{3}\{u+v\} - \{u+v\}^2 + \frac{2c^2}{9} + 4e$$

Con el objeto de emplear el sistema de ecuaciones **(1.10)**, realizamos algunas de las operaciones indicadas, para expresar a μ como:

$$\mu = \frac{c}{3}\{u+v\} - \{u^2 + v^2\} - 3uv + uv + \frac{2c^2}{9} + 4e$$

Aplicando a μ , los sistemas de ecuaciones **(1.10)** y **(1.21)**; resulta al simplificar que:

$$\mu = \frac{c}{3}\{u+v\} - \{u^2 + v^2\} + db + uv - \frac{c^2}{9} \quad (1.29)$$

Reemplazando la ecuación **(1.29)**; en **(1.28)** podemos expresar esta última como:

$$\begin{aligned} \Rightarrow (2mn)^2 &= \frac{1}{4}b^2 \left(u + v + \frac{c}{3}\right)^2 + 4ce - b^2e + \\ &\quad - \left(u + v + \frac{c}{3}\right) \left(\frac{c}{3}\{u+v\} - \{u^2 + v^2\} + db + uv - \frac{c^2}{9}\right) \end{aligned}$$

Ahora, asociando y distribuyendo los términos convenientemente, la última ecuación se puede expresar como:

$$\begin{aligned} \Rightarrow (2mn)^2 &= \frac{1}{4}b^2\left(u+v+\frac{c}{3}\right)^2 - b\left(u+v+\frac{c}{3}\right)d + \\ &\quad \left(u+v+\frac{c}{3}\right)\left(u^2+v^2-uv-\frac{c}{3}(u+v)+\frac{c^2}{9}\right) + \quad (1.30) \\ &\quad 4ce - b^2e \end{aligned}$$

Observemos que los dos primeros términos del miembro de la derecha son, precisamente, parte del trinomio cuadrado perfecto que buscamos. De modo que la demostración quedará realizada si probamos que el resto de la ecuación es igual a d^2 , que es el término restante en el trinomio cuadrado perfecto requerido. De este modo, en lo sucesivo y hasta que se indique, seguiremos trabajando con la expresión:

$$\tau = \left(u+v+\frac{c}{3}\right)\left(u^2+v^2-uv-\frac{c}{3}(u+v)+\frac{c^2}{9}\right) + 4ce - b^2e \quad (1.31)$$

Asociando y distribuyendo términos, podemos expresar a τ como:

$$\begin{aligned} \tau &= 4ce - b^2e - (u+v)uv + (u+v)(u^2+v^2) + \frac{c^3}{27} + \\ &\quad \frac{c}{3}(u^2+v^2) - \frac{c}{3}(u+v)^2 - \frac{c}{3}uv \end{aligned}$$

Realizando algunas operaciones indicadas; y agrupando y reduciendo términos semejantes, τ se puede expresar como:

$$\tau = 4ce - b^2e - uvc + (u^3 + v^3) + \frac{c^3}{27}$$

Ahora, por el sistema de ecuaciones **(1.10)**, podemos decir que:

$$\tau = 4ce - b^2e + \frac{pc}{3} - q + \frac{c^3}{27}$$

Por otro lado, el sistema de ecuaciones **(1.21)** nos permite, a su vez, reemplazar los valores de $p \wedge q$. De este modo, realizando las operaciones indicadas y simplificando, τ se expresa como:

$$\tau = d^2 \quad (1.32)$$

Entonces, por las ecuaciones **(1.30)**, **(1.31)** y **(1.32)**:

$$\begin{aligned} \Rightarrow (2mn)^2 &= \frac{1}{4}b^2 \left(u + v + \frac{c}{3}\right)^2 - b \left(u + v + \frac{c}{3}\right)d + d^2 \\ \Rightarrow (2mn)^2 &= \left[\frac{1}{2}b \left(u + v + \frac{c}{3}\right) - d \right]^2 \end{aligned}$$

Por la ecuación **(1.20)**, finalmente, podemos decir:

$$\therefore 2mn = \frac{1}{2}by - d$$

finalizando la demostración.

1.4. IMPOSIBILIDAD DE LA RESOLUCIÓN DE UNA ECUACIÓN DE GRADO MAYOR QUE CUATRO, POR RADICALES O FÓRMULAS.

Hasta este punto, se ha logrado desarrollar la resolución de ecuaciones polinomiales, hasta el grado cuarto, por medio de radicales o fórmulas.

Aparentemente, la utilización de radicales de grado superior a los empleados anteriormente, y la generalización de los métodos que nos permiten obtener la solución de las ecuaciones anteriores; parecen ser la herramienta “natural” para resolver ecuaciones de cualquier grado.

Sin embargo, debemos observar que: el cálculo real de las raíces de ecuaciones de grado tercero y cuarto es sumamente complicado, e involucra la consideración de casos particulares, cada uno de los cuales requiere un análisis específico; de modo que estas fórmulas poseen sólo valor teórico. Es de esperarse que las fórmulas que permitan la solución de ecuaciones de grado superior al cuarto (asumiendo que éstas existen), sean más complicadas que las desarrolladas anteriormente, y, en consecuencia, de menor utilidad práctica.

Además, matemáticos como Ruffini, Abel y Galois probaron que las ecuaciones polinomiales de grado mayor o igual al quinto no son resolubles mediante fórmulas, en general.

La demostración de estos resultados se realiza dentro de lo que se denomina teoría de Galois. Una orientación para realizar estas pruebas, puede encontrarse en **REINHARDT y SOEDER [28]**, página **109**.

Bajo estos criterios, no tiene sentido buscar fórmulas generales para resolver ecuaciones de grado superior al cuarto, y, así, el problema de encontrar las raíces de una ecuación de grado arbitrario se debe resolver bajo un enfoque completamente distinto al realizado hasta ahora.

**2. MÉTODOS ITERATIVOS PARA ENCONTRAR
LAS RAÍCES DE ECUACIONES ALGEBRAICAS.**

2.1. OBSERVACIONES PRELIMINARES.

De lo tratado al final del capítulo anterior, se ve la necesidad de desarrollar métodos aproximados (que pueden ser iterativos), los cuales permitan calcular las raíces, reales o complejas, de una ecuación algebraica de cualquier grado.

Este trabajo se concretará a desarrollar métodos aplicables a la obtención de raíces reales, únicamente. Para lograr este objetivo, emplearemos la siguiente metodología:

A partir del conocimiento de los coeficientes y el grado del polinomio original, únicamente, estableceremos un intervalo (desde el punto de vista real), en cuyo interior se encuentren todas las raíces de la ecuación. Luego, procederemos a separar dichas raíces, de modo que cada una se encuentre dentro de un subintervalo del intervalo original. Por último, realizaremos la aproximación de cada una de estas raíces, por separado. Esta técnica es aplicable, básicamente, a la obtención de raíces reales y será el núcleo del trabajo a desarrollar.

Por lo general, los métodos a que nos referimos no generan una respuesta exacta y directa, sino que requieren que se establezca una cota superior del error que contiene la solución. Además, deben aplicarse sobre el problema en forma repetitiva, de modo que, en cada aplicación (denominada también iteración), se mejora la respuesta obtenida, al realizar dichas repeticiones un número finito de veces. Es decir, forman parte de los denominados métodos o procedimientos iterativos.

Al considerar las técnicas iterativas, es necesario responder, al menos, a dos preguntas básicas: ¿Converge la iteración al valor verdadero de la raíz?, y si así ocurre: ¿Qué tan rápido converge? Estas preguntas las responderemos en la medida de nuestras posibilidades, para cada método.

Es necesario, al aproximar las raíces, el conocimiento de un buen valor inicial (uno para cada raíz del subintervalo), para que ocurra la convergencia del método a la solución buscada. Además, la presencia de raíces múltiples entorpece a la mayoría de estos métodos, al igual que la de raíces complejas. Sin embargo, si la ecuación posee sólo raíces simples reales, y se tiene además una buena aproximación para cada raíz, estos métodos convergen en forma bastante satisfactoria y rápida a la raíz buscada.

Dentro de la teoría de ecuaciones, existe un teorema (referimos al lector al apéndice de teoría de ecuaciones, teorema número **3**), que nos permite obtener, a partir de un polinomio $f(x)$, con raíces complejas múltiples, otro polinomio $f^*(x)$, que tendrá las mismas raíces de $f(x)$, pero todas simples.

Conjugando las condiciones necesarias y suficientes para que un valor α sea raíz de multiplicidad r de un polinomio $f(x)$ (referimos al lector nuevamente al apéndice de teoría de ecuaciones, teoremas, **4, 5, 6, y 7**), con el teorema previamente mencionado, podemos simplificar considerablemente el desarrollo de este trabajo.

Además, asumiremos a $f(x) \wedge P(x)$ como las representaciones de las ecuaciones algebraicas que aparecerán en lo sucesivo.

2.2. MÉTODOS ITERATIVOS QUE NOS PERMITEN CALCULAR LAS RAÍCES REALES DE UNA ECUACIÓN ALGEBRAICA: ACOTACIÓN SUCESIVA DE LAS RAÍCES.

Éstos son los métodos a que nos hemos referido previamente en este trabajo.

2.2.1. ACOTACIÓN DE RAÍCES.

Desarrollaremos, a continuación, algunos teoremas empleados para la acotación de las raíces reales de una ecuación del tipo:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

En ciertos desarrollos consideraremos, por hipótesis, que los coeficientes de los polinomios no son todos positivos; pues, si esto ocurre, se tiene que cero es una cota superior de las raíces del polinomio, pues $f(x)$ no poseerá raíces positivas (vea el teorema 8 del apéndice de teoría de ecuaciones).

Generalmente, nos limitaremos a demostrar teoremas que permitirán encontrar cotas superiores de las raíces positivas, ya que las cotas inferiores de las raíces negativas se encuentran aplicando dichos teoremas a la ecuación que se obtiene al substituir:

$$y = -x, \tag{2.1}$$

en la ecuación original, siendo la cota inferior, el opuesto del número hallado.

El fundamento de esta afirmación se encuentra en la teoría de transformaciones lineales del álgebra lineal, y consiste, básicamente, en lo siguiente:

Consideremos una ecuación algebraica

$$f(x) = 0 \quad (2.2)$$

Si consideramos que x es, a la vez, una función de otra variable independiente z , podemos decir que:

$$x = \Psi(z) \quad (2.3)$$

que se denomina fórmula de transformación.

Al reemplazar la ecuación **(2.3)** en **(2.2)**, o, como también se dice, efectuando una transformación sobre la función $f(x) = 0$, mediante la función $x = \Psi(z)$, se obtiene otra ecuación:

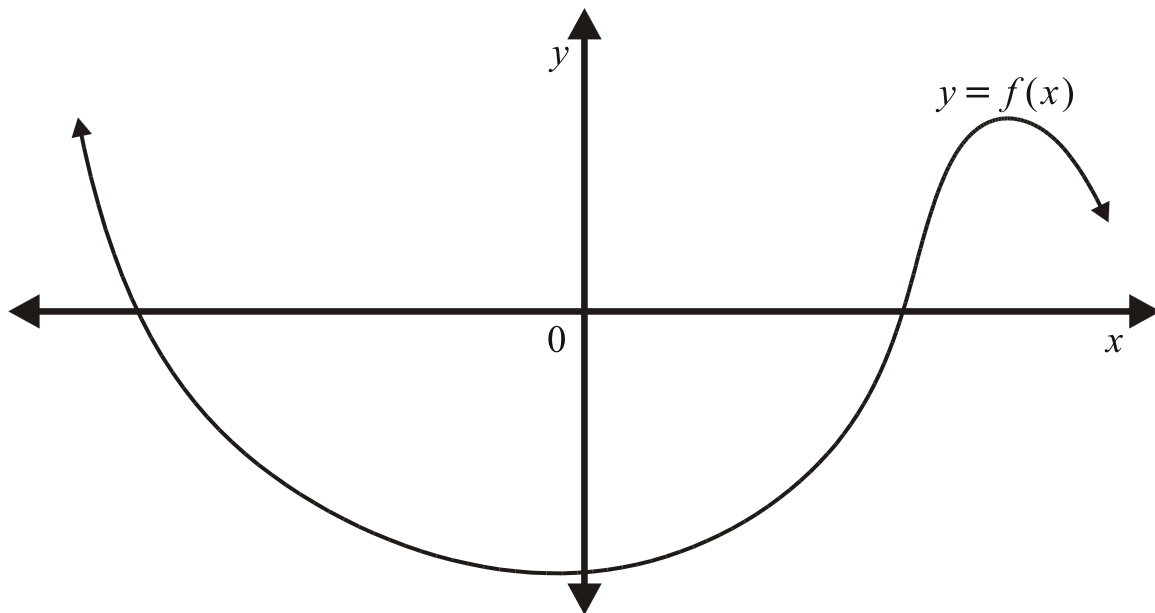
$$F(z) = f[\Psi(z)] = 0 \quad (2.4)$$

que es llamada ecuación transformada. Luego, conociendo un valor de la variable independiente en **(2.4)** o en **(2.2)**, podemos encontrar el valor que le corresponde en la otra variable independiente, empleando **(2.3)**.

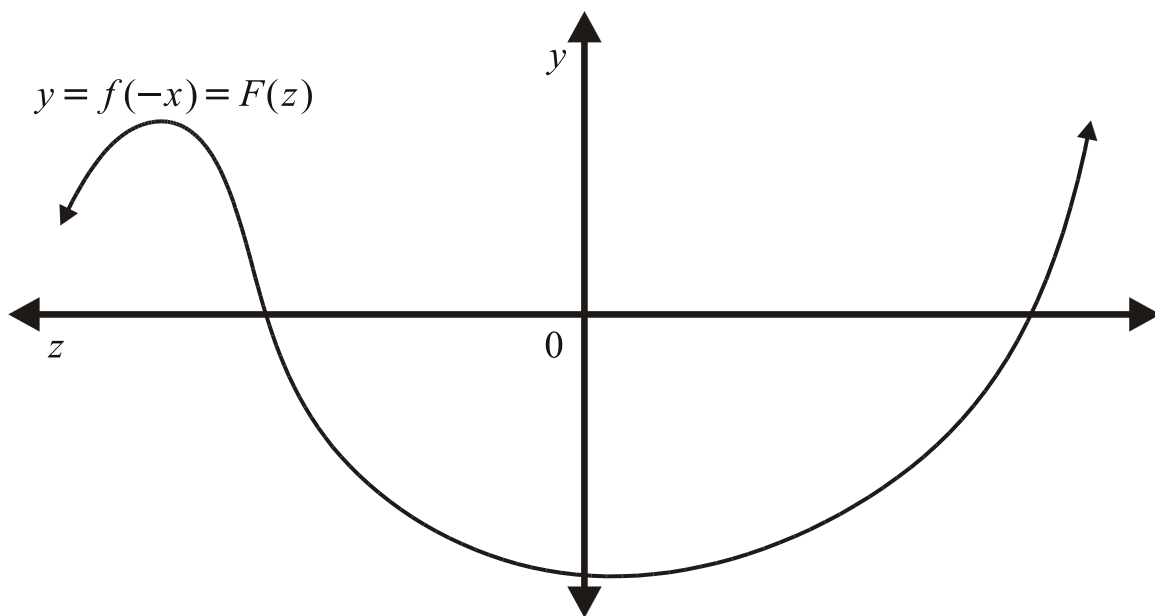
Hablando geométricamente, la transformación que usaremos, es decir, **(2.1)**, equivale a invertir el sentido del eje x , en la gráfica de la ecuación **(2.2)**, de allí que la cota inferior va a estar dada por el opuesto del número hallado al

efectuar dicha transformación. Para ilustrar lo que acabamos de mencionar, véase la Figura 1.

Figura 1



Función original $y = f(x)$



Función transformada $y = f(-x) = F(z), z = -x$

2.2.1.1. MÉTODO DE LAGUERRE-THIBAUT.

El método de Laguerre-Thibault se fundamenta en el siguiente teorema:

TEOREMA DE LAGUERRE-THIBAUT.

Si el coeficiente de x^n en la ecuación $f(x) = 0$ es positivo, y, si al dividir $f(x)$ por $(x - L)$, resultan positivos todos los coeficientes del cociente y el resto, entonces L es una cota superior de las raíces reales positivas de $f(x) = 0$.

DEMOSTRACIÓN.

Sea:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0,$$

donde $a_n > 0$. Dividiendo $f(x)$ entre $(x - L)$ se tiene que:

$$\Rightarrow f(x) = (x - L)(b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + \dots + b_1 x + b_0) + f(L)$$

Donde $f(L)$ es el resto de la división realizada y los b_i son obtenidos aplicando la regla de Ruffini (ver apéndice de teoría de ecuaciones, teorema número 9).

Hipotéticamente, el resto y los coeficientes del cociente son positivos, o sea, $b_i > 0$, $i = 0, 1, \dots, n-1$ y $f(L) > 0$, entonces $f(x) > 0, \forall x > L > 0$, por lo

tanto L es una cota superior de las raíces reales positivas de la ecuación $f(x) = 0$, lo que demuestra el teorema.

Luego, la regla de Laguerre-Thibault consistirá en probar, por el método de división sintética (ver teorema número 9 del apéndice de teoría de ecuaciones), valores crecientes de L , hasta que resulten positivos todos los coeficientes del cociente y el resto.

2.2.1.2. MÉTODO DEL MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN.

El método del mayor de los módulos de los coeficientes negativos de la ecuación se fundamenta en el siguiente teorema:

TEOREMA DEL MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN.

Sea G el mayor de los módulos de los coeficientes negativos de:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0, \quad (2.5)$$

con $a_n > 0$. Si a_k es el primer coeficiente negativo de la ecuación ($k = n-1, n-2, \dots, 2, 1, 0$), entonces:

$$L = 1 + \sqrt[n-k]{\frac{G}{a_n}}$$

es una cota superior de las raíces positivas de (2.5).

DEMOSTRACIÓN.

Tenemos que:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_{k+1} x^{k+1} + a_k x^k + a_{k-1} x^{k-1} + \dots + a_1 x + a_0 = 0$$

Haciendo $a_i = 0, i = n-1, n-2, \dots, k+1$; y reemplazando $a_{k-j} = -G$, con $j = 0, 1, 2, \dots, k$, donde:

$$G = |\min(a_0, a_1, \dots, a_n)|$$

sabiendo, por hipótesis, que $G \neq 0$. Bajo estos supuestos podemos decir que:

$$f(x) \geq a_n x^n + 0 + \dots + (-G)x^k + (-G)x^{k-1} + \dots + (-G)x + (-G)$$

donde la igualdad se da si:

$$a_{n-1} = a_{n-2} = \dots = a_0 = -G$$

$$f(x) \geq a_n x^n - G(x^k + x^{k-1} + \dots + x + 1)$$

Como:

$$x^k + x^{k-1} + \dots + x + 1 = \frac{x^{k+1} - 1}{x - 1}$$

con $x \neq 1$.

$$\Rightarrow f(x) \geq a_n x^n - G \left(\frac{x^{k+1} - 1}{x - 1} \right)$$

$$\Rightarrow f(x) \geq \frac{(x-1)a_n x^n - G(x^{k+1} - 1)}{x - 1}$$

$$\Rightarrow f(x) \geq \frac{a_n x^{n+1} - a_n x^n - Gx^{k+1} + G}{x - 1}$$

$$\Rightarrow f(x) \geq \frac{x^{k+1}(a_n x^{n-k} - a_n x^{n-k-1} - G) + G}{x - 1}$$

$$\Rightarrow f(x) \geq \frac{x^{k+1}(a_n x^{n-k-1}[x-1] - G) + G}{x - 1}$$

$$\Rightarrow f(x) > \frac{x^{k+1}(a_n x^{n-k-1}[x-1] - G)}{x - 1}$$

Dado que $x^{n-k-1} = ([x-1] + 1)^{n-k-1}$:

$$\Rightarrow f(x) > \frac{x^{k+1}(a_n ([x-1] + 1)^{n-k-1} [x-1] - G)}{x - 1}$$

Como $([x-1] + 1)^{n-k-1} > [x-1]^{n-k-1}$, si $x > 1$:

$$\Rightarrow f(x) > \frac{x^{k+1}(a_n (x-1)^{n-k-1} [x-1] - G)}{x - 1}$$

$$\Rightarrow f(x) > \frac{x^{k+1}(a_n [x-1]^{n-k} - G)}{x - 1}$$

Ya que $x > 1 \Rightarrow x - 1 > 0 \wedge x^{k+1} > 1$, luego:

$$\Rightarrow \alpha = \frac{x^{k+1}}{x-1} > 0$$

Luego:

$$f(x) > \alpha [a_n (x-1)^{n-k} - G] \quad (2.6)$$

con $x > 1 \wedge \alpha > 0$.

Tomemos el segundo factor del miembro derecho de **(2.6)**, con el objeto de encontrar los valores de x que lo anulan y de este modo garantizar que $f(x) > 0$, así:

$$\begin{aligned} a_n (x-1)^{n-k} - G &= 0 \\ \Rightarrow x &= 1 + \sqrt[n-k]{\frac{G}{a_n}} = L \end{aligned}$$

Luego, el único valor que anula al miembro derecho de **(2.6)** es L , de modo que $\forall x \geq L, f(x) > 0$, lo que indica que L es una cota superior de las raíces positivas de $f(x)$, lo que culmina la demostración.

Entonces, el método del mayor de los módulos de los coeficientes negativos de la ecuación, consistirá en aplicar directamente el teorema a la ecuación particular, para generar la cota buscada.

2.2.1.3. MÉTODO DE NEWTON.

El método de Newton se fundamenta en el siguiente teorema:

TEOREMA DE NEWTON.

Si para $x = L$, los polinomios $f(x), f'(x), f''(x), \dots, f^{(n)}(x)$ son positivos, L es una cota superior de las raíces de $f(x) = 0$.

DEMOSTRACIÓN.

La fórmula de Taylor (ver teorema número 1 del apéndice de cálculo), nos permite escribir el desarrollo de un polinomio en potencias de $(x - L)$, mediante la siguiente expresión:

$$f(x) = f([x - L] + L)$$

$$f(x) = f(L) + f'(L) \frac{(x - L)}{1!} + f''(L) \frac{(x - L)^2}{2!} + \dots + f^{(n)}(L) \frac{(x - L)^n}{n!}$$

Por hipótesis, $f(x), f'(x), f''(x), \dots, f^{(n)}(x)$ son positivos, de modo que x debe ser mayor que L , para que $f(x) > 0$, (ya que así, todas las potencias de $(x - L)$ serán positivas, de modo que en toda la expresión, sólo habrá términos positivos).

Luego, para $\forall x \geq L$, es $f(x) > 0$, y por tanto, no existen raíces positivas mayores que L , siendo L una cota superior de las raíces positivas de $f(x)$, lo que culmina la demostración.

Así, la regla de Newton consistirá en probar, por el esquema de Horner generalizado (ver teorema número 10 del apéndice de teoría de ecuaciones), valores crecientes de L en la función f y en sus n primeras derivadas, hasta que resulten positivos todos los restos.

2.2.1.4. MÉTODO ALTERNATIVO.

A continuación, enunciaremos, sin demostración, un teorema referente a la acotación de raíces reales; que nos puede ayudar considerablemente a obtener las soluciones reales de una ecuación algebraica.

TEOREMA DE LAGUERRE.

Dado un polinomio:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0 = 0$$

con n ceros reales, todos ellos quedan comprendidos en el intervalo cuyos extremos son las dos raíces de la ecuación cuadrática:

$$nx^2 + 2a_{n-1}x + (2(n-1)a_{n-2} - (n-2)^2 a_{n-1}^2) = 0$$

Luego, la aplicación directa del teorema, a la ecuación particular, generará las cotas superior e inferior de las raíces de la ecuación.

2.2.2. SEPARACIÓN DE RAÍCES.

Desarrollaremos, a continuación, algunos teoremas empleados para la separación de raíces reales en ecuaciones algebraicas. Estos teoremas, en general, requieren que dichas raíces sean acotadas previamente, por lo que consideraremos, si es necesario, que ya se a realizado dicho proceso.

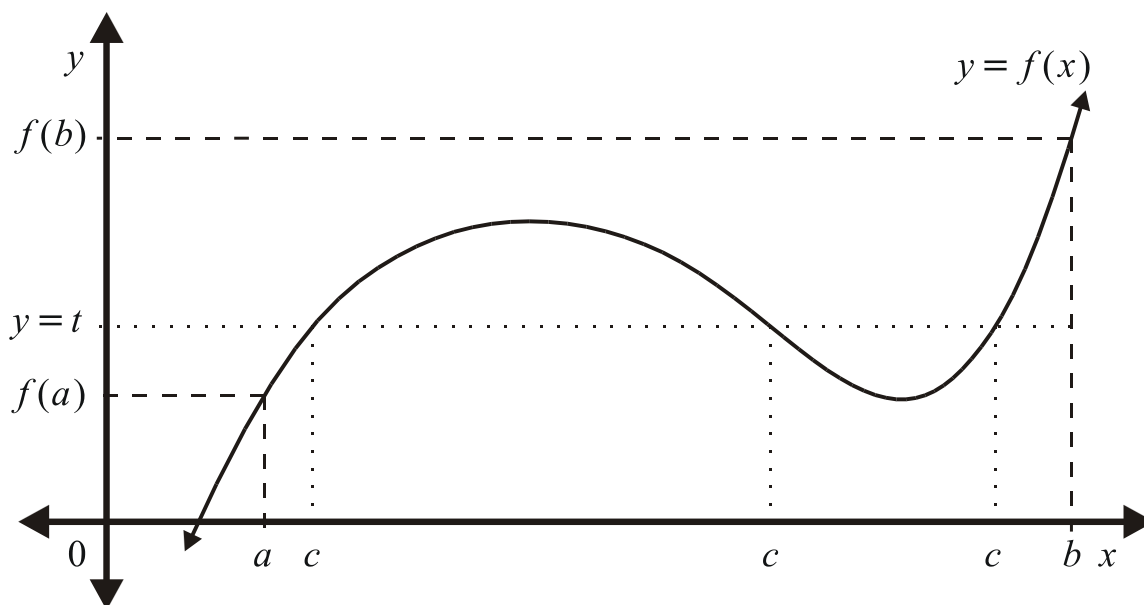
Antes de adentrarnos en la temática de estos métodos, es muy importante dejar sentado que: Estos métodos se aplican a las ecuaciones, para

tratar de generar un “valor inicial”, que ponga a funcionar los métodos que permiten aproximar raíces.

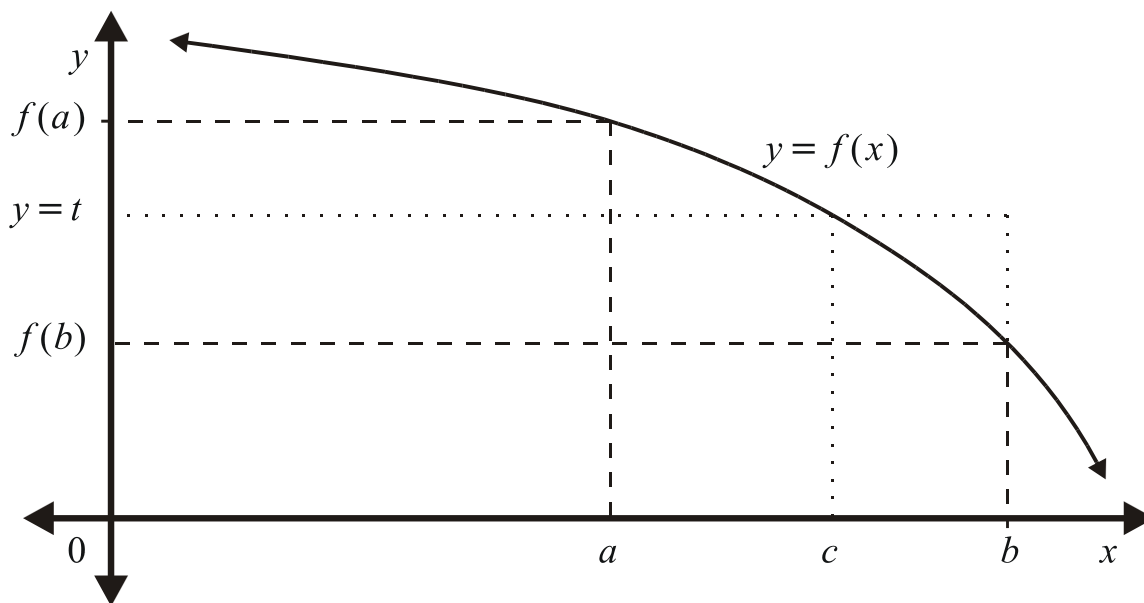
Además, puede ocurrir que: con los valores iniciales generados por uno de estos métodos, no se converja a las raíces buscadas, debido a que dichos valores no cumplen las condiciones propias de convergencia del método de aproximación. En tal caso, se debe reiterar la aplicación del método de separación al intervalo acotador, de modo que se obtengan valores iniciales más precisos, que conduzcan a la convergencia deseada, al aproximar las raíces.

Primeramente, haremos referencia al llamado “teorema del valor intermedio”, el cual enuncia que: Si f es una función continua sobre $[a,b]$ y si la gráfica de f está por debajo (respectivamente por encima) de la recta $y = t$ en a y por encima (respectivamente por debajo) de esta recta en b , entonces la gráfica de f debe intersectar dicha recta en, por lo menos, un punto c situado en (a,b) , (ver Figura 2).

Figura 2



Caso 1: $f(a) < f(b)$

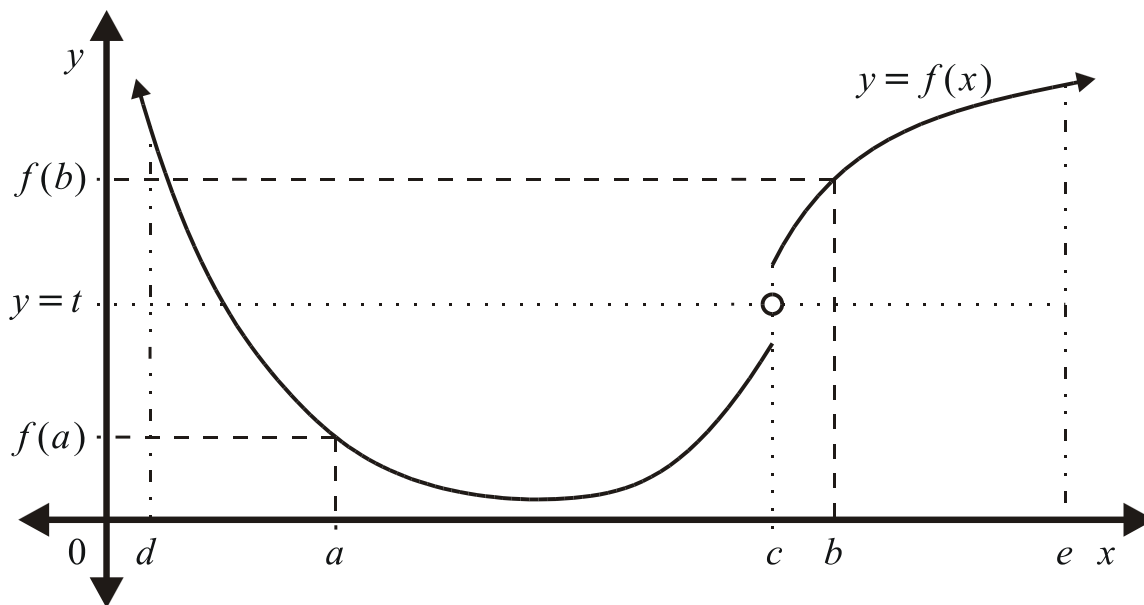


Caso 2: $f(a) > f(b)$

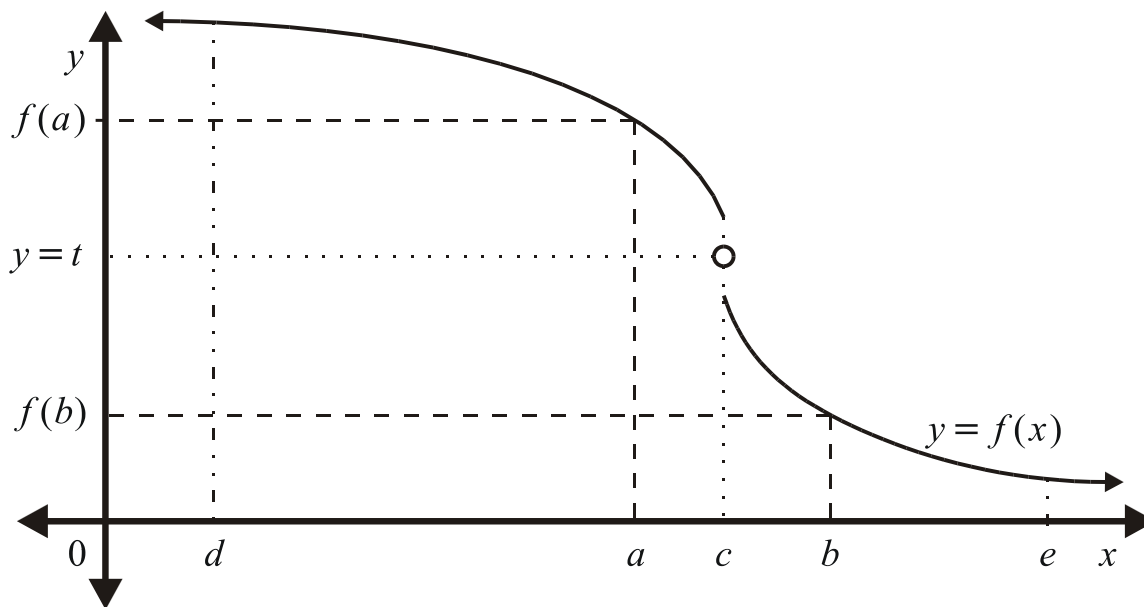
Este teorema se puede interpretar geoméricamente, como sigue: si una función f es continua sobre un intervalo cerrado $[a, b]$, entonces, la curva que representa la gráfica de f no puede tener salto alguno en ese intervalo.

Por otro lado, si una función f es discontinua sobre $[a, b]$, la curva que representa su gráfica es análoga a las curvas que se presentan en la Figura 3, pues f es discontinua en $[d, e]$ (ya que $[a, b] \subset [d, e]$), luego se implica que $\exists c \in (a, b)$ tal que $f(c) = t$.

Figura 3



Caso 1: $f(a) < f(b)$



Caso 2: $f(a) > f(b)$

2.2.2.1. TEOREMA DEL VALOR INTERMEDIO.

Si f es una función continua sobre $[a, b]$ y $f(a) \neq f(b)$ y t es un número cualquiera entre $f(a) \wedge f(b)$, entonces existe un punto $c \in (a, b)$ (no necesariamente único), tal que $f(c) = t$.

DEMOSTRACIÓN.

Como $f(a) \neq f(b)$, surgen dos posibilidades a estudiar:

CASO 1: $f(a) < t < f(b)$.

Denotemos por I al intervalo $[a, b]$. Sea S el subconjunto de I dado por:

$$S = \{x \in I : f(x) < t\} \quad (2.7)$$

Este conjunto no es vacío (por ser $f(a) < t$, luego $a \in S$) y está acotado superiormente (tenemos por hipótesis, que $f(b) > t$, luego $b \notin S$, y por tanto es una cota superior de dicho conjunto). En consecuencia, por el axioma del supremo (ver el teorema número 3 del apéndice de cálculo), S tiene un supremo, que denominaremos c , es decir, $\sup(S) = c$. Debemos probar que $f(c) = t$.

Como $a \in S$ y b es una cota superior de dicho conjunto, podemos afirmar que $c \in [a, b]$.

Al ser c el extremo superior de S , existe una sucesión:

$$\{c_k\}_{k=1}^{\infty}$$

de elementos de S , tal que:

$$\lim_{k \rightarrow \infty} c_k = c$$

(ver en el apéndice de cálculo, el teorema número 4). Luego:

$$\Rightarrow f(c) = f\left(\lim_{k \rightarrow \infty} c_k\right)$$

y por ser f continua en $c \in I$ (ver el teorema número 5 del apéndice de cálculo):

$$\Rightarrow f(c) = f\left(\lim_{k \rightarrow \infty} c_k\right) = \lim_{k \rightarrow \infty} f(c_k)$$

Como todos los elementos de la sucesión son parte de S , por (2.7):

$$\Rightarrow f(c_k) < t,$$

para todo entero positivo k , luego, para k suficientemente grande:

$$f(c) = \lim_{k \rightarrow \infty} f(c_k) \leq t$$

$$\Rightarrow f(c) \leq t \quad (2.8)$$

Definamos ahora la sucesión:

$$\{t_k\}_{k=1}^{\infty}$$

donde los $t_k \in I$ y están definidos por:

$$t_k = c + \frac{1}{k}$$

Es evidente que:

$$\lim_{k \rightarrow \infty} t_k = c$$

y todo t_k resulta ser cota superior de S . Es decir, para k suficientemente grande, $t_k \in I \wedge t_k \notin S$, por lo tanto $f(t_k) > t$, para todo entero positivo k , luego, para k suficientemente grande y como f es continua en I , se tiene:

$$f(c) = f\left(\lim_{k \rightarrow \infty} t_k\right) = \lim_{k \rightarrow \infty} f(t_k) \geq t$$

$$\Rightarrow f(c) \geq t \quad (2.9)$$

Luego de las desigualdades (2.8) y (2.9) $\Rightarrow f(c) = t$.

Además, de las mismas desigualdades (2.8) y (2.9) tenemos que los casos $f(c) > t \wedge f(c) < t$ son excluidos, de la solución, de modo que c no puede ser ni a ni b , lo que implica que $c \in (a, b)$, lo que finaliza la demostración del Caso 1.

CASO 2: $f(a) > t > f(b)$.

Definamos la función $g(x) = -f(x)$.

Dado que f es continua en I , $\Rightarrow g$ lo será (ver apéndice de cálculo, teorema número 6).

También sabemos que:

$$\begin{cases} g(a) = -f(a) \\ g(b) = -f(b) \end{cases}$$

Como $f(a) > t > f(b)$

$$\Rightarrow -f(a) < -t < -f(b)$$

$$\Rightarrow g(a) < -t < g(b)$$

luego, por el Caso 1, $\exists c \in R : a < c < b$ y:

$$-t = g(c) = -f(c)$$

de donde:

$$f(c) = t$$

Finalizando la demostración de ambos casos.

De este teorema podemos establecer en forma inmediata el siguiente corolario, de gran importancia en la matemática, conocido como teorema de bisección de Bolzano, que dice así:

2.2.2.1.1. TEOREMA DE BISECCIÓN DE BOLZANO.

Si f es una función continua sobre $[a, b]$ y $f(a) \neq f(b)$ y 0 es un número entre $f(a) \wedge f(b)$, entonces existe un punto $c \in (a, b)$ (no necesariamente único), tal que $f(c) = 0$.

DEMOSTRACIÓN.

Es evidente que: al hacer $t = 0$ en el teorema del valor intermedio, se obtiene el resultado mencionado.

Debe observarse que: la demostración del teorema del valor intermedio, sólo deja entrever levemente el procedimiento que debe emplearse (en la práctica) para encontrar la raíz mencionada, aunque es de tipo constructivo (pero existencial). Por esta razón, proporcionaremos el procedimiento que, en sí, se puede emplear para establecer dicha raíz.

2.2.2.1.2. PROCEDIMIENTO DE BISECCIÓN.

Consideremos que una función f cambia de signos en los extremos de un intervalo $[a, b]$ donde es continua (nos referimos a que $f(a) > 0 \wedge f(b) < 0$ ó $f(a) < 0 \wedge f(b) > 0$, lo que se puede expresar en forma más compacta, diciendo que $f(a)f(b) < 0$). Además, necesitamos establecer un error en la

aproximación que hacemos de la raíz —denominado tolerancia—, y lo denotaremos ε .

Dividimos el intervalo $[a, b]$ en dos partes iguales y evaluemos el polinomio en el punto central del mismo, es decir, en:

$$p = \frac{a+b}{2}$$

En caso de que el polinomio se anule en dicho punto, obtendremos la raíz que buscamos, y si no, habrá uno sólo de los subintervalos parciales de $[a, b]$ (lo denominaremos como $[a_1, b_1] \subset [a, b]$) en el que f cambia de signo en sus extremos.

Este subintervalo lo construiremos así: Si $f(a)f(p) < 0$, entonces $[a_1, b_1] = [a, p]$, y si $f(a)f(p) > 0$, entonces haremos $[a_1, b_1] = [p, b]$. El caso $f(a)f(p) = 0$, como ya lo hemos dicho, implica que p es la raíz de f .

Repitiendo este procedimiento, obtendremos una sucesión de subintervalos encajados:

$$[a, b] \supset [a_1, b_1] \supset [a_2, b_2] \supset \dots$$

en que para algún $k \in \mathbb{N}$, $|b_k - a_k| < \varepsilon$, es decir, obtendremos que la longitud del subintervalo $[a_k, b_k]$ sea menor que nuestra tolerancia, de modo que podemos tomar el punto:

$$x = \frac{a_k + b_k}{2}$$

como aproximación de la raíz que buscamos.

Los interesados en la demostración rigurosa del teorema de bisección de Bolzano, empleando esta técnica, pueden dirigirse a **SAGASTUME BERRA y FERNÁNDEZ [29]**, página **387** y siguientes para los detalles respectivos.

Por otro lado resulta extraño, a primera vista, que hagamos referencia (al introducirnos al tema de la separación de raíces reales), a un teorema que nos garantiza la existencia de, al menos, una raíz en un intervalo particular, y, prácticamente, sugiere una técnica iterativa para aproximar raíces reales, cuando aún no se ha tratado dicho problema.

Sin embargo, esto se debe a que dicho teorema nos permitirá, con el auxilio de la teoría de ecuaciones clásica, desarrollar un método para separar raíces reales de ecuaciones algebraicas, conocido como el método de Rolle, que detallaremos en el siguiente punto.

Por otro lado, cuando tratemos el tema de la aproximación de raíces reales, volveremos a referirnos al teorema de bisección de Bolzano más a fondo, tratándolo como un método particular para aproximar raíces reales.

2.2.2.2. MÉTODO DE ROLLE.

El método de Rolle no presenta una regla práctica que resuelva el problema de la separación de las raíces reales de una ecuación algebraica, que sea aplicable en todos los casos; sin embargo, ofrece un análisis muy profundo

del problema en mención (al igual que el método de bisección de Bolzano). Por esta razón, lo expondremos.

Dicho método consiste en relacionar las raíces de la ecuación $f(x) = 0$, con las de su derivada, de modo que, conociendo las raíces de f' , podemos separar las de f , empleando, básicamente, dichas raíces como puntos de separación de los intervalos que contienen a las raíces de f .

El método de Rolle se fundamenta en el teorema que enunciamos a continuación.

TEOREMA DE ROLLE.

Entre dos raíces reales consecutivas de un polinomio real

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0,$$

hay un número impar de raíces de su derivada $f'(x) = 0$, contando r veces cada raíz de orden r .

DEMOSTRACIÓN.

Podemos distinguir claramente dos situaciones en el planteamiento del problema.

CASO 1: LAS RAÍCES SON IGUALES.

Si a es una raíz de multiplicidad dos, de las raíces de $f(x)$, éstas se consideran (dentro del teorema) como dos raíces cualesquiera, y el único punto

situado entre $[a, a]$ es el mismo a , y, por el teorema que expresa las condiciones necesarias y suficientes para que un valor a sea raíz de multiplicidad r -eple en la ecuación $f(x)=0$ (ver el apéndice de teoría de ecuaciones, teorema número 4), podemos decir $f'(a)=0$ y por tanto a es raíz de $f'(x)=0$, cumpliéndose así que entre $a \wedge a$ existe un número impar de raíces de $f(x)$.

CASO 2: LAS RAÍCES SON DISTINTAS.

Sean $a_1 \wedge a_2$ dos raíces reales consecutivas de $f(x)=0$. Consideramos que $a_1 < a_2$.

Tomemos las razones:

$$\frac{f(a_1 + \varepsilon)}{f'(a_1 + \varepsilon)} \quad (2.10)$$

$$\frac{f(a_2 - \varepsilon)}{f'(a_2 - \varepsilon)} \quad (2.11)$$

Para $\varepsilon > 0$ y suficientemente pequeño, tenemos que (2.10) es positiva y (2.11) es negativa, pues por el teorema número 11 del apéndice de teoría de ecuaciones, podemos afirmar que:

$$\begin{cases} f(a_1 + \varepsilon)f'(a_1 + \varepsilon) > 0 \\ f(a_2 - \varepsilon)f'(a_2 - \varepsilon) < 0 \end{cases} \quad (2.12)$$

Por hipótesis, podemos tomar ε lo suficientemente pequeño ($\varepsilon < |a_2 - a_1|$), tal que entre $(a_1 + \varepsilon) \wedge (a_2 - \varepsilon)$ no existan raíces de $f(x) = 0$. Luego, por el teorema número **12** del apéndice de teoría de ecuaciones:

$$\Rightarrow f(a_1 + \varepsilon) \wedge f(a_2 - \varepsilon) \quad (2.13)$$

tienen igual signo.

Entonces por **(2.12)** y **(2.13)** se deduce que:

$$f'(a_1 + \varepsilon) \wedge f'(a_2 - \varepsilon)$$

tienen signos opuestos, lo que indica, según el teorema número **13** del apéndice de teoría de ecuaciones, que entre $(a_1 + \varepsilon) \wedge (a_2 - \varepsilon)$ existe un número impar de raíces de $f'(x) = 0$. Como ε es suficientemente pequeño, entre los extremos de cada uno de los siguientes intervalos:

$$(a_1, a_1 + \varepsilon], [a_2 - \varepsilon, a_2)$$

no hay raíces de $f'(x)$, de modo que, finalmente, entre $a_1 \wedge a_2$ existe un número impar de raíces de $f'(x) = 0$, lo que demuestra el teorema.

COROLARIO.

Entre dos raíces consecutivas $\beta_1 \wedge \beta_2$ de $f'(x)$ existe, a lo sumo, una raíz de $f(x)$.

DEMOSTRACIÓN.

Supongamos —por reducción al absurdo—, que entre $\beta_1 \wedge \beta_2$ existen dos raíces consecutivas $\alpha_1 \wedge \alpha_2$ de $f(x)$. En consecuencia, podemos construir la siguiente desigualdad:

$$\beta_1 < \alpha_1 < \alpha_2 < \beta_2$$

El teorema anterior garantiza que entre $\alpha_1 \wedge \alpha_2$ existe un número impar de raíces de $f'(x)$ —al menos una, que llamaremos β_1 —. Sin embargo, esta afirmación contradice la hipótesis de que $\beta_1 \wedge \beta_2$ son dos raíces consecutivas de $f'(x)$, ya que se formaría la desigualdad:

$$\beta_1 < \alpha_1 < \beta_1 < \alpha_2 < \beta_2$$

que resulta absurda, finalizando así la demostración.

Luego, la aplicación del teorema en mención, debe ser realizada siguiendo el siguiente patrón:

En primer término, se deben establecer las cotas superior e inferior de las raíces de la ecuación $f(x) = 0$. Posteriormente, se calcula $f'(x)$.

A continuación, se aplica algún método sencillo para calcular las raíces de $f'(x)$ —tal como el procedimiento de bisección de Bolzano, aplicado con el propósito de calcular sus raíces—, de modo que se establecen éstas. El corolario del teorema de Rolle garantiza que: cada subintervalo que construyamos —partiendo de las cotas de las raíces de la ecuación—,

empleando como extremos a dos raíces consecutivas de $f'(x)$, contendrá cuando mucho, una raíz de $f(x)$, que resuelve el problema de separar las raíces reales de $f(x)$.

2.2.2.3. MÉTODO DE STURM.

El método de Rolle (tratado anteriormente), requiere hallar las raíces de $f'(x)$, para encontrar las de $f(x)$. Como no siempre es posible realizar esta operación, debemos desarrollar algún método alternativo que, tal vez, se pudiera basar en los fundamentos teóricos de éste.

De este modo, desarrollaremos el método de Sturm, que, generalmente, resuelve este problema en forma exacta. Éste, permite obtener el número exacto de raíces de una ecuación $f(x)=0$ en un intervalo (a,b) , donde $a \wedge b$ son las cotas inferior y superior, respectivamente; estas cotas se consideran previamente halladas, mediante la utilización de los métodos de acotación de raíces (ya vistos).

El método requiere la consideración de una ecuación de coeficientes reales:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

que sólo admite raíces simples (reales o complejas), ya que, en caso contrario, se pueden generar resultados incorrectos.

Además, se hace necesario el conocimiento previo de la teoría de las sucesiones de Sturm, particularizadas para funciones polinomiales. Dicha

información puede encontrarse en el teorema número 7 del apéndice de cálculo de este trabajo, y a él, puede referirse el lector cuando se mencionen conceptos referentes a dichas sucesiones.

Establecidos estos preliminares, podemos decir que el método de Sturm se fundamenta en el siguiente teorema:

TEOREMA DE STURM.

Dada la ecuación algebraica sin raíces múltiples $f(x)=0$, y, además, los números reales $a \wedge b, a < b$, que no anulan al polinomio $f(x)$. Entonces, el número de raíces reales de $f(x)$ en el intervalo (a,b) es igual a la diferencia entre el número de cambios de signos (conocidos también como variaciones) de la sucesión de Sturm para $x=a$ y, el número de variaciones, para $x=b$.

DEMOSTRACIÓN.

Primeramente, aclaremos el concepto de variación. Diremos que una sucesión de Sturm dada presenta una variación de signo, cada vez que el signo de un elemento de la sucesión f_i no coincide (es distinto) al de f_{i+1} , para $i=1,2,\dots,(k-1)$. Al calcular las variaciones, los elementos nulos de la sucesión no se toman en cuenta.

Ahora, refiriéndonos a la demostración, consideremos la sucesión de Sturm:

$$f_1(x), f_2(x), f_3(x), \dots, f_k(x) \quad (2.14)$$

cuando el valor de x fluctúa desde a hasta b , el número de cambios de signos de la sucesión **(2.14)** puede ser afectado solamente por una o más de las funciones que tienen un cero en ese intervalo, pues todas son continuas, por la condición número **1** que cumplen las sucesiones de Sturm (ver apéndice de cálculo, teorema número **7**). Veamos pues, lo que ocurre si r es raíz de algún $f_i(x) = 0$ para $i \neq 1 \wedge i \neq k$.

Por la condición número **4** que cumplen las sucesiones de Sturm (ver apéndice de cálculo, teorema número **7**), se tiene que:

$$f_{i-1}(x)f_{i+1}(x) < 0$$

lo que indica que $f_{i-1}(x) \wedge f_{i+1}(x)$ se mantienen con sus signos opuestos en el intervalo $(r-h, r+h)$, siendo h positivo y lo suficientemente pequeño para que en este intervalo no haya raíces de $f_{i-1}(x) = 0$ ó de $f_{i+1}(x) = 0$. Cuando x varía en $(r-h, r+h)$, los polinomios $f_{i-1}(x), f_i(x) \wedge f_{i+1}(x)$ presentan siempre un cambio de signo (una variación), pues los casos posibles se exponen en las tablas 1 y 2:

TABLA 1

x	$f_{i-1}(x)$	$f_i(x)$	$f_{i+1}(x)$	Número de Variaciones
$r-h$	+	\pm	-	1
r	+	0	-	1
$r+h$	+	\pm	-	1

TABLA 2

x	$f_{i-1}(x)$	$f_i(x)$	$f_{i+1}(x)$	Número de Variaciones
$r-h$	—	\pm	+	1
r	—	0	+	1
$r+h$	—	\pm	+	1

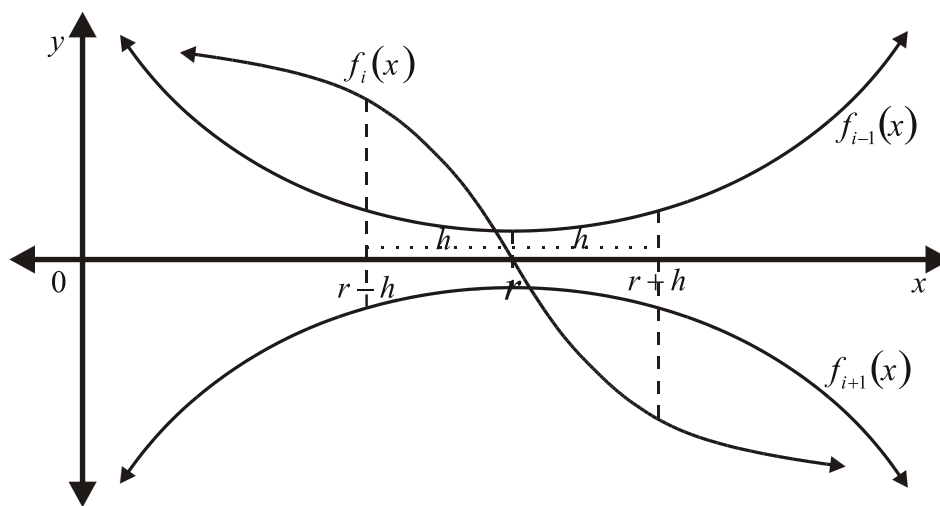
Para ilustrar lo expuesto, presentamos en las figuras 4 y 5, la interpretación geométrica que corresponde al número de variaciones (cambios de signo) que aparecen en las tablas 1 y 2, respectivamente.

Antes, debemos destacar que: el caso del signo \pm , que aparece en ambas tablas origina que, de cada una, se obtenga una figura que consta de dos partes.

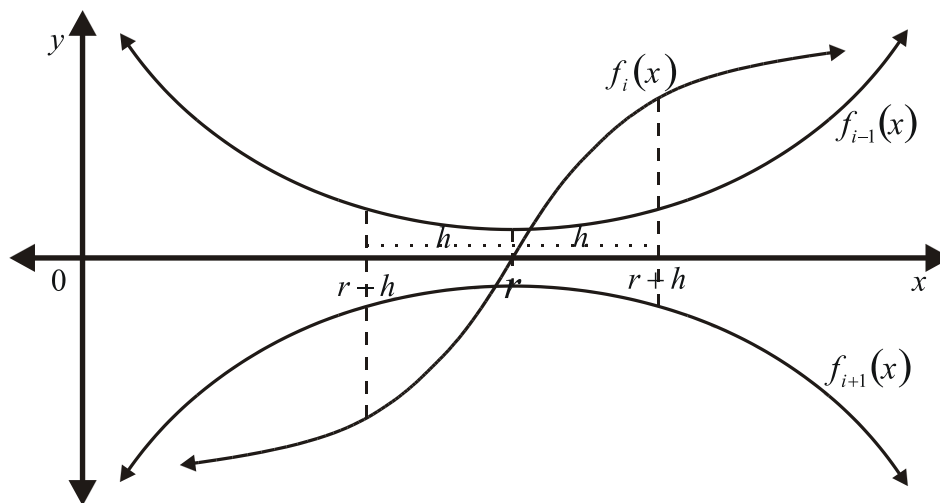
Además, hay que resaltar que para comprender dichas figuras, es necesario tener presentes, en todo momento, las condiciones que cumplen las funciones $f_{i-1}(x), f_i(x) \wedge f_{i+1}(x)$, por ser elementos de una sucesión de Sturm definida en el intervalo (a, b) .

Figura 4

$f_{i-1}(x) > 0$ y $f_{i+1}(x) < 0$
para todo $x \in [r-h, r+h]$



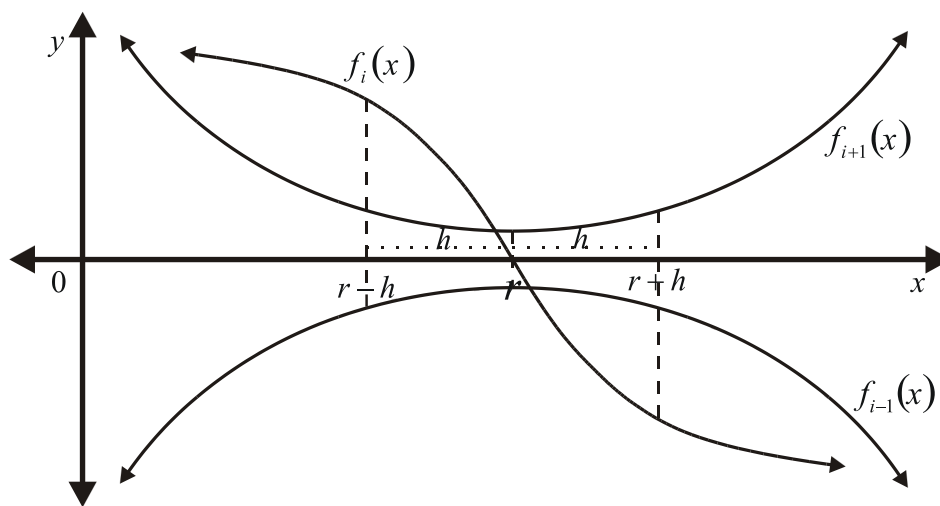
Caso 1: $f_i(r-h) > f_i(r+h)$



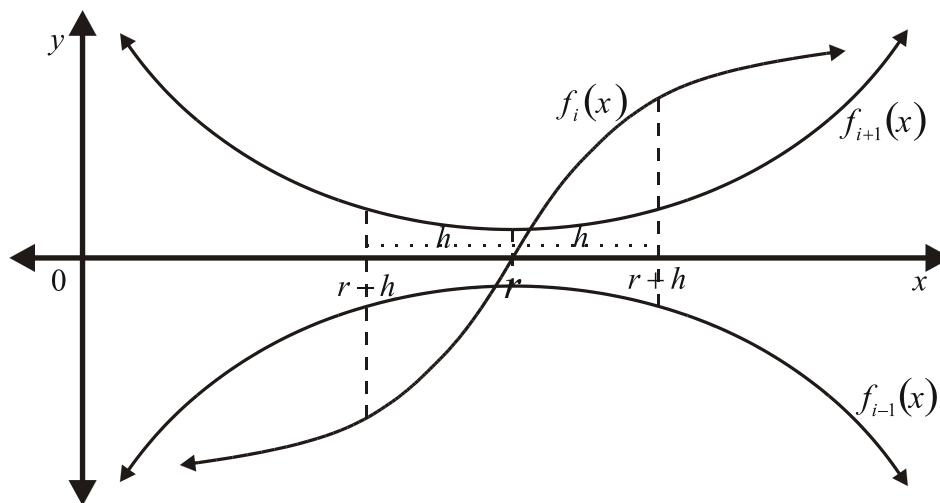
Caso 2: $f_i(r-h) < f_i(r+h)$

Figura 5

$f_{i-1}(x) < 0$ y $f_{i+1}(x) > 0$
para todo $x \in [r-h, r+h]$



Caso 1: $f_i(r-h) > f_i(r+h)$



Caso 2: $f_i(r-h) < f_i(r+h)$

Como en todos los casos se puede observar un cambio de signo, tenemos que: una raíz " r " de este tipo no afecta el número de cambios de signo en el intervalo en estudio, es decir $(r-h, r+h)$, ya que para $x = r-h$ el número de variaciones es uno, al igual que para $x = r+h$.

Si denominamos al número de variaciones de signo que presenta la sucesión de Sturm, para el punto $x = r-h$, como $V(r-h)$ tenemos que $V(r-h) = 1$, y en forma similar $V(r+h) = 1$. Luego, de acuerdo a lo expuesto en el teorema, tendremos que:

$$V(r-h) - V(r+h) = 1 - 1 = 0,$$

lo que indica que la raíz r de $f_i(x) = 0$, $i \neq 1, k$ no afecta la diferencia del número de cambios de signo que sufre la sucesión en el intervalo.

Por otro lado, según la condición **2** de las sucesiones de Sturm (ver apéndice de cálculo, teorema número **7**), la función $f_k(x)$ no puede tener ceros (ya que es una constante no nula por definición).

De este modo, tenemos, por la eliminación sucesiva de las posibilidades, que la única función que podría afectar la diferencia del número de cambios de signo que sufre la sucesión en el intervalo, es $f_1(x) = f(x)$.

Si " s " es una raíz de $f_1(x) = 0$, por la condición **5** de las sucesiones de Sturm (ver apéndice de cálculo, teorema número **7**), podemos decir que $f_1(s-\varepsilon)$ y $f_2(s-\varepsilon)$ tienen signos opuestos, mientras que $f_1(s+\varepsilon)$ y $f_2(s+\varepsilon)$ tienen el mismo signo.

En consecuencia, cuando x varía, asumiendo el valor de una raíz de $f_1(x) = f(x)$, se pierde un cambio de signo entre $f_1(x)$ y $f_2(x)$. Es decir, acontece una variación en el intervalo $(s - \varepsilon, s + \varepsilon)$.

Estas afirmaciones se pueden verificar, al realizar el análisis de las tablas **3** y **4**, que a continuación presentamos.

TABLA 3

x	$f_1(x)$	$f_2(x)$	Número de variaciones
$s - \varepsilon$	+	-	1
s	0	-	0
$s + \varepsilon$	-	-	0

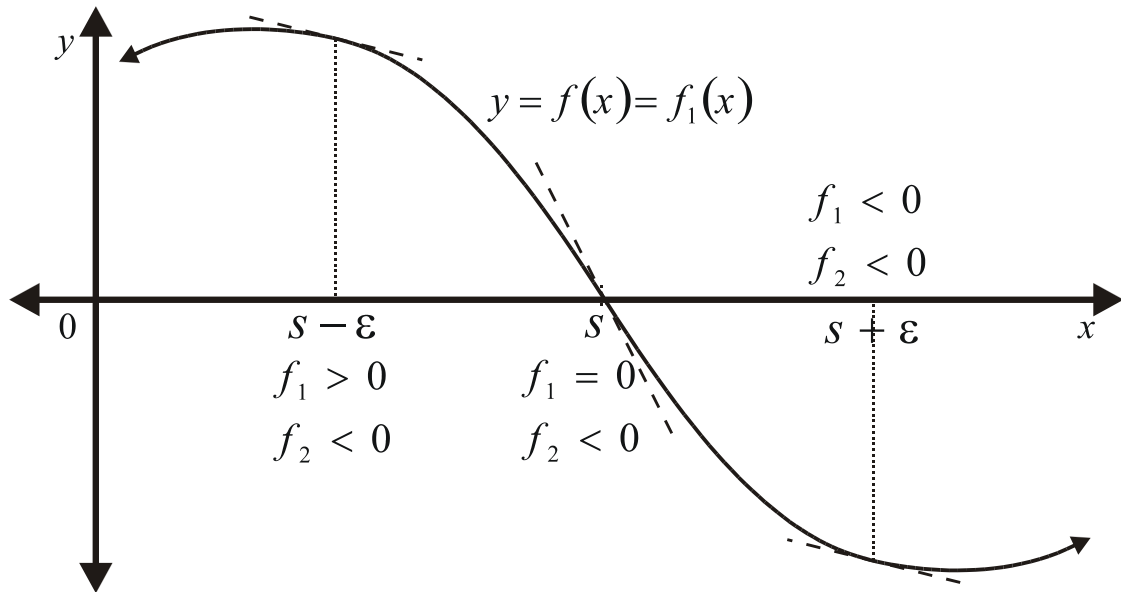
TABLA 4

x	$f_1(x)$	$f_2(x)$	Número de variaciones
$s - \varepsilon$	-	+	1
s	0	+	0
$s + \varepsilon$	+	+	0

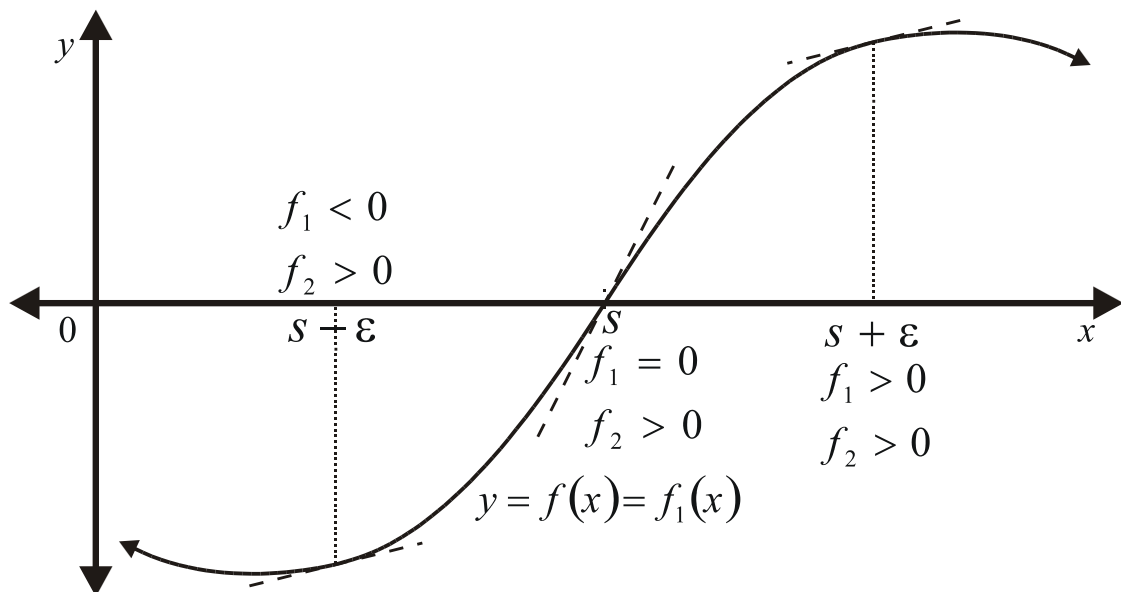
Para aclarar lo expuesto; presentamos en la figura **6** la ilustración geométrica que corresponde al número de variaciones (cambios de signo) que aparece en las tablas **3** y **4**, respectivamente.

Reiteramos que para comprender dicha figura, es necesario tener presentes, en todo momento, las condiciones que cumplen las funciones f_1 y f_2 , por ser elementos de una sucesión de Sturm definida en el intervalo (a, b) .

Figura 6



Caso 1: $f_1(s - \varepsilon) > f_1(s + \varepsilon)$



Caso 2: $f_1(s - \varepsilon) < f_1(s + \varepsilon)$

De este modo, para $x = a$, tenemos que $V(a)$ es el número de variaciones que presenta la sucesión de Sturm. Haciendo variar los valores de x desde " a " hasta " b ", obtenemos que: al tomar x el valor de la raíz $x = s$ en $f_1(x) = f(x)$, se pierde una variación; sin embargo, al pasar por una raíz " r " de $f_i(x) = 0$ (para $i = 2, 3, \dots, k-1$), no aumenta ni disminuye el número de variaciones de la sucesión de Sturm.

Así, si $V(b)$ es el número de variaciones de **(2.14)**, para $x = b$, tenemos que $V(a) - V(b)$ será el número de raíces reales de $f_1(x) = f(x) = 0$ en el intervalo (a, b) , lo que, finalmente, demuestra el teorema.

A continuación, estableceremos, sin demostración, (ver apéndice de teoría de ecuaciones, teorema número **14**) algunos procedimientos útiles que simplifican las operaciones de cálculo, a veces laboriosa, que deben realizarse para obtener la sucesión **(2.14)**.

1. Para evitar fracciones al realizar el proceso de división, en la construcción de los polinomios de Sturm, se pueden multiplicar los coeficientes del dividendo o de cualquier resto parcial por factores positivos, pues esto no cambia el signo de los elementos de la sucesión.
2. Cuando uno de los polinomios de Sturm, digamos $f_i(x)$, mantiene un mismo signo en el intervalo (a, b) , no es necesario tener en cuenta a los polinomios:

$$f_{i+1}(x), f_{i+2}(x), \dots, f_{k-1}(x), f_k(x),$$

pues éstos no alteran el número de variaciones de la sucesión:

$$f_1(x), f_2(x), \dots, f_i(x), \forall x \in (a, b)$$

Esto ocurre cuando $f_i(x) = 0$, en dicho intervalo, sólo tiene raíces complejas.

Ahora, basándonos en los resultados del teorema anterior, podemos establecer el método que nos permitirá separar las raíces reales de una ecuación polinomial cualquiera:

MÉTODO DE STURM.

Dado un polinomio $f(x)$ (que sólo posee raíces simples), y cuyas raíces reales están acotadas en el intervalo (a, b) : se determina el número de raíces reales que están ubicadas en él, al aplicar el teorema de Sturm.

Posteriormente, se consideran valores intermedios entre a y b , para ello, se subdivide el intervalo (a, b) en subintervalos (para este fin, se puede aplicar, por ejemplo, el método de bisección de Bolzano). A cada uno de estos subintervalos se le aplica el teorema de Sturm, y, nuevamente, se subdividen —en nuevos subintervalos— los subintervalos previamente obtenidos.

El procedimiento se reitera hasta que, en cada uno de los subintervalos obtenidos, se encuentre una y sólo una raíz real de $f(x)$.

2.2.2.4. MÉTODOS ALTERNATIVOS.

A continuación, enunciaremos un método que puede usarse para separar raíces reales en polinomios. Es de aplicación muy sencilla y tiene en cuenta el

orden de multiplicidad de éstas; pero no da, como el método de Sturm, el número exacto de raíces de $f(x)$ en el intervalo (a, b) .

TEOREMA DE BUDAN-FOURIER.

Dada la ecuación polinomial $f(x) = 0$ y los números reales a y b , $a < b$, en los que no se anula el polinomio $f(x)$. Entonces, el número de variaciones perdidas por la sucesión:

$$f(x), f'(x), f''(x), \dots, f^n(x)$$

al variar x de a hasta b , es igual al número m de raíces de $f(x) = 0$ en el intervalo (a, b) ; o es mayor que m , en un número entero positivo par. Cada raíz de orden p debe contarse p veces.

Un caso particular de este teorema es la regla de los signos de Descartes. Tiene la ventaja de poderse aplicar sin realizar cálculos.

REGLA DE LOS SIGNOS DE DESCARTES.

El número m de raíces reales positivas de la ecuación algebraica de coeficientes reales:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

(contando r veces cada raíz de orden r), es igual al número de variaciones que presenta la sucesión formada por los coeficientes de la ecuación $f(x) = 0$; o es menor que m , en un número entero positivo par. Para considerar el número

de raíces negativas, se aplica la regla a la ecuación transformada que se obtiene haciendo $x = -y$.

Además, es posible dar un criterio que permite establecer si una ecuación tiene todas sus raíces reales.

Se verifica que: si una ecuación tiene todas sus raíces reales y no nulas, la regla de los signos de Descartes da una respuesta exacta del número de raíces positivas y negativas.

Sean v y v' el número de variaciones de los coeficientes de la ecuación $f(x) = 0$ y de su transformada, respectivamente, se tiene que si $v + v' < n$, el número de raíces complejas de la ecuación es igual a $n - v - v'$, o es menor que $n - v - v'$, por un número entero positivo par.

Luego, la aplicación directa de los teoremas en mención al intervalo particular en que están acotadas las raíces de la ecuación que se analiza y su posterior división en subintervalos, a los que nuevamente se les aplica el teorema, generará los intervalos en que quedarán separadas las raíces de la ecuación en mención (dentro de las restricciones que nos impone el teorema, en lo relativo al grado de exactitud con que se da la respuesta).

2.2.3. APROXIMACIÓN DE RAÍCES REALES.

Desarrollaremos, a continuación, algunos teoremas y algoritmos que se pueden utilizar para la aproximación de raíces reales en ecuaciones algebraicas.

Éstos, en general, requieren que dichas raíces sean acotadas y separadas previamente (además de la supresión de las raíces múltiples de la

ecuación), así, en lo sucesivo, se considerará, si es necesario, que ya se a realizado dicho proceso.

Debe destacarse que: los métodos que aproximan las raíces, en base a los valores que surgen al aplicar un método para separarlas, pueden no converger a éstas (normalmente, esto ocurre por la presencia de raíces complejas, múltiples o muy próximas entre sí).

En tal caso, se debe aplicar reiteradamente el método de separación de raíces fallido (o aplicar otro), hasta lograr que las raíces queden suficientemente separadas entre sí, ya que este paso facilita (como lo veremos en el desarrollo del capítulo) la acción de los métodos de aproximación de las raíces.

Algunos de estos algoritmos son de tipo universal, es decir, aplicables a funciones en general, que sólo cumplen ciertas condiciones, tales como: ser continuas o diferenciables en un intervalo, por ejemplo.

El desarrollo del tema se expondrá empezando con los métodos que presentan las condiciones más débiles (sobre las funciones), para desarrollar de último, los que imponen restricciones más fuertes.

En primer término, haremos referencia a un tema que ya hemos esbozado. Nos referimos al método de bisección de Bolzano, aplicado a la aproximación de raíces reales, que ahora establecemos en términos precisos.

Para facilitar la comprensión de dicho tema, es recomendable que el lector repase los conceptos tratados previamente sobre el teorema de bisección de Bolzano, aplicado a la separación de raíces reales.

2.2.3.1. MÉTODO DE BOLZANO.

Cuando decíamos, en el material tratado previamente, que dividimos el intervalo $[a_k, b_k]$ (en el cual se encuentra una raíz de $f(x) = 0$), en dos partes iguales y evaluamos el polinomio en el punto central, nos referíamos al punto cuya abscisa está dada por:

$$p_{k+1} = \frac{1}{2}[a_k + b_k], k \in N; k \geq 0$$

Por otro lado, debemos observar que el $(k + 1) \in N$, al que hacemos referencia en la ecuación, indica el número de veces que se ha repetido el procedimiento. La serie de repeticiones de este tipo que se deben realizar, para que el método converja al valor buscado, suele denominarse como las iteraciones que el proceso utilizó.

Debe destacarse que: la formulación propia del método, y de su prueba nos garantizan que, siempre que se verifiquen las hipótesis iniciales, el método convergerá a un valor que es raíz de $f(x)$.

Con respecto a otros métodos, suele ser muy lenta la velocidad de convergencia de este método; es decir, k puede llegar a ser un número muy grande, antes que $|b_k - a_k| < \varepsilon$. Sin embargo, éste contiene aspectos muy profundos acerca del tema a exponer, y por eso, será tratado.

El método de bisección de Bolzano, tratado en la sección referente a separación de raíces reales, nos garantiza la convergencia del método a tratar en el caso de la aproximación de raíces (al extrapolar la aplicación del mismo).

En consecuencia, demostraremos un teorema que nos permitirá establecer una cota superior del error ε que posee la raíz aproximada p_k , luego de k iteraciones.

TEOREMA.

Sea $f(x)$ un polinomio de grado n , continuo en el intervalo $[a, b]$ (se denotará $f \in C[a, b]$) y $f(a)$ tiene el signo opuesto al de $f(b)$ (es decir, nos referimos a que $f(a) \cdot f(b) < 0$), entonces, el procedimiento de bisección genera una sucesión:

$$\{p_k\}_{k=1}^{\infty}$$

donde:

$$p_k = \frac{1}{2}[a_{k-1} + b_{k-1}], k \in \mathbb{N} \wedge k \geq 1$$

que converge a la raíz $p \in [a, b]$ de $f(x) = 0$, tal que:

$$|p_k - p| \leq \frac{1}{2^k}[b - a], \text{ para } k \geq 1.$$

DEMOSTRACIÓN.

Primeramente, estableceremos por inducción matemática, como un resultado preliminar en nuestra demostración: el valor numérico de la diferencia

absoluta entre los extremos de un intervalo $[a_k, b_k]$, donde $k \in N$, que ha sido generado a través del empleo del procedimiento de bisección.

Procediendo de este modo, denotemos el intervalo $[a, b]$ como $[a_0, b_0]$, luego, su longitud está dada por:

$$|b_0 - a_0| = [b - a]$$

y su punto de bisección es:

$$p_1 = \frac{1}{2}[a_0 + b_0].$$

Luego, denotando por $[a_1, b_1]$ al primer subintervalo (vea en el procedimiento de bisección, en el apartado **(2.2.2.1.2)** de este trabajo, página 58, las condiciones que deben tenerse en cuenta en la selección de cada nuevo subintervalo), tenemos que su longitud está dada por:

$$|b_1 - a_1| = \frac{1}{2}|b_0 - a_0| = \frac{1}{2}[b - a]$$

y su punto de bisección esta dado por:

$$p_2 = \frac{1}{2}[a_1 + b_1].$$

Ahora, denotando por $[a_2, b_2]$ al segundo subintervalo, tenemos que su longitud está dada por:

$$|b_2 - a_2| = \frac{1}{2}|b_1 - a_1| = \frac{1}{2^2}[b - a]$$

y su punto de bisección por:

$$p_3 = \frac{1}{2}[a_2 + b_2].$$

Prosiguiendo con el proceso, denotemos por $[a_3, b_3]$ al tercer subintervalo, y tendremos así que su longitud está dada por:

$$|b_3 - a_3| = \frac{1}{2}|b_2 - a_2| = \frac{1}{2^3}[b - a]$$

y su punto de bisección por:

$$p_4 = \frac{1}{2}[a_3 + b_3].$$

y en general; si denotamos por $[a_{k-1}, b_{k-1}]$ al $(k - 1)$ -ésimo subintervalo, tenemos que su longitud está dada por:

$$|b_{k-1} - a_{k-1}| = \frac{1}{2}|b_{k-2} - a_{k-2}| = \frac{1}{2^{k-1}}[b - a]$$

siendo su punto de bisección:

$$p_k = \frac{1}{2}[a_{k-1} + b_{k-1}].$$

Ahora, debemos probar que la proposición se verifica para el k -ésimo subintervalo, para lo cual tenemos que:

$$\begin{aligned} |b_k - a_k| &= \frac{1}{2} |b_{k-1} - a_{k-1}| = \frac{1}{2} \cdot \frac{1}{2^{k-1}} [b - a] \\ |b_k - a_k| &= \frac{1}{2^k} [b - a] \end{aligned}$$

lo que completa la prueba del resultado preliminar que necesitamos en nuestra demostración.

Atacando ahora la demostración en sí, debe notarse que, por hipótesis, la raíz p cumple que:

$$p \in [a, b] \wedge p \in [a_{k-1}, b_{k-1}]$$

luego, podemos garantizar que:

$$p \geq a_{k-1}, \Rightarrow p - a_{k-1} \geq 0 \wedge p - a_{k-1} \leq b_{k-1} - a_{k-1}$$

Ahora, calculemos la diferencia absoluta entre la raíz verdadera de la ecuación, y la k -ésima aproximación que hemos obtenido por el procedimiento de bisección, es decir:

$$\begin{aligned} |p_k - p| &= \left| \frac{a_{k-1} + b_{k-1}}{2} - p \right| = \left| \frac{a_{k-1}}{2} + \frac{b_{k-1}}{2} - p \right| = \\ &= \left| \frac{b_{k-1}}{2} - \frac{a_{k-1}}{2} + a_{k-1} - p \right| = \left| \frac{b_{k-1} - a_{k-1}}{2} - [p - a_{k-1}] \right| \end{aligned}$$

Por otro lado, tenemos que:

$$|b_{k-1} - a_{k-1}| = |b_{k-1} - a_{k-1} - 2[p - a_{k-1}] + 2[p - a_{k-1}]|$$

como $p - a_{k-1} \geq 0 \wedge p - a_{k-1} \leq b_{k-1} - a_{k-1}$.

$$\Rightarrow |b_{k-1} - a_{k-1}| \geq |b_{k-1} - a_{k-1} - 2[p - a_{k-1}]|$$

donde la igualdad se cumple para $p = b_{k-1}$.

$$\begin{aligned} \Rightarrow \left| \frac{b_{k-1} - a_{k-1}}{2} \right| &\geq \left| \frac{b_{k-1} - a_{k-1}}{2} - [p - a_{k-1}] \right| \\ \Rightarrow |p_k - p| &= \left| \frac{b_{k-1} - a_{k-1}}{2} - [p - a_{k-1}] \right| \leq \left| \frac{b_{k-1} - a_{k-1}}{2} \right| \\ \Rightarrow |p_k - p| &\leq \frac{1}{2} \left[\frac{1}{2^{k-1}} [b - a] \right] \end{aligned}$$

es decir:

$$|p_k - p| \leq \frac{1}{2^k} [b - a]$$

tomando el límite cuando $k \rightarrow \infty$ en ambos miembros de la ecuación, tenemos que:

$$\lim_{k \rightarrow \infty} |p_k - p| \leq \lim_{k \rightarrow \infty} \frac{1}{2^k} [b - a] = 0$$

Es decir, $|p_k - p| = 0$, cuando k tiende al infinito, de modo que p_k se aproximará, tanto como se desee, a p , lo que demuestra el teorema.

En base a la última desigualdad que obtuvimos, podemos estimar el número de iteraciones que requerirá el procedimiento de bisección, para aproximar la raíz p que buscamos, con un error en la aproximación menor que el ε que fijemos, y viceversa al hacer la mayoración:

$$|p_k - p| \leq \frac{1}{2^k}[b - a] \leq \varepsilon \quad (2.15)$$

y luego se trabaja con el segundo y tercer miembro de **(2.15)** solamente.

Finalmente, debemos resaltar que el procedimiento de bisección —descrito en la sección referente a separación de raíces— establece la forma de operar del método de Bolzano, por lo que, a continuación, no lo estableceremos, como correspondía hacerlo, y referimos al lector a dicha sección, en caso de que no lo tenga claramente establecido.

2.2.3.2. MÉTODO DE RUFFINI-HORNER.

Si analizamos un poco el procedimiento de bisección, nos daremos cuenta que el mismo, al generar la sucesión de intervalos encajados, implícitamente define dos sucesiones monótonas (una creciente y la otra decreciente), que en conjunto definen un número real, que es la raíz que queremos encontrar.

El método de Ruffini-Horner consiste en construir una de estas sucesiones (la creciente), para aproximar la raíz de f comprendida dentro de las condiciones del teorema de bisección. Su construcción se lleva a cabo mediante el empleo de transformaciones que irán trasladando el origen del

sistema de coordenadas a los puntos sucesivos que formarán la sucesión. Estos puntos, serán los extremos inferiores (ínfimos) de los intervalos encajados.

Para simplificar los cálculos en el método, conviene que expresemos como positiva a la raíz que deseamos aproximar. Esto podemos hacerlo con base en la teoría previamente desarrollada en la parte de acotación de raíces, acerca de transformaciones lineales.

Así, consideremos la ecuación:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0 \quad (2.16)$$

que tiene una sola raíz simple real irracional r en el intervalo (α, β) , con $\alpha > 0$ y $\beta > 0$.

Si $\alpha < 0$ y $\beta < 0$, se usan las transformaciones lineales para expresarlas como positivas; y, en caso que α y β sean de signos opuestos, se debe aplicar un método de separación de raíces —como el de Sturm—, que permita ubicar la raíz buscada, dentro de un intervalo cuyos extremos poseen el mismo signo.

Por otro lado, si la ecuación posee raíces racionales, antes de aplicar el método, éstas se deben extraer y se aplica el método a la ecuación reducida. Esta reducción se puede realizar en base al teorema número **15** del apéndice de teoría de ecuaciones, transformando los coeficientes de la ecuación a números enteros y, luego, aplicando el teorema a la ecuación resultante.

Hagamos la convención $f = f_0$ y $x = x_0$. Como sabemos que $r \in (\alpha, \beta)$, podemos decir que:

$$r = \alpha + y \quad (2.17)$$

donde $0 < y < \beta - \alpha$.

La variable y puede ser considerada como el error cometido en la aproximación de la raíz r , mientras que α es el valor que, hasta este momento, consideramos como nuestra aproximación a r .

Desde este punto, la lógica que sigue el método consistirá en: aproximar recursivamente el término de error y que contiene la aproximación de la raíz r que obtenemos, y reemplazarlo en la ecuación (2.17).

Luego, expresando a (2.17) en términos de $(r - \alpha)$:

$$\Rightarrow r - \alpha = y$$

Ahora, desarrollando a $f_0(x_0)$ por la fórmula de Taylor (ver apéndice de cálculo, teorema número 1), según potencias de $(r - \alpha)$, tenemos que la ecuación (2.16) se expresará como:

$$f_0[(r - \alpha) + \alpha] = 0$$

$$f_0[(r - \alpha) + \alpha] = A_n(r - \alpha)^n + A_{n-1}(r - \alpha)^{n-1} + \dots + A_1(r - \alpha) + A_0$$

donde los A_i ($i = 0, 1, \dots, n$) están dados por:

$$A_i = \frac{f_0^i(\alpha)}{i!}$$

y se pueden obtener a través del empleo del esquema de Horner generalizado (ver apéndice de teoría de ecuaciones, teorema número **10**).

Como $r - \alpha = y$, obtenemos la ecuación:

$$f_0(\alpha + y) = f_0(r) = F(y)$$

donde:

$$F(y) = A_n y^n + A_{n-1} y^{n-1} + \dots + A_1 y + A_0 = 0 \quad (2.18)$$

que tiene una raíz simple real en el intervalo $(0, \beta - \alpha)$, ya que la ecuación **(2.18)** es la ecuación **(2.16)**, con sus raíces disminuidas en la cantidad α , y, además, si:

$$\alpha < r < \beta \Rightarrow 0 < r - \alpha < \beta - \alpha$$

Geométricamente, esta operación es equivalente a desplazar el sistema de coordenadas, donde se grafica la ecuación **(2.16)**, hacia la derecha.

Realizando ahora, por conveniencia, la transformación:

$$y^* = \frac{10}{\beta - \alpha} y$$

$$y = \frac{\beta - \alpha}{10} y^* \quad (2.19)$$

y aplicándola sobre F , obtenemos la función:

$$F(y^*) = A_n^*(y^*)^n + A_{n-1}^*(y^*)^{n-1} + \dots + A_1^*(y^*) + A_0^* = 0 \quad (2.20)$$

donde:

$$A_i^* = \frac{A_i(\beta - \alpha)^i}{10^i}$$

para $i = 0, 1, \dots, n$.

En la ecuación **(2.18)**, hemos realizado una transformación sobre su sistema de coordenadas, para ajustar sus raíces —a través de la ecuación **(2.20)**, en el intervalo $(0, 10)$ —, con el objeto de facilitar los cálculos posteriores. De igual manera que antes, si:

$$\begin{aligned} 0 &< y < \beta - \alpha \\ \Rightarrow 0 &< \frac{\beta - \alpha}{10} y^* < \beta - \alpha \\ \Rightarrow 0 &< y^* < 10 \end{aligned}$$

Denotando convenientemente a la función obtenida, diremos ahora que:

$$\begin{cases} f_1 = F \\ x_1 = y^* \end{cases} \quad (2.21)$$

de modo que obtenemos:

$$f_1(x_1) = F(y^*)$$

Probando ahora valores enteros sucesivos desde 0 hasta 10, obtendremos en algún momento dos (llamémoslos α_1 y β_1), para los que $f_1(x_1)$ cambia de signo (realmente, el valor que emplearemos a continuación, en el método, es α_1).

Observemos que α_1 y β_1 son números enteros, y, sin embargo, la raíz de $f_1(x_1)$, que estamos aproximando –que en este momento se encuentra ubicada en (α_1, β_1) –, posee infinitos decimales (en el caso general), de modo que al conocer a α_1 , sólo hemos determinado su parte entera, faltando la determinación de su parte decimal –que, en todo caso, es un número comprendido en $(0, 1)$ –, que denotaremos como $\frac{x_2}{10}$. Esto último, nos indica que debemos hacer el cambio de variable:

$$x_1 = \alpha_1 + \frac{x_2}{10} \quad (2.22)$$

$$\Rightarrow \frac{x_2}{10} = x_1 - \alpha_1$$

donde x_2 es un número comprendido entre 0 y 10 estrictamente.

Luego, al reemplazar convenientemente las ecuaciones (2.19), (2.21) y (2.22) en (2.17), obtendremos que: hasta este punto, el valor de la raíz que estamos aproximando es:

$$r = \alpha + \frac{\beta - \alpha}{10} \alpha_1 + \varepsilon \quad (2.23)$$

donde el término de error en esta aproximación esta dado por:

$$\varepsilon = \frac{\beta - \alpha}{10^2} x_2 \quad (2.24)$$

Desarrollando ahora a f_1 en potencias de $x_1 - \alpha_1$, y asumiendo la convención:

$$f_2\left(\frac{x_2}{10}\right) = f_1(x_1 - \alpha_1)$$

tenemos que:

$$f_2\left(\frac{x_2}{10}\right) = B_n\left(\frac{x_2}{10}\right)^n + B_{n-1}\left(\frac{x_2}{10}\right)^{n-1} + \dots + B_1\left(\frac{x_2}{10}\right) + B_0 = 0$$

donde los B_i , $i = 0, 1, \dots, n$ están dados por:

$$B_i = \frac{f_1^i(\alpha_i)}{i!}$$

Ya que $\frac{x_2}{10} \in (0, 1)$, al calcular $f_2\left(\frac{x_2}{10}\right)$ obtendremos, por lo general, coeficientes con fracciones decimales, de modo que multiplicamos toda la última ecuación por 10^n , con el objeto de suprimir estos decimales.

De este modo, multiplicando la ecuación por 10^n , tenemos que:

$$10^n f_2\left(\frac{x_2}{10}\right) = B_n^*\left(\frac{x_2}{10}\right)^n + B_{n-1}^*\left(\frac{x_2}{10}\right)^{n-1} + \dots + B_1^*\left(\frac{x_2}{10}\right) + B_0^* = 0$$

donde:

$$B_i^* = 10^{n-1} B_i, \quad i = 0, 1, \dots, n$$

Sin embargo, debemos observar que la multiplicación de la ecuación por 10^n suprime, en todas las incógnitas de la ecuación, el denominador de la variable x_2 (que forma parte de nuestro término de error definido en **(2.24)**). Es decir, que la ecuación queda sólo en término de ésta, y, por hipótesis $x_2 \in (0, 10)$, de modo que ahora la ecuación tiene una raíz entre 0 y 10.

Probando ahora valores enteros sucesivos desde 0 hasta 10, obtendremos en algún momento dos (llamémoslos α_2 y β_2), para los que:

$$f_2^*(x_2) = 10^n f_2\left(\frac{x_2}{10}\right)$$

cambia de signo. Entonces, hacemos nuevamente:

$$x_2 = \alpha_2 + \frac{x_3}{10} \tag{2.25}$$

$$\Rightarrow \frac{x_3}{10} = x_2 - \alpha_2$$

donde x_3 es un número comprendido entre 0 y 10 (esta situación se justifica por las mismas razones que nos permitieron justificarla para el caso de x_2).

Nuevamente, al reemplazar convenientemente las ecuaciones **(2.24)** y **(2.25)** en **(2.23)**, obtendremos que: hasta este punto, el valor de la raíz que estamos aproximando es:

$$r = \alpha + \frac{\beta - \alpha}{10} \alpha_1 + \frac{\beta - \alpha}{10^2} \alpha_2 + \varepsilon \quad (2.26)$$

donde el término de error en esta aproximación está dado por:

$$\varepsilon = \frac{\beta - \alpha}{10^3} x_3 \quad (2.27)$$

Desarrollando ahora a $f_2^*(x_2)$ en potencias de $x_2 - \alpha_2$, y asumiendo la convención:

$$f_3\left(\frac{x_3}{10}\right) = f_2^*(x_2 - \alpha_2)$$

tenemos que:

$$f_3\left(\frac{x_3}{10}\right) = C_n\left(\frac{x_3}{10}\right)^n + C_{n-1}\left(\frac{x_3}{10}\right)^{n-1} + \dots + C_1\left(\frac{x_3}{10}\right) + C_0 = 0$$

donde las C_i ($i = 0, 1, \dots, n$) están dadas por:

$$C_i = \frac{[f_2^*](\alpha_2)}{i!}$$

que multiplicada por 10^n , se puede expresar como:

$$10^n f_3\left(\frac{x_3}{10}\right) = C_n^* \left(\frac{x_3}{10}\right)^n + C_{n-1}^* \left(\frac{x_3}{10}\right)^{n-1} + \dots + C_1^* \left(\frac{x_3}{10}\right) + C_0^* = 0$$

donde:

$$C_i^* = 10^{n-1} C_i, \quad i = 0, 1, \dots, n$$

que tiene una raíz entre 0 y 10, por las razones que ya citamos, para el caso de x_2 .

Probando ahora valores sucesivos desde 0 hasta 10, obtendremos en algún momento dos valores sucesivos (llamémoslos α_3 y β_3), para los que:

$$f_3^*(x_3) = 10^n f_3\left(\frac{x_3}{10}\right)$$

cambia de signo. Entonces, hacemos nuevamente:

$$x_3 = \alpha_3 + \frac{x_4}{10} \tag{2.28}$$

$$\Rightarrow \frac{x_4}{10} = x_3 - \alpha_3$$

donde x_4 es un número comprendido entre 0 y 10 estrictamente.

Nuevamente, al reemplazar convenientemente las ecuaciones **(2.27)** y **(2.28)** en **(2.26)**, obtendremos que: hasta este punto, el valor de la raíz que estamos aproximando es:

$$r = \alpha + \frac{\beta - \alpha}{10} \alpha_1 + \frac{\beta - \alpha}{10^2} \alpha_2 + \frac{\beta - \alpha}{10^3} \alpha_3 + \varepsilon$$

donde el término de error en esta aproximación está dado por:

$$\varepsilon = \frac{\beta - \alpha}{10^4} x_4$$

Repitiendo el proceso antes descrito en forma iterativa, llegará el momento en que, basados en una cota mínima para el error que contendrá la aproximación de la raíz buscada (por ejemplo, la demostrada previamente para el método de bisección), se obtendrá la misma, con un error en la aproximación, menor que el establecido previamente.

El cálculo de los coeficientes de cada uno de los polinomios transformados se puede realizar aplicando el denominado esquema de Horner, detallado en el teorema número 10 del apéndice de teoría de ecuaciones.

Como este método se desarrolló basándose en su parte fundamental en el algoritmo de bisección (pudiéndosele considerar como una representación alternativa de éste, especialmente adaptada para ser aplicada en polinomios), sus condiciones de convergencia son similares a las de éste, por lo que no las desarrollaremos.

Una vez que se obtiene la raíz aproximada, las restantes raíces del polinomio se pueden obtener, al extraer de éste, por medio de la división sintética, la raíz obtenida, y aplicando nuevamente el esquema completo, al polinomio que resulta.

2.2.3.3. MÉTODO DE APROXIMACIONES SUCESIVAS (ITERACIÓN DE PUNTO FIJO).

El método de aproximaciones sucesivas (también conocido como iteración de punto fijo), tiene sus fundamentos (desde un punto de vista teórico) en la denominada teoría de las contracciones.

Un breve recuento de ésta se encuentra en el teorema número **8** del apéndice de cálculo de este trabajo, y (a menos que el lector tenga conocimientos previos de ella) es conveniente que se revisen sus conceptos fundamentales, para facilitar la comprensión del material tratado en esta sección.

Recordemos que $f(x)$ y $P(x)$ serán las representaciones para las ecuaciones algebraicas que mencionemos en las demostraciones.

Primeramente, debemos observar que toda ecuación polinomial:

$$f(x) = 0 \tag{2.29}$$

se puede expresar como otra función:

$$F(x) = x \tag{2.30}$$

donde:

$$F(x) = f(x) + x$$

Debe observarse, entonces, que todo valor de la variable x que anule a **(2.29)** (digamos $x = a$, de modo que $f(a) = 0$), originará en **(2.30)** una igualdad tal como:

$$F(a) = a \tag{2.31}$$

Ahora, por definición, todo valor que satisface **(2.31)** se denominará punto fijo de F , y es evidente, entonces, que, todo punto fijo de **(2.30)** es raíz de **(2.29)** y viceversa. Luego, el problema de encontrar las raíces de una ecuación se puede reducir a encontrar los puntos fijos de otra ecuación.

Tenemos que: encontrar los puntos fijos de una ecuación —como **(2,30)** en el caso general— equivale geométricamente, a resolver un sistema de ecuaciones simultáneas (no lineales en el caso general), compuesto por las ecuaciones:

$$\begin{cases} y = x \\ y = F(x) \end{cases}$$

es decir, encontrar el o los puntos de intersección de la recta $y = x$ con la curva $y = F(x)$.

Este procedimiento geométrico se puede establecer, en los siguientes términos: A partir de un punto situado sobre el eje x de abscisa x_0 , se traza su

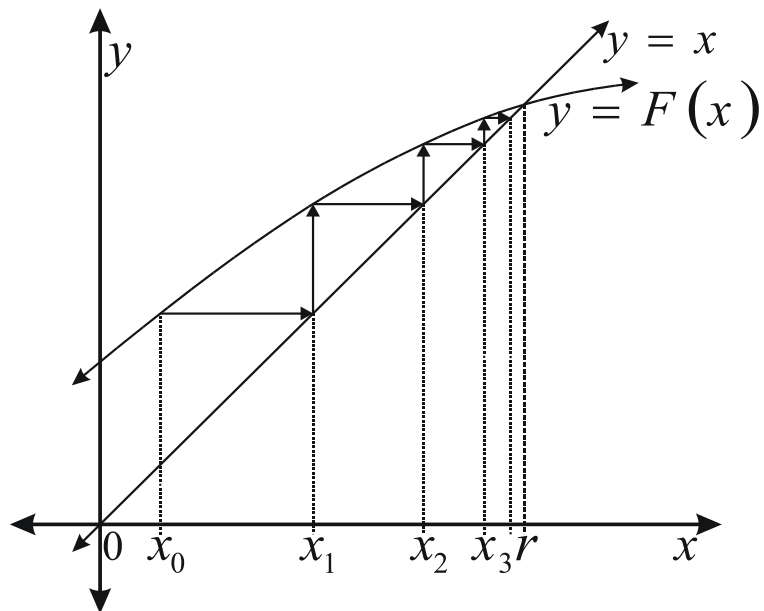
proyección vertical hasta cortar a la curva $y = F(x)$. Por este punto de corte, se traza una horizontal hasta cortar a la recta $y = x$, luego, se proyecta este último punto sobre el eje x y su abscisa será la nueva aproximación (llamada x_1).

Repitiendo este proceso a partir de la nueva abscisa x_1 , se obtiene x_2 ; de x_2 , se obtiene x_3 ; es decir, el procedimiento, consiste en ir de una aproximación cualquiera, en forma vertical, a la curva $y = F(x)$; y de ésta, horizontalmente, a la recta $y = x$, y así sucesivamente, hasta obtener un valor aproximado del punto fijo de **(2.30)**, con la precisión deseada.

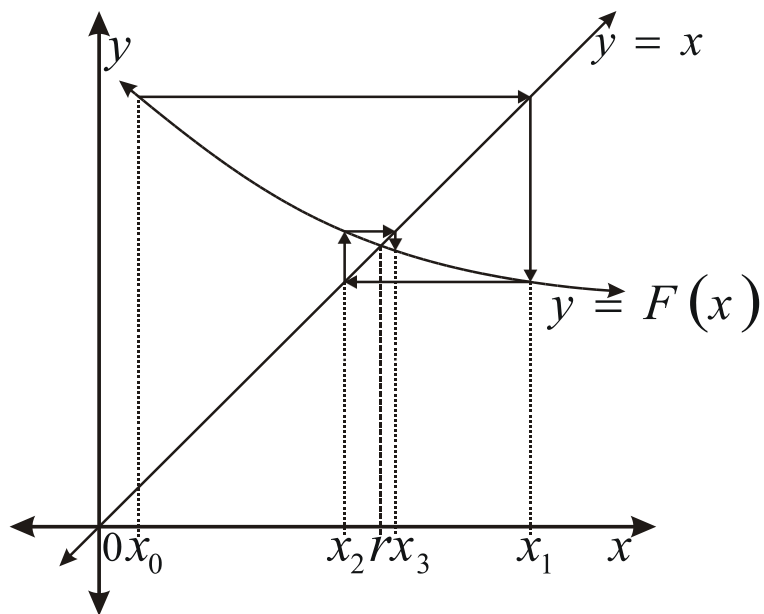
Según lo antes dicho, la convergencia o divergencia del procedimiento anterior depende, exclusivamente, de la forma de la curva $F(x)$ en las cercanías de un intervalo I , donde I contiene a x_0 . En las figuras **7** y **8** mostramos algunas de las situaciones que se pueden dar, en función de los valores de la derivada de F en el intervalo I .

Debemos resaltar que: el intervalo I contiene las secciones del eje x que incluyen a las figuras mostradas, por lo que no se muestran los extremos de dicho intervalo.

Figura 7

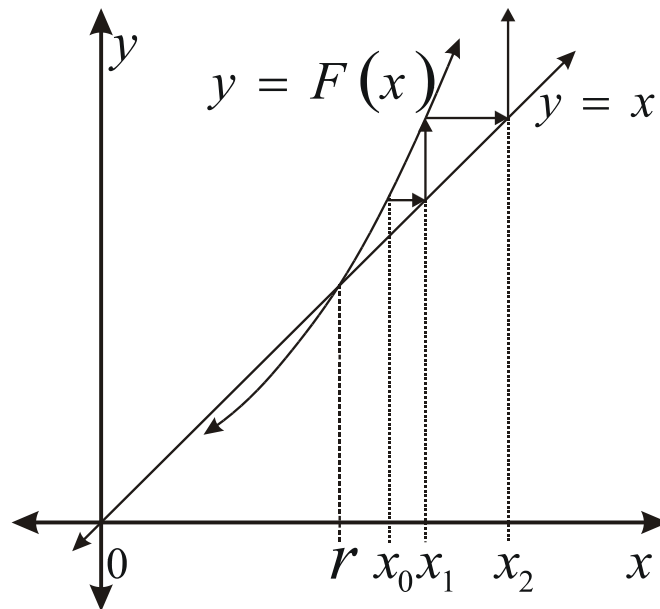


Caso 1: $0 < F'(x) < 1$, hay convergencia

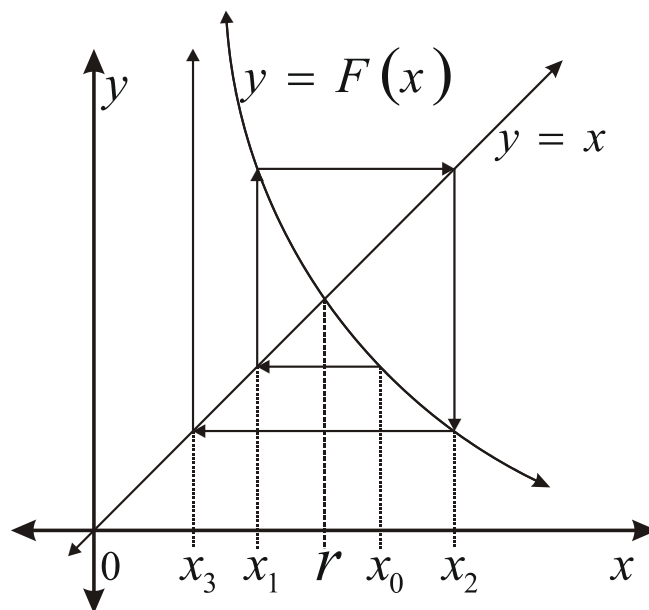


Caso 2: $-1 < F'(x) < 0$, hay convergencia

Figura 8



Caso 3: $F'(x) > 1$, no hay convergencia



Caso 4: $F'(x) < -1$, no hay convergencia

En primer término, observemos que la secuencia:

$$x_0, x_1, x_2, \dots$$

define una sucesión de valores reales (en función de las proyecciones sobre $F(x)$ y la recta $y = x$).

Puede notarse en las figuras **7** y **8** la existencia de casos, en los que las proyecciones trazadas entre F y la recta en mención van desplazando nuestra aproximación de la raíz buscada, en el sentido positivo del eje X , dándose la convergencia en unas situaciones, o la divergencia en otras.

Por otro lado, también hay casos en los que las proyecciones desplazan nuestra aproximación de la raíz, en forma alternada, unas veces en sentido positivo del eje X , otras veces en sentido negativo; ocurriendo la convergencia en unos casos y la divergencia en otros.

Y, finalmente, pareciera que en ciertos casos donde hubo divergencia, con una mejor aproximación inicial, se habría logrado la convergencia deseada.

Estas situaciones que hemos apreciado en forma geométrica nos dejan desconcertados, ya que, a primera vista, no sugieren las condiciones matemáticas precisas que debe cumplir la función F en el intervalo I , que nos garanticen la convergencia o divergencia de la sucesión de aproximaciones sucesivas.

Sin embargo, debemos observar que siempre que en el intervalo I se cumple, la desigualdad:

$$|F'(x)| \leq \alpha < 1, \forall x \in I = [a, b] \quad (2,32)$$

se verificará la convergencia hacia un único punto fijo de la función F en el intervalo I , dicha desigualdad establece que la función $F(x)$ posee su derivada acotada en dicho intervalo.

La condición antes mencionada tiene un sentido geométrico muy sencillo: El teorema del valor medio (ver apéndice de cálculo, teorema número 9) nos garantiza que en el intervalo I , a cada recta secante de la función $F(x)$, se le puede yuxtaponer una recta paralela, tangente a dicha función. Entonces, la condición **(2.32)** nos garantiza que ninguna de las rectas secantes tendrá mayor pendiente que α , es decir, estarán acotadas.

Las consideraciones geométricas previamente analizadas, se pueden establecer en términos matemáticos precisos. Para tal efecto, a continuación, demostraremos un teorema, que se fundamenta en la teoría de contracciones, y nos establece condiciones suficientes que garantizan la convergencia de la función $F(x)$ a la solución de **(2.30)**. Dichas condiciones no son necesarias (pueden encontrarse funciones $F(x)$ que no las cumplen y, sin embargo, convergen a la solución de **(2.30)**).

TEOREMA.

Sea $I = [a, b]$ un intervalo cerrado, y sea F una función que satisface las siguientes condiciones:

1. F es continua en I ;
2. $F(x) \in I, \forall x \in I$;

3. F satisface la condición de Lipschitz (ver apéndice de cálculo, teorema número 8) dada por:

$$|F(x) - F(y)| \leq \alpha |x - y|, \forall x, y \in I, \alpha \in (0,1)$$

Entonces, para cualquier elección de $x_0 \in I$ la sucesión definida por la fórmula:

$$x_k = F(x_{k-1}) \quad (2.33)$$

converge a la solución única s de la ecuación (2.30).

DEMOSTRACIÓN.

Primeramente, probaremos que el punto fijo de la ecuación (2.30), antes mencionada, existe en el intervalo I , en base a las condiciones establecidas previamente.

Consideremos la función definida por la ecuación:

$$g(x) = x - F(x), x \in I = [a, b]$$

por las condiciones 1 y 2, tenemos que F es continua en $I = [a, b]$, además de que: $a \leq F(a) \leq b \wedge a \leq F(b) \leq b$, por ser $F(x) \in I, \forall x \in I$. Luego, podemos decir que:

$$g(a) \leq 0 \wedge g(b) \geq 0$$

(por ser F continua, podemos decir que g también lo es), luego, el teorema de Bolzano nos garantiza la existencia de un punto $s \in [a, b]$ tal que $g(s) = 0$ lo que implica, a su vez, que $F(s) = s$, y garantiza la existencia de la solución buscada.

A continuación, demostraremos que: al verificarse que la función F cumple la condición de Lipschitz (ver apéndice de cálculo, teorema 8) dada por:

$$|F(x) - F(y)| \leq \alpha |x - y|, \forall x, y \in I; \alpha \in (0, 1);$$

se garantiza que F cumple la condición **(2.32)**, donde el valor de α es el mismo en ambos casos.

Como F cumple la condición de Lipschitz, tenemos que:

$$|F(x) - F(y)| \leq \alpha |x - y|, \forall x, y \in I; \alpha \in (0, 1)$$

reescribiendo la ecuación, tenemos que:

$$\begin{aligned} \Rightarrow \frac{|F(x) - F(y)|}{|x - y|} &\leq \alpha \\ \Rightarrow \left| \frac{F(x) - F(y)}{x - y} \right| &\leq \alpha \end{aligned} \tag{2.34}$$

como la última desigualdad se verifica $\forall x, y \in I$ (en particular para $x > y$), donde $\alpha \in (0, 1)$, el teorema del valor medio para derivadas (ver apéndice de cálculo, teorema número 9), nos garantiza que: $\exists \varepsilon \in I$, tal que:

$$F(x) - F(y) = F'(\varepsilon)(x - y)$$

o, lo que es igual:

$$\frac{F(x) - F(y)}{x - y} = F'(\varepsilon) \quad (2.35)$$

luego, de (2.35) y (2.34), se obtiene que:

$$|F'(\varepsilon)| \leq \alpha$$

es decir, que si se cumple la condición de Lipschitz en el intervalo I , se garantiza que también se cumple la condición (2.32), para un mismo valor de α .

Ahora mostraremos que: al cumplir F la condición de Lipschitz (con $0 < \alpha < 1$) en el intervalo I , se garantiza que la solución de (2.30) es única.

De este modo, —reduciendo al absurdo el problema planteado—, supongamos que existen dos soluciones distintas para (2.30), $s_1, s_2 \in I$, podemos entonces decir, por (2.31) que:

$$\begin{cases} s_1 = F(s_1) \\ s_2 = F(s_2) \end{cases}$$

donde $s_1 \neq s_2$. Reduciendo el sistema a una ecuación, podemos decir que:

$$s_1 - s_2 = F(s_1) - F(s_2)$$

aplicando valores absolutos:

$$\Rightarrow |s_1 - s_2| = |F(s_1) - F(s_2)| \quad (2.36)$$

Como F satisface la condición de Lipschitz, tenemos que $\exists \alpha \in \mathbb{R}, 0 < \alpha < 1$, tal que:

$$|F(s_1) - F(s_2)| \leq \alpha |s_1 - s_2| \quad (2.37)$$

En consecuencia, de (2.36) y (2.37) podemos decir que:

$$|s_1 - s_2| \leq \alpha |s_1 - s_2| \quad (2.38)$$

Por hipótesis, $s_1 \neq s_2$, de modo que (2.38), se reduce a $1 \leq \alpha$, lo que contradice una de las hipótesis iniciales de la condición de Lipschitz ($1 > \alpha > 0$), lo que es absurdo; completando la prueba de que la solución es única.

Hasta este punto, hemos probado que: si se verifican las condiciones iniciales del teorema, existe una única solución en el intervalo I para la ecuación F (es decir, existe un único punto $s \in I$, tal que $F(s) = s$).

A continuación, probaremos que $\forall x_0 \in I$, la sucesión cuyo k -ésimo término está dado por la ecuación (2.33), converge a la solución de F en el intervalo I (es decir, al punto $s \in I$, tal que $F(s) = s$, el único punto fijo de F en el intervalo I).

Para lograr esto, calculemos la diferencia absoluta de la solución real de la ecuación (es decir, s) y la solución aproximada (es decir, x_k). Así, tenemos que:

$$\begin{cases} s = F(s) \\ x_k = F(x_{k-1}) \end{cases} \quad (2.39)$$

La diferencia en mención está dada por:

$$|x_k - s| \quad (2.40)$$

substituyendo las incógnitas del sistema de ecuaciones **(2.39)** en la expresión **(2.40)**, tenemos que:

$$\Rightarrow |x_k - s| = |F(x_{k-1}) - F(s)| \quad (2.41)$$

y por la condición de Lipschitz, obtenemos que:

$$\Rightarrow |F(x_{k-1}) - F(s)| \leq \alpha |x_{k-1} - s| \quad (2.42)$$

luego, de **(2.41)** y **(2.42)** podemos decir que:

$$\Rightarrow |x_k - s| \leq \alpha |x_{k-1} - s|$$

realizando sobre $|x_{k-1} - s|$ el mismo razonamiento que aplicamos a $|x_k - s|$, en forma reiterada, podemos decir que:

$$|x_k - s| \leq \alpha |x_{k-1} - s| \leq \alpha^2 |x_{k-1} - s|$$

$$|x_k - s| \leq \alpha |x_{k-1} - s| \leq \alpha^2 |x_{k-2} - s| \leq \alpha^3 |x_{k-3} - s| \leq \dots \leq \alpha^k |x_0 - s|$$

\vdots

y, finalmente, podemos reducir la serie de desigualdades a:

$$|x_k - s| \leq \alpha^k |x_0 - s|$$

Tomando ahora el límite cuando $k \rightarrow \infty$, tenemos que:

$$\lim_{k \rightarrow \infty} |x_k - s| \leq \lim_{k \rightarrow \infty} \alpha^k |x_0 - s| \quad (2.43)$$

como $0 < \alpha < 1$ por hipótesis, tenemos que: $\alpha^k \rightarrow 0$ en el caso que $k \rightarrow \infty$, de modo que (2.43) se reduce a:

$$\lim_{k \rightarrow \infty} |x_k - s| \leq \lim_{k \rightarrow \infty} \alpha^k |x_0 - s| = 0$$

es decir, la diferencia entre s y x_k podrá llegar a ser tan pequeña como se quiera, siempre que la fórmula de recurrencia (2.33) se aplique el número de veces apropiado (k), lo que finaliza la demostración.

A continuación, demostraremos un teorema que nos permitirá establecer una cota superior del error ε que posee la raíz aproximada x_k , luego de (k) iteraciones.

TEOREMA.

Sea $F(x)$ un polinomio de grado n , que cumple las condiciones del teorema anterior. Entonces el procedimiento de aproximaciones sucesivas genera una sucesión:

$$\{x_n\}_{k=0}^{\infty}$$

donde x_k está dado por la ecuación **(2.33)**, que converge a la raíz $s \in [a, b]$ de $f(x) = 0$, tal que:

$$|x_k - s| \leq \frac{\alpha^k}{1 - \alpha} |x_1 - x_0|$$

DEMOSTRACIÓN.

Para realizar la demostración, requerimos de un resultado auxiliar, que a continuación probamos. Así, demostraremos que $\forall k \in \mathbb{N}$:

$$|x_{k+1} - x_k| \leq \alpha^k |x_1 - x_0|, \quad \alpha \in \mathbb{R}^+$$

Es evidente que la afirmación es cierta para $k = 0$, ya que:

$$|x_1 - x_0| \leq \alpha^0 |x_1 - x_0|$$

Ahora, suponiendo la validez del resultado para $k = m - 1$, tenemos que:

$$|x_m - x_{m-1}| \leq \alpha^{m-1} |x_1 - x_0| \quad (2.44)$$

Luego, para $k = m$, podemos decir —por **(2.33)**—, que:

$$|x_{m+1} - x_m| = |F(x_m) - F(x_{m-1})| \quad (2.45)$$

Además, la condición de Lipschitz nos permite decir que:

$$|F(x_m) - F(x_{m-1})| \leq \alpha |x_m - x_{m-1}|$$

Empleando ahora a **(2.44)**, tenemos que:

$$|F(x_k) - F(x_{k-1})| \leq \alpha [\alpha^{k-1} |x_1 - x_0|] \quad (2.46)$$

Finalmente, al reemplazar **(2.46)** en **(2.45)**, haciendo $m = k$, y simplificando la expresión resultante, obtenemos que:

$$|x_{k+1} - x_k| \leq \alpha^k |x_1 - x_0| \quad (2.47)$$

que es el resultado auxiliar que deseabamos demostrar.

Calculemos ahora la diferencia —en valor absoluto—, entre dos iteraciones cualesquiera (no sucesivas), $x_m \wedge x_k$, donde $m > k$, es decir:

$$\pi = |x_m - x_k| \quad (2.48)$$

Sumando y restando x_i ($i = m-1, m-2, \dots, k+1$) a **(2.48)** tenemos que:

$$\pi = |(x_m - x_{m-1}) + (x_{m-1} - x_{m-2}) + \dots + (x_{k+1} - x_k)| \quad (2.49)$$

Aplicando la desigualdad triangular generalizada a la desigualdad **(2.49)** (ver apéndice de cálculo teorema número **10**), tenemos que:

$$\pi \leq |x_m - x_{m-1}| + |x_{m-1} - x_{m-2}| + \dots + |x_{k+1} - x_k| \quad (2.50)$$

Ahora, por **(2.47)** podemos estimar cada uno de los sumandos de la expresión **(2.50)**, y, factorizando la expresión resultante por $|x_1 - x_0|$, podemos decir que:

$$\pi \leq (\alpha^{m-1} + \alpha^{m-2} + \alpha^{m-3} + \dots + \alpha^{k+1} + \alpha^k) |x_1 - x_0| \quad (2.51)$$

factorizando a **(2.51)**, por α^k , reordenando los términos de esta expresión y reemplazando el valor de π dado por **(2.48)**, tenemos que:

$$|x_m - x_k| \leq \alpha^k (1 + \alpha + \alpha^2 + \dots + \alpha^{m-k-1}) |x_1 - x_0|$$

Tomando ahora el límite cuando $m \rightarrow \infty$, y manteniendo a k fija, obtenemos que:

$$\lim_{m \rightarrow \infty} |x_m - x_k| \leq \lim_{m \rightarrow \infty} \alpha^k (1 + \alpha + \alpha^2 + \dots + \alpha^{m-k-1}) |x_1 - x_0|$$

extrayendo las constantes del límite, tenemos que:

$$\lim_{m \rightarrow \infty} |x_m - x_k| \leq \alpha^k |x_1 - x_0| \lim_{m \rightarrow \infty} (1 + \alpha + \alpha^2 + \dots + \alpha^{m-k-1})$$

es decir:

$$\lim_{m \rightarrow \infty} |x_m - x_k| = \lim_{m \rightarrow \infty} |x_k - x_m| \leq \alpha^k |x_1 - x_0| \sum_{i=0}^{\infty} \alpha^i$$

Ahora, en el miembro derecho de **(2.52)**, tenemos una serie geométrica convergente, (ver apéndice de cálculo, teorema número **11**) ya que por hipótesis, $0 < \alpha < 1$, luego, como la serie es convergente, tenemos que:

$$\lim_{m \rightarrow \infty} |x_k - x_m| \leq |x_1 - x_0| \frac{\alpha^k}{1 - \alpha} \quad (2.53)$$

Por otro lado, como el teorema anterior nos garantiza que: cuando $m \rightarrow \infty$, la m -ésima iteración de **(2.33)** tiende a la solución de **(2.30)**, es decir, $x_m \rightarrow s$, luego, al introducir el límite dentro del valor absoluto del miembro derecho de **(2.53)** y aplicar lo antes dicho, podemos decir finalmente, que:

$$|x_k - s| \leq |x_1 - x_0| \frac{\alpha^k}{1 - \alpha} \quad (2.54)$$

lo que demuestra finalmente el teorema.

En base a la desigualdad **(2.54)**, podemos estimar el número de iteraciones que se requerirá para que la sucesión de aproximaciones sucesivas nos permita aproximar la raíz s que buscamos, con un error en la aproximación menor que el ε que fijemos, y viceversa, al hacer la mayoración indicada en:

$$|x_k - s| \leq |x_1 - x_0| \frac{\alpha^k}{1 - \alpha} < \varepsilon \quad (2.55)$$

y luego se trabaja con el segundo y tercer miembro de **(2.55)**, solamente.

En base a los teoremas previamente demostrados, podemos establecer el método de aproximaciones sucesivas en los siguientes términos:

Partiendo de una ecuación $f(x)=0$, que posee sus raíces separadas —por ejemplo, a través de alguno de los métodos tratados con anterioridad—, para cada intervalo que contiene una raíz de f , se reescribe la ecuación según la

forma $x = F(x)$, de modo que ésta cumpla con las condiciones de convergencia establecidas. Luego, aplicamos la fórmula recursiva **(2.33)**, empleando como valor inicial cualquiera de los puntos del intervalo en mención. El procedimiento se repite hasta obtener la raíz buscada, con la exactitud deseada —en función de la desigualdad **(2.55)**—.

Si alguno de los subintervalos que contiene una raíz de f no cumple con los requisitos exigidos por el teorema de convergencia, se debe reiterar el procedimiento de separación de raíces en ese subintervalo, hasta lograr que los cumpla.

2.2.3.4. MÉTODO DE NEWTON-RAPHSON-FOURIER.

A continuación, trataremos uno de los métodos numéricos, más eficaces, que nos permite aproximar las raíces de una ecuación algebraica o trascendente. Éste implica evaluar la derivada de la función a la que deseamos encontrar sus raíces.

Nos referimos al método de Newton-Raphson-Fourier, que es equivalente al conocido método de las tangentes, pero considerando las precauciones de Fourier. Primeramente lo desarrollaremos a través de su interpretación geométrica, para luego aplicarlo en el cálculo aproximado de las raíces de una ecuación algebraica polinomial de la forma:

$$y = f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0 \quad \textbf{(2.56)}$$

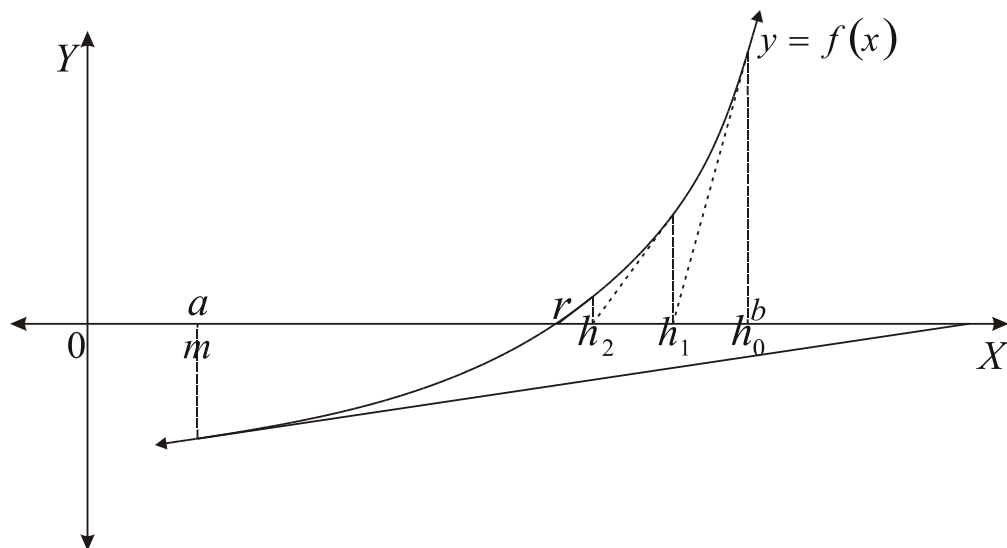
Consideremos que:

1. La ecuación **(2.56)** tiene un cero simple real en el intervalo $[a, b]$, esta consideración se hace efectiva mediante el empleo de algún método de separación de raíces, de los ya vistos.
2. $f(a)$ y $f(b)$ sean distintos de cero.
3. Existen $f'(x)$ y $f''(x)$, pero no se anulan en el intervalo (a, b) .

Damos inmediatamente una interpretación geométrica de este método, lo que nos facilita la comprensión de los conceptos necesarios para su posterior desarrollo analítico.

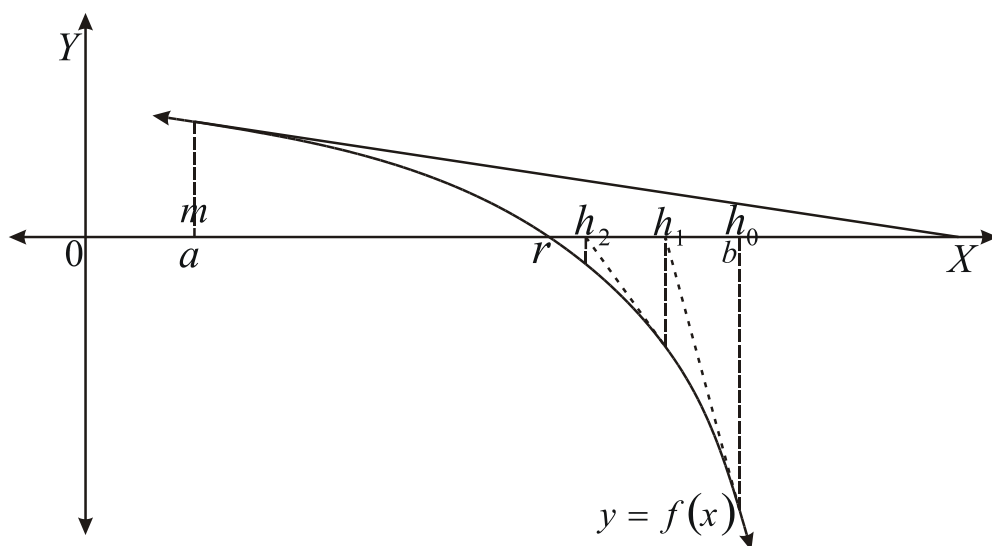
Las consideraciones hechas a **(2.56)** nos indican que la curva representativa de la ecuación $y = f(x)$ en el intervalo $[a, b]$, puede ser creciente o decreciente; sólo atraviesa una vez al eje X ; no posee extremos relativos (máximos o mínimos); no posee puntos de inflexión, pues si $f''(x) > 0$, la curva es cóncava hacia arriba, y, si $f''(x) < 0$, es cóncava hacia abajo (vea las figuras **9** y **10**) —el lector puede, además, consultar cualquier texto de cálculo para aclaraciones adicionales—.

Figura 9



$f''(x) > 0$ para todo $x \in [a, b]$, $f(b) > 0$, $f''(b) \cdot f(b) > 0$

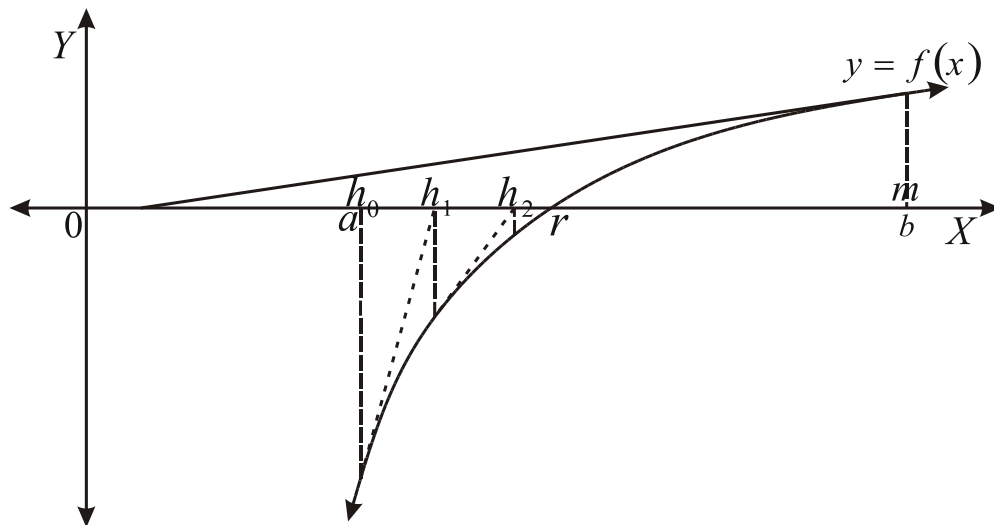
entonces nuestra primera aproximación es $h_0 = b$



$f''(x) < 0$ para todo $x \in [a, b]$, $f(b) < 0$, $f''(b) \cdot f(b) > 0$

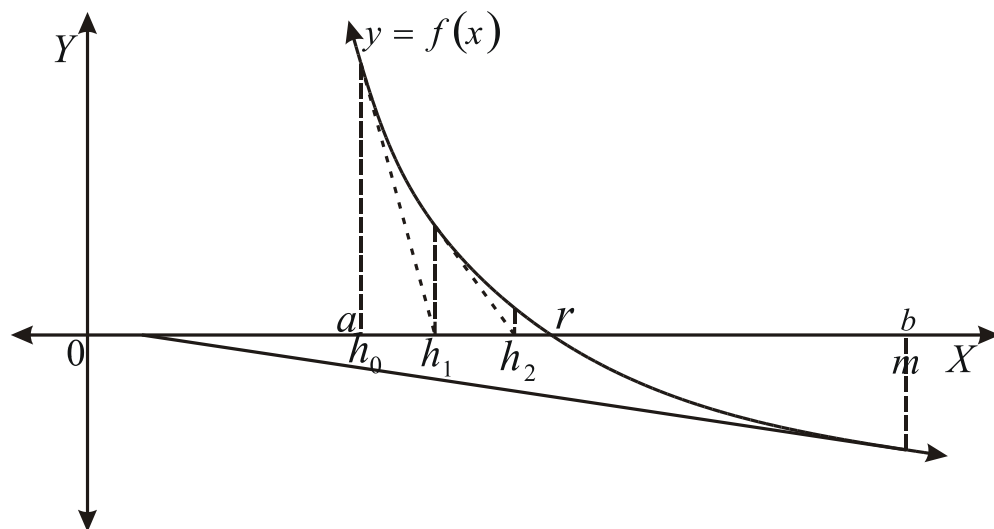
entonces nuestra primera aproximación es $h_0 = b$

Figura 10



$f''(x) < 0$ para todo $x \in [a, b]$, $f(a) < 0$, $f''(a) \cdot f(a) > 0$

entonces nuestra primera aproximación es $h_0 = a$



$f''(x) > 0$ para todo $x \in [a, b]$, $f(a) > 0$, $f''(a) \cdot f(a) > 0$

entonces nuestra primera aproximación es $h_0 = a$

El método de Newton-Raphson-Fourier consiste (teniendo en cuenta las condiciones antes mencionadas) en trazar una vertical por la primera aproximación $h_0 \in [a, b]$ —vea las figuras 9 y 10— hasta cortar la curva de ecuación $y = f(x)$; y por el punto de corte se traza una tangente a dicha curva hasta intersectar el eje X ; es justamente en ese punto de intersección donde se obtiene la abscisa que va servir de nueva aproximación a la raíz de $f(x)$. Este proceso se repite tantas veces como sea necesario hasta obtener una aproximación de la raíz, con la precisión deseada.

PRECAUCIONES DE FOURIER.

En las figuras 9 y 10, observamos que la reiteración del proceso, comenzando por el extremo h_0 , conduce a nuevas aproximaciones h_1, h_2, \dots , que se acercan a la raíz " r "; sin embargo, si hubiésemos elegido el extremo " m " como primera aproximación caeríamos, en general, fuera del intervalo (a, b) .

Ahora, es interesante conocer la técnica empleada para elegir la aproximación inicial $(x = h_0)$ que nos conduzca a una buena aproximación a la raíz real de la ecuación $y = f(x)$, o sea que nos garantice la convergencia del método. Este problema queda resuelto haciendo uso de las precauciones de Fourier que enunciaremos a continuación.

Atendiendo a las consideraciones hechas para $y = f(x)$, sabemos que ésta tiene un cero simple real en el intervalo (a, b) , entonces:

1. Si $f(b) \cdot f''(b) > 0$, debemos hacer $h_0 = b$ y $m = a$, o sea, la aproximación inicial es $h_0 = b$ (vea la figura 9).

2. Si $f(a) \cdot f''(a) > 0$, debemos hacer $h_0 = a$ y $m = b$, o sea, la aproximación inicial es $h_0 = a$ (vea la figura 10).

Si no se toman en cuenta estas precauciones, que demostraremos más adelante, puede ocurrir que la intersección de la recta tangente con el eje de las X caiga fuera del intervalo $[a, b]$, lo cual nos puede alejar de la raíz buscada.

De aquí se deduce la importancia de verificar que cada una de las aproximaciones h_i , para $i = 0, 1, 2, \dots$, estén dentro del intervalo $[a, b]$, y, en caso contrario, se interrumpirá el proceso iterativo, indicando que el método está trabajando incorrectamente, debido a que no se escogió el h_0 apropiado, atendiendo a las precauciones de Fourier.

Para hallar las aproximaciones h_1, h_2, \dots , a partir de la aproximación inicial $x = h_0$, se emplean las ecuaciones de las rectas tangentes a la curva $y = f(x)$. La primera recta tangente toca la curva en el punto $(h_0, f(h_0))$ y tiene una pendiente $f'(h_0)$. Así, la ecuación de la primera recta tangente está dada por:

$$f'(h_0) = \frac{y - f(h_0)}{x - h_0}, \quad \forall (x, y) \in \mathbb{R}^2$$

$$\Rightarrow y - f(h_0) = f'(h_0)(x - h_0)$$

Consideremos que la intersección de esta recta con el eje X es: $x = h_1$ y la determinaremos haciendo $y = 0 \wedge x = h_1$

$$\Rightarrow 0 - f(h_0) = f'(h_0)(h_1 - h_0)$$

despejando la incógnita h_1 se tiene:

$$h_1 = h_0 - \frac{f(h_0)}{f'(h_0)}, \quad f'(h_0) \neq 0$$

El valor de h_1 obtenido es el que utilizamos en la siguiente iteración, de modo que la nueva recta tangente a la curva $y = f(x)$, en el punto $(h_1, f(h_1))$ corta el eje X en el punto de abscisa desconocida, $x = h_2$, la cual está dada por:

$$h_2 = h_1 - \frac{f(h_1)}{f'(h_1)}, \quad f'(h_1) \neq 0$$

Continuando el proceso, llegamos a obtener la sucesión

$$\{h_{k+1}\}_{k=0}^{\infty},$$

determinada a partir de la fórmula de recurrencia:

$$h_{k+1} = h_k - \frac{f(h_k)}{f'(h_k)}, \quad f'(h_k) \neq 0 \quad (2.57)$$

para $k = 0, 1, \dots$.

DEMOSTRACIÓN DE LAS PRECAUCIONES DE FOURIER.

A partir de las consideraciones hechas a $y = f(x)$, vamos a probar las precauciones de Fourier. Éstas nos indican, como ya se mencionó, la forma

correcta de elegir la primera aproximación " h_0 " que conducirá a la convergencia del método.

Las precauciones de Fourier señalan que si $y = f(x)$ tiene una raíz real " r " en el intervalo (a, b) , entonces, la primera aproximación h_0 se debe elegir así:

1. Si $f(b) \cdot f''(b) > 0$, debemos hacer $h_0 = b$ y $m = a$.
2. Si $f(a) \cdot f''(a) > 0$, debemos hacer $h_0 = a$ y $m = b$.

DEMOSTRACIÓN.

Las consideraciones hechas a **(2.56)**:

$$y = f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0,$$

debemos tenerlas siempre presentes.

Así, sea " r " la raíz simple real de **(2.56)** en el intervalo (a, b) y $h_0 \in [a, b]$, la primera aproximación a dicha raíz. Sabemos, además, que $f(a) \neq 0 \wedge f(b) \neq 0$. Por conveniencia de notación, haremos $h = h_0$.

Desarrollando a **(2.56)** en el intervalo (h, r) ó (r, h) , —dependiendo de si $h < r$ ó $h > r$ respectivamente—, según la fórmula de Taylor (ver apéndice de cálculo, teorema número **12**), expresamos:

$$f(r) = \sum_{i=0}^k \frac{f^{(i)}(h)}{i!} (r-h)^i + \frac{f^{(k+1)}(\xi)}{(k+1)!} (r-h)^{k+1} \quad (2.58)$$

donde ξ está entre $h \wedge r$.

Observemos que, para $k = 1$, (2.58) se convierte en:

$$f(r) = f(h) + \frac{f'(h)}{1!} (r-h) + \frac{f''(\xi)}{2!} (r-h)^2$$

Como " r " es raíz de $f(x) \Rightarrow f(r) = 0$, es decir:

$$f(r) = f(h) + \frac{f'(h)}{1!} (r-h) + \frac{f''(\xi)}{2!} (r-h)^2 = 0$$

por hipótesis, $f'(h) \neq 0$, luego podemos dividir por este valor. Expresando convenientemente la expresión resultante, tenemos que:

$$\Rightarrow r = h - \frac{f(h)}{f'(h)} - \frac{f''(\xi)}{2! f'(h)} (r-h)^2 \quad (2.59)$$

Por hipótesis, $f''(x)$ tiene el mismo signo $\forall x \in [a, b]$, y el intervalo de extremos $h \wedge r$, ya sea (r, h) ó (h, r) , está incluido en $[a, b]$.

Como $f(a)$ y $f(b)$ son de signos opuestos, elegimos el extremo que hemos designado como h , para el cual:

$$f(h) \cdot f''(h) > 0$$

de modo que:

$$1. \quad \text{Si } f(b) \cdot f''(b) > 0, \Rightarrow h_0 = b \text{ (vea la figura número 9).}$$

$$2. \quad \text{Si } f(a) \cdot f''(a) > 0, \Rightarrow h_0 = a \text{ (vea la figura número 10).}$$

Si se verifica el caso número 1, tenemos que $h = b$ lo que nos indica que debemos aproximarnos a la raíz " r " de derecha a izquierda, trabajando en el intervalo $(r, h]$; en tanto que, si se cumple el caso número 2, entonces $h = a$, lo que indica que debemos aproximarnos a la raíz " r " de izquierda a derecha, trabajando en el intervalo $[h, r)$.

Vamos a demostrar que, en realidad, se da en ambos casos la respectiva aproximación a la raíz " r ". Como $y = f(x)$ tiene una única raíz $r \in (a, b)$, en caso que se verifique la desigualdad número 1, ésta se cumplirá $\forall x \in (r, h]$, $h = b$; en caso que se dé la desigualdad número 2, ésta se cumplirá $\forall x \in [h, r)$, $h = a$. Como $f'(h)$ no se anula en $[a, b]$ las razones:

$$\begin{cases} \frac{f(h)}{f'(h)} \\ \frac{f''(\xi)}{f'(h)} \end{cases} \quad (2.60)$$

poseen el mismo signo, ambas son positivas o ambas son negativas, ya que $f(h)$ tiene igual signo que $f''(\xi)$. Consideremos las situaciones que pueden darse.

CASO 1: SI LAS RAZONES (2.60) SON POSITIVAS

SIGNIFICA QUE:

$f(h) > 0 \wedge f'(h) > 0$, por tanto $f''(\xi) > 0$. Además, como $a < b \Rightarrow f(a) < f(b)$; por ser $f(x)$ creciente en $[a, b]$, pero $f(a)$ tiene el signo opuesto a $f(b) \Rightarrow f(b) > 0$ y por tanto $h = b$; estamos trabajando en el intervalo $(r, h]$, $h = b$ —ver, en la figura 9, su parte superior—.

Por otro lado, también puede ocurrir que: $f(h) < 0 \wedge f'(h) < 0$, por tanto, $f''(\xi) < 0$. Además, como $a < b \Rightarrow f(a) > f(b)$; por ser $f(x)$ decreciente en $[a, b]$, pero $f(a)$ tiene el signo opuesto a $f(b)$; luego, $\Rightarrow f(b) > 0$ y por tanto $h = b$; estamos trabajando en el intervalo $(r, h]$, $h = b$ —ver, en la figura 9, su parte inferior—.

Observemos que en ambos casos llegamos a la conclusión de que trabajaremos con el intervalo $(r, h]$, $h = b$.

Hechas estas consideraciones, podemos decir, en la ecuación (2.59), que sus dos últimos términos son positivos. Además, si suprimimos su último término y llamamos h_1 a la parte restante, tendremos que:

$$h_1 = h - \frac{f(h)}{f'(h)} \quad (2.61)$$

ahora, de (2.59) y (2.61), podemos decir que:

$$h_1 > r \quad (2.62)$$

Además, al suprimir el último término de **(2.61)**, podemos afirmar que:

$$h_1 < h \quad (2.63)$$

pues éste también es positivo —por las ecuaciones **(2.60)**—.

De **(2.62)** y **(2.63)**, podemos decir que $r < h_1 < h$, entonces $(r, h] \supset (r, h_1]$ y, a partir del nuevo subintervalo $(r, h_1]$, trabajamos con h_1 de igual modo como lo hicimos con h , en el intervalo $(r, h]$ y obtenemos el nuevo h_2 , que verificará la desigualdad $r < h_2 < h_1 < h$.

Reiterando este proceso, es evidente que nos acercamos a la raíz " r " a partir de cada nuevo h_{i+1} con $i = 0, 1, \dots, \wedge h = h_0$, de modo que:

$$r < \dots < h_{i+1} < \dots < h_2 < h_1 < h$$

CASO 2: SI LAS RAZONES (2.60) SON NEGATIVAS SIGNIFICA QUE:

$f(h) < 0 \wedge f'(h) > 0$, por tanto, $f''(\xi) < 0$. Además, como $a < b \Rightarrow f(a) < f(b)$; por ser $f(x)$ creciente en $[a, b]$, pero $f(a)$ tiene el signo opuesto a $f(b) \Rightarrow f(a) < 0$ y, por tanto, $h = a$; estamos trabajando en el intervalo $[h, r)$, $h = a$ (ver, en la figura **10**, su parte superior).

Por otro lado, también puede ocurrir que $f(h) > 0 \wedge f'(h) < 0$, por tanto, $f''(\xi) > 0$. Además, como $a < b \Rightarrow f(a) > f(b)$; por ser $f(x)$ decreciente en $[a, b]$, pero $f(a)$ tiene el signo opuesto a $f(b)$, luego, $\Rightarrow f(a) < 0$ y por tanto

$h = a$; estamos trabajando en el intervalo $[h, r)$, $h = a$ (ver, en la figura 10, su parte inferior).

Observemos que, en ambos casos, llegamos a la conclusión de que trabajaremos con el intervalo $[h, r)$, $h = a$.

Hechas estas consideraciones, podemos decir —en la ecuación (2.59)— que sus dos últimos terminos son negativos. Además, si suprimimos su último término y llamamos h_1 a su parte restante, tendremos que:

$$h_1 = h - \frac{f(h)}{f'(h)} \quad (2.64)$$

ahora, de (2.59) y (2.64), podemos decir que:

$$h_1 < r \quad (2.65)$$

Además, al suprimir el último término de (2.64), podemos afirmar que:

$$h_1 > h \quad (2.66)$$

pues éste también es negativo —por las ecuaciones (2.60)—.

De (2.65) y (2.66), podemos decir que $h < h_1 < r$, entonces $[h, r) \supset [h_1, r)$. Podemos, por tanto, tomar el intervalo $[h_1, r)$, como se hizo con $[h, r)$ y obtenemos intervalo $[h_2, r)$ contenido en $[h_1, r)$, donde $h_1 < h_2$, de modo que se verificará la desigualdad $h < h_1 < h_2 < r$.

Así, nos acercamos nuevamente a la raíz " r " de $f(x)$ a partir de cada nuevo h_{i+1} con $i = 0, 1, \dots$, $\wedge h = h_0$, de modo que:

$$h < h_1 < h_2 < \dots < h_{i+1} < \dots < r$$

Debemos observar que de ambos casos hemos construido dos sucesiones monótonas acotadas; la primera:

$$h_0 = b, h_1, h_2, \dots, h_{i+1}, \dots h$$

es decreciente, y, la segunda:

$$h_0 = a, h_1, h_2, \dots, h_{i+1}, \dots$$

es creciente. La teoría de las sucesiones nos garantiza que ambas sucesiones tienen un límite (ver apéndice de cálculo, teorema número **13**).

Supongamos que dicho límite es k , o sea:

$$\lim_{i \rightarrow \infty} h_i = k$$

probaremos que dicho límite es la raíz que buscamos en la ecuación **(2.56)**.

Consideremos la diferencia absoluta entre dos valores consecutivos de una de esas sucesiones, es decir:

$$|e_{i+1}| = |h_{i+1} - h_i|$$

La convergencia de ambas sucesiones nos garantiza que:

$$\lim_{i \rightarrow \infty} |e_{i+1}| = \lim_{i \rightarrow \infty} |h_{i+1} - h_i| = 0 \quad (2.67)$$

Ahora, por definición, para ambas sucesiones tenemos que:

$$\begin{aligned} \lim_{i \rightarrow \infty} |h_{i+1} - h_i| &= \lim_{i \rightarrow \infty} \left| h_i - \frac{f(h_i)}{f'(h_i)} - h_i \right| \\ &= \lim_{i \rightarrow \infty} \left| -\frac{f(h_i)}{f'(h_i)} \right| \end{aligned}$$

y al aplicar las propiedades del valor absoluto, tenemos que:

$$= \lim_{i \rightarrow \infty} \frac{f(h_i)}{f'(h_i)}, \text{ con } \frac{f(h_i)}{f'(h_i)} > 0$$

Por ser f continua, $\forall x \in I$, en particular para $x = h_i$ (ver apéndice de cálculo, teorema número 5) y por los teoremas acerca del límite de funciones (ver apéndice de cálculo, teorema número 6):

$$\begin{aligned} \lim_{i \rightarrow \infty} |h_{i+1} - h_i| &= \frac{f\left(\lim_{i \rightarrow \infty} h_i\right)}{f'\left(\lim_{i \rightarrow \infty} h_i\right)} \\ &= \frac{f(k)}{f'(k)} \end{aligned}$$

Ahora, por (2.67), podemos decir que:

$$\frac{f(k)}{f'(k)} = 0$$

y como por hipótesis, $\forall x \in [a, b], f(x) \neq 0$ —en particular para $x = k \in [a, b]$ —, podemos decir que:

$$\Rightarrow f(k) = 0$$

lo que implica que $k = r$, pues " r " es, por hipótesis, el único valor para el que $f(x) = 0$ en el intervalo (a, b) , de modo que se garantiza también que ambas sucesiones convergen al mismo límite.

Hasta aquí, hemos demostrado la validez de las precauciones de Fourier, y que ambas sucesiones:

$$h = h_0, h_1, h_2, \dots, h_{i+1}, \dots, h$$

son monótonas, acotadas (una es creciente y la otra decreciente), y tienen como límite a " r ".

La fórmula iterativa empleada en este método:

$$h_{i+1} = h_i - \frac{f(h_i)}{f'(h_i)}, \quad f'(h_i) \neq 0 \quad (2.68)$$

para $i = 0, 1, \dots$; nos señala que cada nueva aproximación de la raíz " r " de $f(x)$ se obtiene de operaciones hechas a partir de la interacción inmediatamente anterior. Este procedimiento se repetirá las veces necesarias hasta obtener una buena aproximación, lo que ocurre cuando dos valores

consecutivos de la raíz aproximada difieren en una cantidad menor que cierto valor épsilon (ε) preestablecido, que controla el error permisible en la raíz.

Demostraremos que este error cometido en la última etapa de aproximación es menor que:

$$\frac{|\Delta f''(x)|(m - h_i)^2}{2|f'(h_i)|}$$

siendo $|\Delta f''(x)|$ una cota superior de $|f''(x)|$ y " m ", el valor desechado como aproximación inicial, en base a las precauciones de Fourier.

DEMOSTRACIÓN.

Consideremos la diferencia absoluta entre el valor verdadero de la raíz que buscamos " r " y la última aproximación que calculamos de ella, h_{i+1} , es decir, queremos calcular a:

$$|e_{i+1}| = |r - h_{i+1}| \quad (2.69)$$

Reemplazando el valor de r dado por (2.59) y el de h_{i+1} por el dado en (2.68), tenemos que (2.69), al realizar las simplificaciones correspondientes, se puede expresar como:

$$|e_{i+1}| = \left| -\frac{f''(\xi)}{2f'(h_i)}(r - h_i)^2 \right|$$

debemos observar que desconocemos el signo de la expresión encerrada en el valor absoluto. sin embargo, al aplicar las propiedades del valor absoluto, podemos decir que:

$$\Rightarrow |e_{i+1}| = \frac{f''(\xi)}{2f'(h_i)}(r-h_i)^2, \text{ con } \frac{f''(\xi)}{2f'(h_i)} > 0$$

obsérvese que no sabemos si la sucesión esta acotada en el intervalo $[h, r)$ o $(r, h]$, de modo que se nos presentan dos casos posibles:

CASO 1: LOS VALORES QUE TOMA LA SUCESIÓN ESTÁN UBICADOS EN EL INTERVALO $[h, r)$.

Si este es el caso, hagamos la mayoración:

$$|\Delta f''(x)| \geq |f''(\xi)|, \forall x \in [h, r)$$

donde $|\Delta f''(x)|$ es una cota superior de $|f''(x)|$, $\forall x \in [h, r)$, en particular para $x = \xi$ —ya que según el desarrollo de Taylor, $\xi \in [h, r)$ —, de modo que obtenemos:

$$\Rightarrow |e_{i+1}| \leq \frac{|\Delta f''(x)|}{2|f'(h_i)|}(r-h_i)^2$$

Por otro lado, como desconocemos el valor de " r ", en su lugar colocamos el extremo " b " del intervalo (a, b) de modo que nos aseguramos así

que la función f cambia de signo en los extremos del intervalo $[h_i, b]$, pues esto garantiza que: $[h_i, r) \subset [h_i, b)$, $\Rightarrow r - h_i < b - h_i$.

$$\Rightarrow |e_{i+1}| < \frac{|\Delta f''(x)|}{2|f'(h_i)|} (r - h_i)^2$$

lo que concluye la demostración para el caso 1.

CASO 2: LOS VALORES QUE TOMA LA SUCESIÓN ESTAN UBICADOS EN EL INTERVALO $(r, h]$.

Si este es el caso, hagamos la mayoración:

$$|\Delta f''(x)| \geq |f''(\xi)|, \quad \forall x \in (r, h]$$

donde $|\Delta f''(x)|$ es una cota superior de $|f''(x)|$, $\forall x \in (r, h]$, en particular para $x = \xi$ —ya que según el desarrollo de Taylor, $\xi \in (r, h]$ —, de modo que obtenemos:

$$\Rightarrow |e_{i+1}| \leq \frac{|\Delta f''(x)|}{2|f'(h_i)|} (r - h_i)^2$$

Por otro lado, como desconocemos el valor de " r " en su lugar colocamos el extremo " a " del intervalo (a, b) de modo que nos aseguramos así que la función f cambia de signo en los extremos del intervalo $[a, h_i]$, pues esto garantiza que: $(r, h_i] \subset (a, h_i]$, $\Rightarrow h_i - r < h_i - a$.

$$\Rightarrow |e_{i+1}| < \frac{|\Delta f''(x)|}{2|f'(h_i)|} (r - h_i)^2$$

lo que concluye la demostración para el caso **2**.

Debe observarse que en el caso **1**, obtuvimos $m=b$; para la aproximación inicial $h_0=a$; y en el caso **2**, obtuvimos $m=a$ para $h_0=b$; siendo m , en ambos casos, el valor no escogido como valor inicial, atendiendo a las precauciones de Fourier, lo que finaliza la demostración.

En base a los teoremas previamente demostrados, podemos establecer el método de Newton-Raphson-Fourier en los siguientes términos:

PROCEDIMIENTO DE NETTON-RAPHSON-FOURIER.

Sea f una función real de variable real. El empleo del método de Newton-Raphson-Fourier —encaminado a hallar las raíces de la ecuación $f(x)=0$ —, requiere que dichas raíces hayan sido separadas previamente, de modo que en cada subintervalo (a,b) exista una única raíz; que $f(a) \neq 0 \wedge f(b) \neq 0$; además $f'(x) \wedge f''(x)$ existen, pero no se anulan en el intervalo (a,b) .

Considerando las precauciones de Fourier, elegimos la aproximación inicial $x_0 \in (a,b)$ a la raíz " r " de $f(x)=0$ en dicho intervalo. El método procede así:

Sea " r " una raíz de la ecuación $f(x)=0$ en el intervalo (a,b) , donde la función f satisface las condiciones anteriormente establecidas. No conocemos

" r " pero con la estimación " x_0 ", obtenida a partir de las precauciones de Fourier, hallamos valores sucesivos de " x_i ", para $i = 1, 2, \dots$, a través de la aplicación de la fórmula **(2,57)**, siendo cada nueva aproximación mejor que las anteriores, generándose así, una sucesión de números reales:

$$\{x_i\}_{i=0}^{\infty}$$

que converge a " r " con la requerida precisión.

2.2.3.5. MÉTODOS ALTERNATIVOS.

Existe toda una gama de métodos para aproximar raíces reales, ya sea de funciones cualesquiera o de un tipo de función específico.

Muchos se derivan de los métodos que hemos tratado hasta este momento -por ejemplo. de los métodos de Iteraciones sucesivas y Newton-Raphson, se derivan "familias" de métodos relacionados a ellos

Algunos presentan ventajas sobre aspectos en los que los métodos tratados son deficientes; sin embargo, suelen presentar un procedimiento de aplicación₁ a su vez, mas difícil.

A continuación, haremos mención de algunos métodos alternativos a los que hemos desarrollado en nuestra exposición.

No pretendemos realizar su desarrollo completo, como los previamente tratados, ni sus respectivas demostraciones. Sólo resaltaremos algunos aspectos prácticos de los métodos alternativos —ciertas ventajas y desventajas que presenta su aplicación con respecto a los métodos desarrollados en nuestra

exposición, por ejemplo—, con el objeto de destacar que existen alternativas a las que hemos desarrollado.

MÉTODO DE APROXIMACIONES SUCESIVAS MODIFICADO.

El método de aproximaciones sucesivas modificado involucra las mismas hipótesis que el método que lo origina, en lo referente al intervalo I ; sin embargo, es más flexible que éste con respecto a los valores de $F'(x)$ en dicho intervalo.

Dicho método busca acelerar la convergencia de la sucesión en los casos en que el método de aproximaciones sucesivas converja, y contempla situaciones que permiten corregir ciertos casos en que la sucesión es divergente.

Este método requiere de dos valores iniciales —llamémoslos $x_0 \wedge x_1$ —, donde x_0 se toma en base a los teoremas de convergencia del método de aproximaciones sucesivas previamente tratado, y x_1 , puede ser generado por dicho método.

El método en mención consiste en aplicar, para cada iteración, la fórmula:

$$x_{i+1} = x_i + \alpha(F(x_i) - x_i)$$

donde:

$$\alpha = \frac{1}{1 - F'(\xi)}$$

y:

$$F'(\xi) = \frac{F(x_i) - x_i}{x_i - x_{i-1}}$$

Uno de los logros más importantes de este método consiste en que: cuando $F'(x) > 1$ en el intervalo I , se autocorrigien los valores de la sucesión, de modo que se da la convergencia. Además, el método de Newton-Raphson puede deducirse fácilmente de él.

MÉTODO DE NEWTON DE SEGUNDO ORDEN.

Este método se distingue del de Newton-Raphson, previamente tratado, en que considera más términos del desarrollo de Taylor de la función f , y está sujeto, por tanto, a restricciones y condiciones similares —aunque más rigurosas— a las que se le imponen a Newton-Raphson, en lo referente al comportamiento de la función f en el intervalo (a, b) , en función de sus derivadas, así como en la escogencia del valor inicial h_0 .

El método en mención está definido por las fórmulas de recurrencia:

$$x_{i+1} = x_i + \Delta x_i \quad (2.70)$$

con:

$$\frac{1}{\Delta x_i} = -\frac{f'(x_i)}{f(x_i)} + \frac{f''(x_i)}{2f'(x_i)} \quad (2.71)$$

donde **(2.70)** y **(2.71)** definen al método en mención, que aproximará las raíces de la ecuación $f(x) = 0$.

El método de Newton de segundo orden parece mejor que el de Newton-Raphson, en cuanto que considera mayor número de términos en el desarrollo de la función f a través de la fórmula de Taylor; lo que permite, en algunos casos, obtener más rápidamente la raíz buscada, que si se empleara el método de Newton-Raphson.

Sin embargo, podemos observar también que las fórmulas (2.70) y (2.71) son más complicadas que la de Newton-Raphson y requerirán, por tanto, de mayores consideraciones para asegurar su convergencia.

MÉTODO DE VON MISES.

El método de Newton-Raphson requiere el cálculo de $f'(x_i)$ para sustituirlo en el denominador de su fórmula de recurrencia; lo que indica, geométricamente, que pasamos de una aproximación a la siguiente a través de la recta tangente a la curva $y = f(x)$, trazada en el punto correspondiente a la aproximación precedente. Sin embargo, esto trae dificultades cuando se está en puntos donde el valor de $f'(x)$ es cercano a cero.

Para resolver este problema, Von Mises propuso sustituir $f'(x_i)$ por $f'(x_0)$, lo que equivale geométricamente a pasar de una aproximación a la siguiente por medio de paralelas a la primera tangente.

En consecuencia, el método de Von Mises para resolver la ecuación $f(x) = 0$, queda definido por la siguiente fórmula iterativa:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_0)}, \quad f'(x_0) \neq 0$$

Obviamente este método es más lento que el de Newton-Raphson, pero resuelve, en gran medida, el problema indicado.

MÉTODO DE LA SECANTE.

El método de Newton-Raphson-Fourier es una técnica muy satisfactoria, pero puede presentar dificultades en su aplicación, especialmente cuando se tratan funciones trascendentes; pues la necesidad de evaluar la derivada de f —en cada aproximación—, conlleva, frecuentemente, a un cálculo más complicado y que requiere más operaciones aritméticas, que las necesarias para evaluar a f .

Con el objeto de evitar esta dificultad, se ha introducido al método de Newton-Raphson una pequeña variación, consistente en reemplazar la derivada de f , dada por definición como:

$$f'(h_i) = \lim_{x \rightarrow h_i} \frac{f(x) - f(h_i)}{x - h_i}$$

por su aproximación, haciendo $x = h_{i-1}$:

$$f'(h_i) \approx \frac{f(h_i) - f(h_{i-1})}{h_i - h_{i-1}} \quad (2.72)$$

Y luego, este valor de $f'(h_i)$ se reemplaza en la fórmula común de Newton-Raphson, dada por (2.68), de modo que reduciendo (2.68) y (2.72) a una sola fórmula, tenemos que:

$$h_{i+1} = h_i - f(h_i) \left(\frac{h_i - h_{i-1}}{f(h_i) - f(h_{i-1})} \right) \quad (2.73)$$

Luego, el nuevo método queda definido al aplicar, en cada iteración, la fórmula **(2.73)**. Esta fórmula de recurrencia es conocida como la empleada en el método de la Secante.

MÉTODO DE LAGUERRE.

Practicamente, todos los métodos vistos en el desarrollo de este trabajo requieren de buenos valores Iniciales que los pongan en marcha, para asegurar la convergencia. Esta situación, obviamente, no se puede resolver con facilidad en todos los casos.

El método de Laguerre suple, en gran medida esta dificultad, con un aumento relativamente pequeño en el número de operaciones aritméticas a realizar.

Dicho método converge a la raíz buscada, en forma más veloz que el mismo método de Newton-Raphson-Fourier, y, prácticamente, se puede emplear como valor inicial una cantidad cualquiera, aun cuando se aproximen raíces múltiples.

Este método ha sido desarrollado recientemente y sus consideraciones teóricas, por tanto, no están totalmente definidas; sin embargo, toda la evidencia recopilada hasta el momento por los investigadores apunta a que la falta de convergencia, en él, es muy poco usual.

El método en cuestión consiste en la aplicación de las fórmulas de recurrencia:

$$H(x_i) = (n-1)\{(n-1)[f'(x_i)]^2 - nf(x_i)f''(x_i)\}$$

$$x_{i+1} = x_i - \frac{nf(x_i)}{f'(x_i) + \operatorname{sgn}[f'(x_i)]\sqrt{H(x_i)}}$$

donde, obviamente, n es el grado del polinomio al que se le aproximan las raíces.

El método, tal como lo hemos tratado, se debe aplicar sobre polinomios que posean sólo raíces reales.

2.3. COMPORTAMIENTO PRÁCTICO DE LOS MÉTODOS ITERATIVOS AL CALCULAR RAÍCES.

Recapitulando lo realizado hasta este punto, hemos desarrollado toda una gama de métodos iterativos aproximados, con el objeto de resolver el problema de determinar las raíces reales de un polinomio real dado.

Con base a la experiencia que obtuvimos en el capítulo 1, es conveniente, entonces, tratar de responder a algunas preguntas: ¿Los métodos iterativos son, en la práctica, la solución para nuestro problema? Y si lo son: ¿Son completamente efectivos dichos métodos? ¿Pueden ser aplicados en forma inmediata y eficiente para resolver un problema específico?

La primera pregunta podemos responderla con seguridad y categóricamente con un si. En efecto, en las ramas de la matemática aplicada, la resolución de ecuaciones debe realizarse en función de obtener una buena respuesta aproximada y no tanto en función de una respuesta exacta.

Así, observemos que sólo al calcular la raíz cuadrada dentro de la ecuación cuadrática —salvo contadas excepciones— lo hacemos en forma aproximada y sin poder conocer qué error cometemos en su cálculo. Dicho error, se transmite en consecuencia al resto de la expresión. de modo que no podemos determinar, en realidad, el error que estamos cometiendo en el cálculo.

En cambio, los métodos iterativos permiten realizar la corrección de la respuesta que originan, después que sean aplicados el número debido de veces; de modo que se alcanza el grado de precisión deseado.

Además, a manera de ejemplo, para un ingeniero no tiene sentido obtener una respuesta exacta al cálculo de las raíces de una ecuación, dentro de un problema de aplicación, si los instrumentos con que aplicará posteriormente los resultados de dicha solución no poseen todo el número de cifras significativas que posee la respuesta obtenida,

Por otro lado, la segunda pregunta posee una respuesta más difícil de dar. Esto se debe a que los desarrollos de los métodos fueron realizados en forma matemática, de modo que los números involucrados en las sucesiones a que nos referimos en esos desarrollos, poseen un número infinito de cifras significativas.

En tanto, ningún ser humano es capaz de manejar dichas cifras mucho menos los aparatos que el hombre ha creado de modo que necesariamente se trabaja con valores truncados a un número fijo de cifras significativas.

Sin embargo, nuevamente la característica de autocorrección de los métodos iterativos hará que el proceso alcance el grado de precisión deseado, siempre que se apliquen el número debido de veces.

Ahora, hay que resaltar que pueden construirse ecuaciones para las que un método cualquiera sea divergente, como ocurre cuando una ecuación posee raíces múltiples, complejas o de módulos muy cercanos entre si.

Esta situación justifica, en parte, la existencia de la gran cantidad de métodos —de los que sólo hemos visto algunos— que se han desarrollado para resolver este problema.

Sin embargo, la utilización juiciosa de todos —incluyendo los muchos que hemos dejado hasta de mencionar—, en conjunto, garantizará, en la mayoría de los casos, que se obtendrán soluciones satisfactorias para las aplicaciones que requieran resolver este tipo de problema.

Finalmente, la tercera pregunta posee una respuesta ambigua, ya que: si disponemos de todo el tiempo necesario, además de una gran habilidad en el cálculo aritmético, podremos obtener las soluciones que busquemos, la mayoría de las veces, al aplicar los métodos iterativos.

Sin embargo, debemos sopesar la pérdida de tiempo que supondría, por ejemplo, la aplicación de un algoritmo relativamente sencillo, como el procedimiento de bisección de Bolzano, si deseamos obtener nuestra respuesta con precisión en el Orden de 10^{-12} ; y la aplicación del criterio de la cota al error cometido en la última iteración, para dicho método —es decir la ecuación **(2.15)**—, nos indica que debemos iterar el método, por lo menos 100 veces.

Es evidente, entonces, que debemos buscar mecanismos que nos permitan agilizar el cálculo aritmético, para hacer de dichos procedimientos una herramienta verdaderamente eficiente para la resolución del problema en mención.

**3. PROGRAMACIÓN Y COMPARACIÓN DE LOS
MÉTODOS ITERATIVOS PARA ENCONTRAR
LAS RAÍCES DE ECUACIONES ALGEBRAICAS.**

3.1. OBSERVACIONES PRELIMINARES.

Lo tratado al final del capítulo anterior deja claramente establecido que es necesario desarrollar algún tipo de procedimiento que nos permita agilizar el cálculo de las raíces de una ecuación algebraica de cualquier grado, al aplicar los esquemas iterativos previamente tratados. De lo contrario, éstos serán tan inoperantes en la práctica, como la obtención de las raíces de una ecuación cúbica o cuártica mediante sus respectivas fórmulas.

La agilización de cálculo a que nos referimos tiene un origen de carácter teórico: el deseo de resolver en forma eficiente el problema que hemos desarrollado. También surge de la necesidad de aplicar estos métodos a la solución de las situaciones prácticas que se presentan dentro de las otras ramas de las matemáticas que requieren de estos conceptos (por ejemplo, en la solución de ecuaciones diferenciales lineales).

Por otro lado, los problemas dentro de la ingeniería exigen, por lo general, de una gran cantidad de esfuerzo y tiempo aritmético, que podrían utilizarse en algo de más provecho.

Por fortuna, la creación de las computadoras electrónicas automáticas revolucionó estas situaciones.

La asombrosa rapidez que poseen para realizar operaciones aritméticas y lógicas, aunadas a su enorme capacidad de manejo de información, las faculta para ser catalogadas como las herramientas más adecuadas que posee el hombre, en la actualidad, para resolver problemas que involucran procesos e información en cantidades voluminosas.

Por otro lado, dichas máquinas no son seres pensantes, de modo que la eficiencia de su trabajo depende directamente de la forma en que se les suministra la información. Además, ellas no son capaces de manejar los números y las operaciones en la forma en que acostumbramos usarlos en la matemática clásica.

Estas situaciones nos indican que existen factores que posibilitan y restringen la aplicación de las computadoras a la solución del problema que nos concierne.

Establecemos que, en lo sucesivo, nuestra discusión supondrá que el lector tiene conocimientos elementales acerca de la programación de computadoras en el lenguaje BASIC, específicamente , y de teoría de errores aplicada a la computación.

Estos temas no se han incluido en el desarrollo de nuestra obra, ya que, por un lado, sólo nos interesan como herramienta para proseguir con el desarrollo del tema que nos atañe, y por otro lado, ni siquiera los incluimos en apéndices propios porque son temas muy abarcadores y que harían crecer en forma exorbitante el tamaño de nuestra obra. Sin embargo, dado que la teoría de error es un aspecto más matemático que otra cosa, se le realizará una breve mención en un apartado posterior de este capítulo.

El lector que presente deficiencias en lo que respecta a la programación en el lenguaje BASIC, puede consultar las obras de **JOYANES AGUILAR [15]**, **GOTTFRIED [7]** y **ADAMIS [1]**, que ofrecen una exposición completa de este tema. Por otro lado, las referencias de la teoría del error se realizarán más tarde, cuando tratemos algunos aspectos de ella.

Retomando, pues, el hilo de nuestra exposición, las situaciones que hemos tratado previamente nos indican que existen factores que posibilitan y restringen la aplicación de las computadoras a la solución del problema que nos atañe. Dichos factores los analizaremos, de manera escueta, a continuación.

3.2. VENTAJAS Y RESTRICCIONES QUE PRESENTA LA APLICACIÓN DEL USO DE COMPUTADORAS EN LA SOLUCIÓN DE NUESTRO PROBLEMA.

Al considerar las restricciones que existen al aplicar, en el problema que nos compete, el uso de las computadoras, tenemos que: ellas representan a un número cualquiera con una cantidad máxima de cifras significativas, y no con el número infinito de cifras significativas que se acostumbra.

Las operaciones que involucran procesos infinitos o infinitesimales en general como el valor a que converge la suma de una serie, el valor a que converge una sucesión, evaluar la derivada de una función en un punto dado, calcular el valor de una integral definida, etc. , se deben aproximar a través del empleo de un número finito de aplicaciones de las operaciones particulares necesarias, lo que conlleva la existencia de cierto grado de error en la respuesta.

Además, por sí solas, no resuelven ningún tipo de problema, únicamente son capaces de seguir procedimientos computacionales cuidadosamente establecidos.

Para que sean capaces de resolver un problema cualquiera, se debe llevar a cabo todo un proceso previo de preparación del problema mucho más cuidadoso que si se resolviera manualmente , que se puede caracterizar por sus rasgos generales, como:

1. Identificar y formular claramente el problema y los objetivos que se deben alcanzar para considerar resuelto dicho problema, de modo que éste quede planteado sin ningún tipo de ambigüedades.

2. Describir matemáticamente el problema, en forma tan precisa como sea posible, ya que las computadoras sólo pueden realizar operaciones aritméticas y lógicas. Dicha descripción se puede realizar en términos de la economía del tiempo de máquina requerido para procesar el problema, o en dejar claramente establecido el procedimiento general que se analiza.
3. Realizar el análisis numérico del esquema matemático que pretende describir el problema en mención, incluyendo las situaciones que pueden presentar errores al realizar los cálculos aritméticos, por efecto de la naturaleza finita de los números que maneja la máquina (análisis numérico es la rama aplicada de las matemáticas, que pretende plantear métodos constructivos algorítmicos para resolver problemas matemáticos, a través del empleo de operaciones aritméticas).
4. Diseñar un método de cálculo o algoritmo conjunto de procedimientos completo y que no posee ambigüedades, y conduce a la solución de un problema matemático cualquiera particular que establezca los procedimientos que la computadora debe realizar para obtener la solución del planteamiento matemático del problema. Tales procedimientos deben comprender, entre otras cosas: establecer claramente el dominio de sus posibles datos iniciales, todas las posibles fuentes de error que pueden afectar los resultados, proporcionar medios adecuados para verificar la exactitud de la solución obtenida y dejar abierta la opción de que se realicen correcciones, en caso que no se obtenga la respuesta buscada.
5. Traducir el método de cálculo a un lenguaje de computadora acorde con las necesidades del usuario, convirtiéndose en un conjunto de instrucciones detalladas y claras para el computador mejor conocido como programa .
6. Realizar pruebas experimentales del programa desarrollado, para detectar posibles errores en la realización de los pasos previos, llevando a cabo las

correcciones que sean necesarias, en función de que la información generada sea útil.

7. Ejecutar el planteamiento matemático del problema en mención al aplicar el programa previamente desarrollado a los datos del problema específico que se quiere resolver (en la actualidad, las computadoras sólo pueden operar con cantidades aritméticas, sin embargo, en el futuro se espera lograr que ellas puedan operar con cantidades algebraicas).
8. Interpretar los resultados que genera la respuesta de la computadora, de modo que se ubiquen dentro de los términos del problema originalmente planteado.

Podemos ver que la computadora en ningún momento releva a su usuario de la responsabilidad de plantear cuidadosamente su trabajo, ni de poseer un dominio amplio y profundo acerca de la parte matemática que involucra el tema de su problema.

Por otro lado, debemos destacar como ventajas que: cuando los requerimientos previamente establecidos son cumplidos a cabalidad por el usuario de la computadora, ésta es enormemente más rápida y precisa que cualquier ser humano. De este modo la computadora brinda una ayuda muy valiosa al permitirnos explorar las distintas alternativas que presenta un problema, en un tiempo relativamente corto.

En consecuencia, podemos percatarnos de que las computadoras nos pueden brindar un sólido apoyo en la solución de nuestro problema.

3.3. ALGUNOS ASPECTOS IMPORTANTES ACERCA DE LA TEORÍA DEL ERROR.

Debemos partir señalando que las respuestas que genera la computadora, por sí solas, no son siempre precisas y exactas, ya que existe todo un proceso que verificar, cuando resolvemos un problema cualquiera empleando computadoras, y en cada una de las partes de este proceso estamos supuestos a cometer equivocaciones.

Primeramente, estamos expuestos a errores de aproximación a la situación física, es decir, que el modelo planteado para la situación física dada no sea exacto. Este tipo de error no se da dentro de un planteamiento matemático, ya que los datos iniciales se suponen exactos normalmente .

Por otro lado, existen errores propios de los datos, es decir, que la información experimental que surge del planteamiento del problema contiene un cierto grado de incertidumbre en su magnitud, como producto de los procesos de medición aplicados. Nuevamente, este tipo de error no nos incumbe.

Además, existe el error de confusión de datos típico para cualquier persona que ha tenido oportunidad de realizar de seguido un cierto número de operaciones aritméticas. Este tipo de error puede ocurrir dentro de nuestro campo, pero, por suerte, la computadora nos permite controlarlo casi por completo.

Por otro lado, existe el error por truncamiento, que surge cuando aproximamos un proceso infinito (cuando sumamos una serie, por ejemplo) a través del empleo de sólo un número finito de los términos que involucra dicho proceso. Este tipo de error puede darse dentro de nuestro campo y un ejemplo del mismo se hace patente al desarrollar la teoría de los métodos iterativos, cuando establecíamos cotas superiores

de los errores en que se incurría en la i -ésima iteración, con respecto al valor verdadero que deseábamos aproximar.

Finalmente, el error más peligroso al que podemos enfrentarnos en nuestro trabajo es el error por redondeo o aproximación. Éste surge porque la computadora sólo puede manejar cantidades que poseen un número finito de cifras significativas, y todo número que posea más cifras significativas que las que puede manejar la máquina es truncado a ese número de cifras significativas.

Este fenómeno se da en cada operación aritmética que realiza la máquina, de modo que los resultados que se van obteniendo adolecen de la propagación del error de las operaciones previamente realizadas, es decir, la acción conjunta de los errores por redondeo de la operación anterior y los de la operación que se realiza, hace que el error acumulado aumente o disminuya.

Es claro, entonces, que al darse esta situación, a medida que aumenta el número de operaciones aritméticas que involucra un método de cálculo dado, aumentará el efecto del error de redondeo, de modo que puede llegar a ocurrir que los resultados que surgen de la aplicación del algoritmo, sean totalmente erróneos.

Los algoritmos que presentan este comportamiento explosivo en el error de redondeo, se denominan numéricamente inestables ejemplo de este tipo de algoritmo es la regla de los rombos ; por otro lado, los que no presentan este tipo de comportamiento, se denominan numéricamente estables.

Cuando se presenta este tipo de error al aplicar un método de cálculo a un problema en particular se puede presentar en un método cualquiera , sólo puede controlarse a través de complejos procesos. A continuación enunciaremos dos técnicas aplicables. No las desarrollaremos, ya que casi todos los algoritmos que

desarrollamos involucran, en la práctica, pocas operaciones aritméticas en cada iteración, de modo que este tipo de error no es apreciable en ellos.

1. El primer criterio involucra representar a cada número como un intervalo, representado por los valores máximo y mínimo que se pueden asumir. Este método evidentemente complica cualquier tipo de cálculo que involucre dicho número. Para más detalles al respecto, el lector interesado se puede referir a **HAMMING [11]**, página **166** y siguientes.
2. El otro criterio consiste en tratar de resolver el problema a través de la aplicación de un enfoque estadístico a la presencia de errores de redondeo en un método de cálculo dado. Esta técnica involucra considerar los errores de redondeo locales, en el método de cálculo, como un conjunto de variables aleatorias normalmente distribuidas, y luego aplicando técnicas estadísticas, se obtienen los parámetros de la distribución de variables, y, en función de ellos, se estiman los valores verdaderos que debió generar el algoritmo. Esta técnica involucra un análisis particular para cada método de cálculo, así como un conjunto de operaciones que se añaden al mismo. Sin embargo, los resultados que se han obtenido hasta el momento se presentan en completo acuerdo con la realidad empírica; pero su planteamiento práctico a cada método involucra la consideración de casos particulares, lo que indica que es una técnica que debe ser investigada más a fondo. Los interesados en los detalles de esta técnica, pueden referir a **HENRICI [12]**, página **330** y siguientes; **RALSTON [26]**, página **31** y siguientes, y **HAMMING [11]**, página **170** y siguientes.

En última instancia, sin información preliminar a mano, ningún método de cálculo debe ser considerado como libre de este flagelo, de modo que siempre se debe tratar de verificar los resultados obtenidos, empleando un medio alternativo.

Además existen algunas reglas empíricas acerca de cómo tratar de controlarlo, sin embargo, éstas no resuelven el problema por completo, y por eso no las enunciaremos. El lector interesado en conocerlas, puede dirigirse a **MCCRACKEN y DORN [23]**, página 79.

3.4. ALGUNAS LIMITACIONES QUE SE PRESENTAN AL TRATAR DE DETERMINAR EL MEJOR MÉTODO.

Al tener en cuenta tantos métodos como los que hemos desarrollado, surge inmediatamente la duda de ¿cuál de ellos será el mejor? Esta es una pregunta difícil de responder, ya que hay muchos criterios en función de los cuales podemos contestarla. A continuación, trataremos de dar algunas luces a respecto.

Primeramente, dejaremos establecido claramente que no hay ningún método que sea superior en un cien por ciento a los otros. Esta situación es la que justifica, en primer término, la existencia de tantos métodos. Cada método presenta situaciones para las que se presenta como muy eficiente, mientras que en otras, puede fallar estrepitosamente.

Así, a manera de ejemplo, la teoría y la experiencia nos indican que el método de Newton-Raphson-Fourier, cuando es aplicado a una función que cumple las condiciones de Fourier en un intervalo donde está ubicado el valor inicial que se usa, convergerá de manera rápida y eficiente a la raíz buscada. Sin embargo, la presencia de raíces cercanas, complejas o múltiples entorpece el desempeño de este método.

En cambio, un método como el de bisección de Bolzano, convergerá a una raíz a un ritmo mucho menor que Newton-Raphson-Fourier, y, sin embargo, la presencia de raíces cercanas, complejas o múltiples prácticamente no afecta su desempeño.

Así, es evidente que debemos considerar, para cada método, qué condiciones favorecen o entorpecen su aplicación. Enunciaremos, algunas de estas condiciones relacionadas con los métodos de cálculos desarrollados en este trabajo.

1. Primeramente, debe tomarse en cuenta qué condiciones debe reunir la función en un intervalo dado, para que se dé la convergencia a las soluciones buscadas, como por ejemplo: la presencia de raíces múltiples, complejas, cercanas en el intervalo, la continuidad o diferenciabilidad de la función en el intervalo, etc. Aquellos métodos que exigen condiciones más débiles a la función, pueden considerarse como superiores bajo este criterio.
2. Por otro lado, si asumimos que la función verifica las condiciones exigidas por cada método en su intervalo respectivo, podemos considerar entonces qué método exige menor cantidad de cálculos aritméticos y lógicos, es decir, considerar el procedimiento que es, desde el punto de vista computacional, más eficiente. Es lógico que el método que presenta menos cálculos aritméticos se coloca como candidato a sufrir menos problemas de propagación de error de redondeo aunque no lo garantiza.
3. Aquellos métodos de cálculo que, independientemente del número de operaciones aritméticas que involucran, convergen más rápido que los demás, se presentan como buenas opciones, ya que significarán poco trabajo numérico por parte de la máquina.
4. Finalmente, los métodos de cálculo que, independientemente del número de operaciones aritméticas que involucran, no presentan la propagación del error por redondeo se presentan como altamente deseables, ya que sus resultados, independientemente de lo lentos que sean, serán precisos.

Es evidente, entonces que ninguno de los métodos que hemos desarrollado así como los que hemos mencionado alternativamente , se presenta como superior en todas las situaciones mencionadas. Esto, refuerza la hipótesis de que no hay un método completamente más efectivo que los demás, y que sólo un concienzudo análisis de las necesidades particulares de cada problema, podrá conducirnos a las soluciones que buscamos.

A continuación, pues, estableceremos las convenciones que emplearemos en cuanto a la confección de métodos de cálculo, listas de variables, programas y ejemplos que probarán la eficiencia de dichos programas.

3.5. CONVENCIONES EN CUANTO A LA IMPLEMENTACIÓN COMPUTACIONAL DE LOS MÉTODOS DESARROLLADOS.

La secuencia de pasos que describiremos a continuación será aplicada sobre cada uno de los métodos que hemos desarrollado teóricamente a lo largo del capítulo 2.

En primer término, dejaremos establecido que no plasmaremos la lógica del método a través de un algoritmo claramente definido, como usualmente se hace, sino que nos dedicaremos de lleno a los problemas relativos a la implementación del método en la computadora, ya que dentro del programa se encuentra implícita la lógica del método, es decir, su algoritmo.

Además, en cada etapa de la aplicación de los métodos acotación, separación y aproximación , se organizará un listado que contendrá las variables más importantes de los programas, así como su representación dentro de los métodos en general. También se establecerá que tipo de variables son.

Hay que recalcar que la lista de variables a que nos referimos será común a cada etapa del proceso de cálculo de las raíces y no a cada programa desarrollado , ya que las aplicaciones creadas en cada etapa guardan ciertas semejanzas en cuanto a las variables usadas, de modo que construir una lista de variables para cada programa que se desarrolle involucra un trabajo innecesario.

De este modo detallaremos, en primer término, los ajustes que realizaremos al método, en función de las necesidades computacionales, así como cualquier aclaración que se crea conveniente hacer respecto al mismo.

Seguido a esto, presentaremos la codificación, en el lenguaje BASIC, del método y una tabla que contendrá un grupo de problemas escogidos, con respuestas que consideraremos como exactas y las que genera el programa. Finalmente, si es necesario, se realizarán algunos comentarios acerca del comportamiento del método.

Luego de desarrollar los métodos de cada sección (acotación, separación y aproximación de raíces), se fusionarán los mejores métodos desarrollados dentro de las limitaciones que existen para determinar la efectividad de cada uno , con miras a crear métodos de aplicación general, al final del trabajo.

3.6. DESARROLLO DE LOS MÉTODOS SEGÚN EL PLAN TRAZADO.

La implementación en computadora de todos los métodos que desarrollaremos, se realizará empleando una computadora personal Epson Equity I; con el sistema operativo MS-DOS, versión 3.30, y utilizando como compilador de lenguaje BASIC, al Turbo BASIC, versión 1.0. Consideraremos como valores exactos de las raíces, los generados al resolver los polinomios respectivos con el programa Eureka, versión 1.0.

3.6.1. MÉTODOS QUE ACOTAN RAÍCES REALES.

A continuación, implementaremos, en computadora, los métodos que hemos desarrollado para acotar los ceros de una ecuación algebraica de raíces reales. En primer término, presentamos un listado de las variables más importantes que usarán los programas de acotación de raíces.

VARIABLES MÁS IMPORTANTES QUE USARÁN LOS PROGRAMAS SOBRE ACOTACIÓN DE RAÍCES REALES.

VARIABLE	SIGNIFICADO
TI\$(4), TI\$(5)	Vectores alfanuméricos donde se guardan las salidas del programa.
ETIQ	Variable numérica (de tipo precisión simple) que representa el número de etiquetas a imprimir, en un momento dado en la ejecución del programa.
N	Variable numérica (de tipo precisión simple) que representa el grado del polinomio.
P\$(N), N\$(N), M\$(N) y N\$(N+1,N)	Vectores numéricos (de tipo doble precisión) donde se guardan los polinomios originales y los obtenidos.
R y K	Variables numéricas (de tipo precisión simple) que representan un control para la cota inferior y superior, respectivamente.
SW	Variable numérica (de tipo precisión simple); es un interruptor, controla la salida de la información, según se busque la cota inferior o superior.

VARIABLE	SIGNIFICADO
SUP	Variable numérica (de tipo precisión simple); es un límite de la cota que se busca.
Z\$	Variable alfanumérica; es un interruptor, controla el avance del programa, para que el usuario pueda leer las pantallas.
COTA\$, VARIABLE\$, RAICES\$ y SIGNO\$	Variables alfanuméricas usadas para confeccionar las salidas.
P, M, G# y A#	Variables numéricas (de simple y doble precisión) que en la ecuación original representan, respectivamente: la posición del primer coeficiente negativo, la posición del coeficiente mínimo, el valor absoluto del coeficiente mínimo y el coeficiente del término x^N .
L	Variable numérica (de tipo precisión simple) que representa el valor de la cota localizada en los métodos del mayor de los módulos de los coeficientes negativos de la ecuación y el mixto de acotación.

3.6.1.1. MÉTODO DE LAGUERRE-THIBAUT.

Implementar un programa que realice la acotación de las raíces de una ecuación algebraica de raíces reales, por este método, según lo desarrollado en el capítulo 2, es una tarea relativamente sencilla.

Sin embargo, para impedir que el programa lleve a la máquina dentro de un ciclo sin fin o extremadamente largo (que sería el caso cuando las raíces de la

ecuación se ubican en posiciones muy alejadas del origen del sistema de coordenadas), hemos considerado dentro del programa que el usuario establezca hasta qué valores se probarán, en la ecuación, las cotas superiores e inferiores.

A continuación, pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO DE LAGUERRE-THIBAUT.

```

10 REM Programa número 1.
20 REM Cálculo de las cotas inferior y superior de
30 REM las raíces reales de una ecuación algebraica
40 REM de coeficientes reales, empleando el método de
50 REM Laguerre-Thibault.
60 REM Bloque número 1.
70 REM Presentación del programa.
80 KEY OFF
90 CLS
100 CLEAR
110 DIM TI$(4)
120 TI$(0)="Cálculo de las cotas inferior y superior de"
130 TI$(1)="las raíces reales de una ecuación algebraica"
140 TI$(2)="de coeficientes reales, empleando"
150 TI$(3)="el método de Laguerre-Thibault"
160 TI$(4)="Presione cualquier tecla para continuar"
170 ETIQ=4
180 GOSUB 2500
190 GOSUB 3000
200 REM Bloque número 2.
210 REM Carga del grado del polinomio.
220 CLS
230 CLEAR
240 DIM TI$(4)
250 TI$(0)="Por favor, introduzca el grado del"
260 TI$(1)="polinomio (N) y pulse ``Enter'"
270 TI$(2)="para continuar"
280 ETIQ=2
290 GOSUB 2500
300 PRINT
310 PRINT
320 INPUT "N=",N
330 IF N>0 AND N=FIX(N) THEN 410

```

```

340 CLS
350 TI$(0)="La información introducida es incorrecta"
360 TI$(1)="Presione cualquier tecla para continuar"
370 ETIQ=1
380 GOSUB 2500
390 GOSUB 3000
400 GOTO 220
410 REM Bloque número 3.
420 REM Dimensiones de los vectores usados.
430 DIM P#(N),N#(N),M#(N)
440 REM Bloque número 4.
450 REM Carga de los coeficientes del polinomio.
460 CLS
470 TI$(0)="Por favor, introduzca los coeficientes del"
480 TI$(1)="polinomio, en orden descendente y pulse"
490 TI$(2)="``Enter'' para continuar, en cada caso"
500 TI$(3)="Presione cualquier tecla para continuar"
510 ETIQ=3
520 GOSUB 2500
530 GOSUB 3000
540 CLS
550 FOR I=N TO 0 STEP -1
560 PRINT "Introduzca el coeficiente de X^";I;
570 INPUT "=",P#(I)
580 PRINT
590 PRINT
600 NEXT I
610 IF P#(N)<>0# AND P#(0)<>0# THEN 710
620 CLS
630 TI$(0)="El polinomio introducido puede simplificarse,"
640 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+" ,"
650 TI$(2)="ni el de X^ 0, pueden ser nulos"
660 TI$(3)="Presione cualquier tecla para continuar"
670 ETIQ=3
680 GOSUB 2500
690 GOSUB 3000
700 GOTO 220
710 REM Bloque número 5.
720 REM Carga de los límites inferior y superior de
730 REM las cotas a calcular.
740 CLS
750 TI$(0)="Introduzca un número entero"
760 TI$(1)="negativo (R), que permitirá finalizar el"
770 TI$(2)="cálculo de la cota inferior buscada y pulse"
780 TI$(3)="``Enter'' para continuar"
790 ETIQ=3
800 GOSUB 2500

```

```

810 PRINT
820 PRINT
830 INPUT "R=",R
840 IF R=<0 AND R=FIX(R) THEN 920
850 CLS
860 TI$(0)="La información introducida es incorrecta"
870 TI$(1)="Presione cualquier tecla para continuar"
880 ETIQ=1
890 GOSUB 2500
900 GOSUB 3000
910 GOTO 740
920 CLS
930 TI$(0)="Introduzca un número entero"
940 TI$(1)="positivo (K), que permitirá finalizar el"
950 TI$(2)="cálculo de la cota superior buscada y pulse"
960 TI$(3)="`Enter` para continuar"
970 ETIQ=3
980 GOSUB 2500
990 PRINT
1000 PRINT
1010 INPUT "K=",K
1020 IF K=>0 AND K=FIX(K) THEN 1100
1030 CLS
1040 TI$(0)="La información introducida es incorrecta"
1050 TI$(1)="Presione cualquier tecla para continuar"
1060 ETIQ=1
1070 GOSUB 2500
1080 GOSUB 3000
1090 GOTO 920
1100 REM Bloque número 6.
1110 REM Inicialización del polinomio de trabajo.
1120 CLS
1130 FOR I=0 TO N
1140 N#(I)=CDBL((-1)^I)*P#(I)
1150 NEXT I
1160 REM Bloque número 7.
1170 REM Verificar que el coeficiente de X^N es
1180 REM mayor que cero.
1190 GOSUB 3500
1200 REM Bloque número 8.
1210 REM Cálculo de la cota inferior de las raíces del
1220 REM polinomio.
1230 SW=1
1240 SUP=-R
1250 COTA$="inferior"
1260 RAICES$="negativas"
1270 VARIABLE$="S"

```

```

1280 SIGNO$="menor"
1290 GOSUB 4000
1300 REM Bloque número 9.
1310 REM Reinicialización del polinomio de trabajo.
1320 FOR I=0 TO N
1330 N#(I)=P#(I)
1340 NEXT I
1350 REM Bloque número 10.
1360 REM Verificar que el coeficiente de  $X^N$  es
1370 REM mayor que cero.
1380 GOSUB 3500
1390 REM Bloque número 11.
1400 REM Cálculo de la cota superior de las raíces del
1410 REM polinomio.
1420 SW=0
1430 SUP=K
1440 COTA$="superior"
1450 RAICES$="positivas"
1460 VARIABLE$="L"
1470 SIGNO$="mayor"
1480 GOSUB 4000
1490 REM Bloque número 12.
1500 REM Fin del programa.
1510 CLS
1520 TI$(0)="¿Desea ejecutar nuevamente el programa?"
1530 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
1540 TI$(2)="en caso contrario, cualquier otra tecla para"
1550 TI$(3)="finalizar la ejecución del programa"
1560 ETIQ=3
1570 GOSUB 2500
1580 GOSUB 3000
1590 IF Z$<>"1" THEN 1700
1600 CLS
1610 TI$(0)="¿Desea realizar los cálculos sobre el mismo"
1620 TI$(1)="polinomio? Si es así, pulse ``1'' para"
1630 TI$(2)="hacerlo, y, en caso contrario, cualquier otra"
1640 TI$(3)="tecla para cargar todos los datos nuevamente"
1650 ETIQ=3
1660 GOSUB 2500
1670 GOSUB 3000
1680 IF Z$="1" THEN 710 ELSE GOSUB 5500
1690 GOTO 200
1700 KEY ON
1710 CLS
1720 GOSUB 6000
1730 STOP
2000 REM Subrutinas del programa.

```

```
2500 REM Subrutina número 1.
2510 REM Impresión centrada de las etiquetas del
2520 REM programa, por página.
2530 FOR I=0 TO ETIQ
2540 FOR J=1 TO 3
2550 PRINT
2560 NEXT J
2570 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
2580 NEXT I
2590 RETURN
3000 REM Subrutina número 2.
3010 REM Paralización del avance del programa mientras
3020 REM el usuario lee los mensajes.
3030 Z$=""
3040 WHILE Z$=""
3050 Z$=INKEY$
3060 WEND
3070 RETURN
3500 REM Subrutina número 3.
3510 REM Verificar que el coeficiente de  $X^N$  es
3520 REM mayor que cero.
3530 IF N#(N)>0# THEN 3570
3540 FOR I=0 TO N
3550 N#(I)=-N#(I)
3560 NEXT I
3570 RETURN
4000 REM Subrutina número 4.
4010 REM Método de Laguerre-Thibault.
4020 FOR J=0 TO SUP
4030 M#(N)=N#(N)
4040 FOR M=N-1 TO 0 STEP -1
4050 M#(M)=CDBL(J)*M#(M+1)+N#(M)
4060 NEXT M
4070 FOR M=0 TO N
4080 IF M#(M)<=0# THEN 4160
4090 NEXT M
4100 L=J
4110 GOSUB 4500
4120 ETIQ=2
4130 GOSUB 2500
4140 GOSUB 3000
4150 GOTO 4210
4160 NEXT J
4170 GOSUB 5000
4180 ETIQ=2
4190 GOSUB 2500
4200 GOSUB 3000
```

```

4210 RETURN
4500 REM Subrutina número 5.
4510 REM Mensajes de salida cuando se encuentra la cota.
4520 TI$(0)="Una cota "+COTA$+" de las raíces "+RAICES$
4530 TI$(0)=TI$(0)+" de"
4540 TI$(1)="la ecuación es "+VARIABLE$+"="+STR$(L*(-1)^SW)
4550 TI$(2)="Presione cualquier tecla para continuar"
4560 RETURN
5000 REM Subrutina número 6.
5010 REM Mensajes de salida cuando no se encuentra la cota.
5020 TI$(0)="Cualquier cota "+COTA$+" de las raíces "
5030 TI$(0)=TI$(0)+RAICES$+" de"
5040 TI$(1)="la ecuación es "+SIGNO$+" que "
5050 TI$(1)=TI$(1)+STR$(SUP*(-1)^SW)
5060 TI$(2)="Presione cualquier tecla para continuar"
5070 RETURN
5500 REM Subrutina número 7.
5510 REM Inicialización de las variables del programa
5520 REM para otra ejecución.
5530 ERASE P#,N#,M#
5540 SW=0
5550 SUP=0
5560 COTA$=""
5570 RAICES$=""
5580 VARIABLE$=""
5590 SIGNO$=""
5600 RETURN
6000 REM Subrutina número 8.
6010 REM Fin del programa.
6020 END

```

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido acotadas empleando el programa que acabamos de mostrar.

TABLA 5

ECUACIÓN: $P(x)=0$	RAÍCES VERDADERAS	INTERVALO SEGÚN EL MÉTODO
$x^3 - 5x + 1$	-2.330, 0.202, 2.128	$(-3,3)$
$x^3 - 7x - 7$	-1.692, -1.357, 3.049	$(-3,4)$
$x^4 - 41x^2 + 400$	-5, -4, 4, 5	$(-7,7)$
$3x^4 - 40x^3 + 130x^2 - 120x + 27$	0.333, 1, 3, 9	$(0,14)$
$x^4 - 40x^3 + 390x^2 - 1080x + 729$	1, 3, 9, 27	$(0,41)$
$x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	-2,1,3,4	$(-3,9)$

Es evidente que las cotas de las raíces que genera este método, pueden llegar a ser muy distantes de los valores reales de las raíces; sin embargo, son un buen punto de partida para iniciar el proceso de separación de raíces, que refinará estas acotaciones.

Además, debemos dejar sentado que se emplearon límites de las cotas superior e inferior suficientemente grandes números mayores que 100 en valor absoluto, cuando las raíces de las ecuaciones no eran mayores que 30. Es por eso que el método localizó, en todos los casos, los intervalos que acotan las raíces.

3.6.1.2. MÉTODO DEL MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN.

Implementar un programa que realice la acotación de los ceros de una ecuación algebraica de raíces reales, según el desarrollo del método realizado en el capítulo 2, es una tarea relativamente sencilla, ya que el único inconveniente que pudiera surgir cuando la ecuación posee todos los coeficientes del mismo signo se remedia gracias a un teorema de teoría de ecuaciones que nos garantiza que cero es la cota que se busca (ver teorema número 8 del apéndice de teoría de ecuaciones).

Por otro lado, dado que deseamos obtener cotas de las raíces de la ecuación, hemos modificado la fórmula del método del mayor de los módulos de los coeficientes negativos de la ecuación, de modo que origine las cotas, considerándolas en valor absoluto, como el menor entero mayor que la cota que genera dicho método normalmente.

Otro problema que pudiera surgir aunque poco probable y trivial ocurre cuando alguno de los valores que genera la fórmula del mayor de los módulos exceda la capacidad del compilador usado. Dicha situación sólo se puede remediar, básicamente, empleando un compilador más poderoso.

A continuación, pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO DEL MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN.

```

10 REM Programa número 2.
20 REM Cálculo de las cotas inferior y superior de
30 REM las raíces reales de una ecuación algebraica
40 REM de coeficientes reales, empleando el método
50 REM del mayor de los módulos de los coeficientes
60 REM Negativos de la ecuación.
70 REM Bloque número 1.
80 REM Presentación del programa.
90 KEY OFF
100 CLS
110 CLEAR
120 DIM TI$(5)
130 TI$(0)="Cálculo de las cotas inferior y superior de"
140 TI$(1)="las raíces reales de una ecuación algebraica"
150 TI$(2)="de coeficientes reales, empleando el método"
160 TI$(3)="del mayor de los módulos de los coeficientes:"
170 TI$(4)="negativos de la ecuación"
180 TI$(5)="Presione cualquier tecla para continuar"
190 ETIQ=5
200 GOSUB 2000

```

```

210 GOSUB 2500
220 REM Bloque número 2.
230 REM Carga del grado del polinomio.
240 CLS
250 CLEAR
260 DIM TI$(5)
270 TI$(0)="Por favor, introduzca el grado del"
280 TI$(1)="polinomio (N) y pulse ``Enter'"
290 TI$(2)="para continuar"
300 ETIQ=2
310 GOSUB 2000
320 PRINT
330 PRINT
340 INPUT "N=",N
350 IF N>0 AND N=FIX(N) THEN 430
360 CLS
370 TI$(0)="La información introducida es incorrecta"
380 TI$(1)="Presione cualquier tecla para continuar"
390 ETIQ=1
400 GOSUB 2000
410 GOSUB 2500
420 GOTO 240
430 REM Bloque número 3.
440 REM Dimensiones de los vectores usados.
450 DIM P#(N),N#(N)
460 REM Bloque número 4.
470 REM Carga de los coeficientes del polinomio.
480 CLS
490 TI$(0)="Por favor, introduzca los coeficientes del"
500 TI$(1)="polinomio, en orden descendente y pulse"
510 TI$(2)="``Enter'" para continuar, en cada caso"
520 TI$(3)="Presione cualquier tecla para continuar"
530 ETIQ=3
540 GOSUB 2000
550 GOSUB 2500
560 CLS
570 FOR I=N TO 0 STEP -1
580 PRINT "Introduzca el coeficiente de X^";I;
590 INPUT "=",P#(I)
600 PRINT
610 PRINT
620 NEXT I
630 IF P#(N)<>0# AND P#(0)<>0# THEN 730
640 CLS
650 TI$(0)="El polinomio introducido puede simplificarse,"
660 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+" ,"
670 TI$(2)="ni el de X^ 0, pueden ser nulos"

```

```
680 TI$(3)="Presione cualquier tecla para continuar"
690 ETIQ=3
700 GOSUB 2000
710 GOSUB 2500
720 GOTO 240
730 REM Bloque número 5.
740 REM Inicialización del polinomio de trabajo.
750 CLS
760 FOR I=0 TO N
770 N#(I)=CDBL((-1)^I)*P#(I)
780 NEXT I
790 CLS
800 REM Bloque número 6.
810 REM Verificar que el coeficiente de  $X^N$  es
820 REM mayor que cero.
830 GOSUB 3000
840 REM Bloque número 7.
850 REM Examinar si cero es o no cota inferior de las
860 REM raíces del polinomio.
870 CONTROL$=""
880 GOSUB 3500
890 REM Bloque número 8.
900 REM Cálculo de la cota inferior de las raíces del
910 REM polinomio.
920 SW=1
930 COTA$="inferior"
940 RAICES$="negativas"
950 VARIABLE$="S"
960 GOSUB 4000
970 REM Bloque número 9.
980 REM Reinicialización del polinomio de trabajo.
990 FOR I=0 TO N
1000 N#(I)=P#(I)
1010 NEXT I
1020 REM Bloque número 10.
1030 REM Verificar que el coeficiente de  $X^N$  es
1040 REM mayor que cero.
1050 GOSUB 3000
1060 REM Bloque número 11.
1070 REM Examinar si cero es o no cota superior de las
1080 REM raíces del polinomio.
1090 CONTROL$=""
1100 GOSUB 3500
1110 REM Bloque número 12.
1120 REM Cálculo de la cota superior de las raíces del
1130 REM polinomio.
1140 SW=0
```

```

1150 COTA$="superior"
1160 RAICES$="positivas"
1170 VARIABLE$="L"
1180 GOSUB 4000
1190 REM Bloque número 13.
1200 REM Fin del programa.
1210 CLS
1220 TI$(0)="¿Desea ejecutar nuevamente el programa?"
1230 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
1240 TI$(2)="en caso contrario, cualquier otra tecla para"
1250 TI$(3)="finalizar la ejecución del programa"
1260 ETIQ=3
1270 GOSUB 2000
1280 GOSUB 2500
1290 IF Z$="1" THEN 220
1300 KEY ON
1310 CLS
1320 GOSUB 5500
1330 STOP
1500 REM Subrutinas del programa.
2000 REM Subrutina número 1.
2010 REM Impresión centrada de las etiquetas del
2020 REM programa, por página.
2030 FOR I=0 TO ETIQ
2040 FOR J=1 TO 3
2050 PRINT
2060 NEXT J
2070 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
2080 NEXT I
2090 RETURN
2500 REM Subrutina número 2.
2510 REM Paralización del avance del programa mientras
2520 REM el usuario lee los mensajes.
2530 Z$=""
2540 WHILE Z$=""
2550 Z$=INKEY$
2560 WEND
2570 RETURN
3000 REM Subrutina número 3.
3010 REM Verificar que el coeficiente de  $X^N$  es
3020 REM mayor que cero.
3030 IF N#(N)>0# THEN 3070
3040 FOR I=0 TO N
3050 N#(I)=-N#(I)
3060 NEXT I
3070 RETURN
3500 REM Subrutina número 4.

```

```

3510 REM Examinar si cero es o no cota de las raíces
3520 REM del polinomio.
3530 FOR I=N-1 TO 0 STEP -1
3540 IF N#(I)<0# THEN 3590
3550 NEXT I
3560 L=0
3570 CONTROL$="CERO"
3580 GOTO 3600
3590 P=I
3600 RETURN
4000 REM Subrutina número 5.
4010 REM Método del mayor de los módulos de los
4020 REM coeficientes negativos de la ecuación.
4030 IF CONTROL$="CERO" THEN 4120
4040 M=0
4050 FOR I=1 TO N
4060 IF N#(M)<=N#(I) THEN 4080
4070 M=I
4080 NEXT I
4090 G#=ABS(N#(M))
4100 A#=N#(N)
4110 L=1+INT(1#+(G#/A#)^(1/(N-P)))
4120 GOSUB 4500
4130 ETIQ=2
4140 GOSUB 2000
4150 GOSUB 2500
4160 RETURN
4500 REM Subrutina número 6.
4510 REM Mensajes de salida cuando se encuentra la cota.
4520 TI$(0)="Una cota "+COTA$+" de las raíces "+RAICES$
4530 TI$(0)=TI$(0)+" de"
4540 TI$(1)="la ecuación es "+VARIABLE$+"="+STR$(L*(-1)^SW)
4550 TI$(2)="Presione cualquier tecla para continuar"
4560 RETURN
5000 REM Subrutina número 7.
5010 REM Inicialización de las variables del programa
5020 REM para otra ejecución.
5030 ERASE P#,N#
5040 SW=0
5050 COTA$=""
5060 RAICES$=""
5070 VARIABLE$=""
5080 RETURN
5500 REM Subrutina número 8.
5510 REM Fin del programa.
5520 END

```

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales, éstas han sido acotadas empleando el programa que acabamos de mostrar.

TABLA 6

ECUACIÓN: $P(x)=0$	RAÍCES VERDADERAS	INTERVALO SEGÚN EL MÉTODO
$x^3 - 5x + 1$	- 2.330, 0.202, 2.128	(- 4,4)
$x^3 - 7x - 7$	- 1.692, - 1.357, 3.049	(- 4,4)
$x^4 - 41x^2 + 400$	- 5, - 4, 4, 5	(- 8,8)
$3x^4 - 40x^3 + 130x^2 - 120x + 27$	0.333, 1, 3, 9	(0,42)
$x^4 - 40x^3 + 390x^2 - 1080x + 729$	1, 3, 9, 27	(0,1082)
$x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	- 2,1,3,4	(- 6,162)

Es evidente que las cotas de las raíces que genera este método pueden llegar a ser muy distantes de los valores reales de las raíces; sin embargo, son un buen punto de partida para iniciar el proceso de separación de raíces que refinará estas acotaciones.

Además, debemos dejar sentado que al contrario del método de Laguerre-Thibault, este método siempre generará las cotas buscadas, por lo que es un buen candidato para poner en marcha a métodos como el de Laguerre-Thibault, ya que éste, en casi todos los casos, genera valores menores que el método del mayor de los módulos, pero requiere una cota del número máximo de iteraciones que se realizarán.

3.6.1.3. MÉTODO DE NEWTON.

La implementación de un programa que realice la acotación de las raíces de una ecuación algebraica de raíces reales, por este método, es una tarea relativamente sencilla. Básicamente se enmarca en los mismos términos que los del método de Laguerre-Thibault.

Así, para impedir que el programa lleve a la máquina a un ciclo sin fin o extremadamente largo (que sería el caso cuando las raíces de la ecuación se ubican en posiciones muy alejadas del origen del sistema de coordenadas), hemos considerado dentro del algoritmo, que el usuario establezca hasta qué valores se probarán, en la ecuación, las cotas superiores e inferiores; iguales consideraciones fueron hechas para el método de Laguerre-Thibault.

Como este método presenta la particularidad de necesitar sólo los signos de las k -ésimas derivadas —evaluados en determinados puntos—, podemos emplear un teorema (ver apéndice de teoría de ecuaciones teorema número 10) que nos permite conocer dicho signo, a través de la división sintética, que resulta más práctico que calcular y evaluar las k -ésimas derivadas del polinomio original.

A continuación pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO DE NEWTON

```

10 REM Programa número 3.
20 REM Cálculo de las cotas inferior y superior de
30 REM las raíces reales de una ecuación algebraica
40 REM de coeficientes reales, empleando el método de
50 REM Newton.
60 REM Bloque número 1.
70 REM Presentación del programa.
```

```
80 KEY OFF
90 CLS
100 CLEAR
110 DIM TI$(4)
120 TI$(0)="Cálculo de las cotas inferior y superior de"
130 TI$(1)="las raíces reales de una ecuación algebraica"
140 TI$(2)="de coeficientes reales, empleando"
150 TI$(3)="el método de Newton"
160 TI$(4)="Presione cualquier tecla para continuar"
170 ETIQ=4
180 GOSUB 2500
190 GOSUB 3000
200 REM Bloque número 2.
210 REM Carga del grado del polinomio.
220 CLS
230 CLEAR
240 DIM TI$(4)
250 TI$(0)="Por favor, introduzca el grado del"
260 TI$(1)="polinomio (N) y pulse ``Enter'"
270 TI$(2)="para continuar"
280 ETIQ=2
290 GOSUB 2500
300 PRINT
310 PRINT
320 INPUT "N=",N
330 IF N>0 AND N=FIX(N) THEN 410
340 CLS
350 TI$(0)="La información introducida es incorrecta"
360 TI$(1)="Presione cualquier tecla para continuar"
370 ETIQ=1
380 GOSUB 2500
390 GOSUB 3000
400 GOTO 220
410 REM Bloque número 3.
420 REM Dimensiones del vector y matriz usados.
430 DIM P#(N),N#(N+1,N)
440 REM Bloque número 4.
450 REM Carga de los coeficientes del polinomio.
460 CLS
470 TI$(0)="Por favor, introduzca los coeficientes del"
480 TI$(1)="polinomio, en orden descendente y pulse"
490 TI$(2)="``Enter'" para continuar, en cada caso"
500 TI$(3)="Presione cualquier tecla para continuar"
510 ETIQ=3
520 GOSUB 2500
530 GOSUB 3000
540 CLS
```



```

550 FOR I=N TO 0 STEP -1
560 PRINT "Introduzca el coeficiente de X^";I;
570 INPUT "=",P#(I)
580 PRINT
590 PRINT
600 NEXT I
610 IF P#(N)<>0# AND P#(0)<>0# THEN 710
620 CLS
630 TI$(0)="El polinomio introducido puede simplificarse,"
640 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+", "
650 TI$(2)="ni el de X^ 0, pueden ser nulos"
660 TI$(3)="Presione cualquier tecla para continuar"
670 ETIQ=3
680 GOSUB 2500
690 GOSUB 3000
700 GOTO 220
710 REM Bloque número 5.
720 REM Carga de los límites inferior y superior de
730 REM las cotas a calcular.
740 CLS
750 TI$(0)="Introduzca un número entero"
760 TI$(1)="negativo (R), que permitirá finalizar el"
770 TI$(2)="cálculo de la cota inferior buscada y pulse"
780 TI$(3)="``Enter'' para continuar"
790 ETIQ=3
800 GOSUB 2500
810 PRINT
820 PRINT
830 INPUT "R=",R
840 IF R=<0 AND R=FIX(R) THEN 920
850 CLS
860 TI$(0)="La información introducida es incorrecta"
870 TI$(1)="Presione cualquier tecla para continuar"
880 ETIQ=1
890 GOSUB 2500
900 GOSUB 3000
910 GOTO 740
920 CLS
930 TI$(0)="Introduzca un número entero"
940 TI$(1)="positivo (K), que permitirá finalizar el"
950 TI$(2)="cálculo de la cota superior buscada y pulse"
960 TI$(3)="``Enter'' para continuar"
970 ETIQ=3
980 GOSUB 2500
990 PRINT
1000 PRINT
1010 INPUT "K=",K

```

```
1020 IF K=>0 AND K=FIX(K) THEN 1100
1030 CLS
1040 TI$(0)="La información introducida es incorrecta"
1050 TI$(1)="Presione cualquier tecla para continuar"
1060 ETIQ=1
1070 GOSUB 2500
1080 GOSUB 3000
1090 GOTO 920
1100 REM Bloque número 6.
1110 REM Inicialización del polinomio de trabajo.
1120 CLS
1130 FOR I=N TO 0 STEP -1
1140 N#(0,I)=CDBL((-1)^I)*P#(I)
1150 NEXT I
1160 REM Bloque número 7.
1170 REM Verificar que el coeficiente de X^N es
1180 REM mayor que cero.
1190 GOSUB 3500
1200 REM Bloque número 8.
1210 REM Cálculo de la cota inferior de las raíces del
1220 REM polinomio.
1230 SW=1
1240 SUP=-R
1250 COTA$="inferior"
1260 RAICES$="negativas"
1270 VARIABLE$="S"
1280 SIGNO$="menor"
1290 GOSUB 4000
1300 REM Bloque número 9.
1310 REM Reinicialización del polinomio de trabajo.
1320 FOR I=0 TO N
1330 N#(0,I)=P#(I)
1340 NEXT I
1350 REM Bloque número 10.
1360 REM Verificar que el coeficiente de X^N es
1370 REM mayor que cero.
1380 GOSUB 3500
1390 REM Bloque número 11.
1400 REM Cálculo de la cota superior de las raíces del
1410 REM polinomio.
1420 SW=0
1430 SUP=K
1440 COTA$="superior"
1450 RAICES$="positivas"
1460 VARIABLE$="L"
1470 SIGNO$="mayor"
1480 GOSUB 4000
```

```

1490 REM Bloque número 12.
1500 REM Fin del programa.
1510 CLS
1520 TI$(0)="¿Desea ejecutar nuevamente el programa?"
1530 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
1540 TI$(2)="en caso contrario, cualquier otra tecla para"
1550 TI$(3)="finalizar la ejecución del programa"
1560 ETIQ=3
1570 GOSUB 2500
1580 GOSUB 3000
1590 IF Z$<>"1" THEN 1700
1600 CLS
1610 TI$(0)="¿Desea realizar los cálculos sobre el mismo"
1620 TI$(1)="polinomio? Si es así, pulse ``1'' para"
1630 TI$(2)="hacerlo, y, en caso contrario, cualquier otra"
1640 TI$(3)="tecla para cargar todos los datos nuevamente"
1650 ETIQ=3
1660 GOSUB 2500
1670 GOSUB 3000
1680 IF Z$="1" THEN 710 ELSE GOSUB 5500
1690 GOTO 200
1700 KEY ON
1710 CLS
1720 GOSUB 6000
1730 STOP
2000 REM Subrutinas del programa.
2500 REM Subrutina número 1.
2510 REM Impresión centrada de las etiquetas del
2520 REM programa, por página.
2530 FOR I=0 TO ETIQ
2540 FOR J=1 TO 3
2550 PRINT
2560 NEXT J
2570 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
2580 NEXT I
2590 RETURN
3000 REM Subrutina número 2.
3010 REM Paralización del avance del programa mientras
3020 REM el usuario lee los mensajes.
3030 Z$=""
3040 WHILE Z$=""
3050 Z$=INKEY$
3060 WEND
3070 RETURN
3500 REM Subrutina número 3.
3510 REM Verificar que el coeficiente de  $X^N$  es
3520 REM mayor que cero.

```

```

3530 IF N#(0,N)>0# THEN 3570
3540 FOR I=0 TO N
3550 N#(0,I)=-N#(0,I)
3560 NEXT I
3570 RETURN
4000 REM Subrutina número 4.
4010 REM Método de Newton.
4020 FOR J=1 TO N+1
4030 N#(J,N)=N#(0,N)
4040 NEXT J
4050 FOR P=0 TO SUP
4060 GR=0
4070 FOR M=1 TO N
4080 FOR J=N-1 TO GR STEP -1
4090 N#(M,J)=N#(M,J+1)*CDBL(P)+N#(M-1,J)
4100 NEXT J
4110 GR=GR+1
4120 NEXT M
4130 J=0
4140 FOR M=1 TO N+1
4150 IF N#(M,J)<=0# THEN 4240
4160 J=J+1
4170 NEXT M
4180 L=P
4190 GOSUB 4500
4200 ETIQ=2
4210 GOSUB 2500
4220 GOSUB 3000
4230 GOTO 4290
4240 NEXT P
4250 GOSUB 5000
4260 ETIQ=2
4270 GOSUB 2500
4280 GOSUB 3000
4290 RETURN
4500 REM Subrutina número 5.
4510 REM Mensajes de salida cuando se encuentra la cota.
4520 TI$(0)="Una cota "+COTA$+" de las raíces "+RAICES$
4530 TI$(0)=TI$(0)+" de"
4540 TI$(1)="la ecuación es "+VARIABLE$+"="+STR$(L*(-1)^SW)
4550 TI$(2)="Presione cualquier tecla para continuar"
4560 RETURN
5000 REM Subrutina número 6.
5010 REM Mensajes de salida cuando no se encuentra la cota.
5020 TI$(0)="Cualquier cota "+COTA$+" de las raíces "
5030 TI$(0)=TI$(0)+RAICES$+" de"
5040 TI$(1)="la ecuación es "+SIGNO$+" que "

```

```

5050 TI$(1)=TI$(1)+STR$(SUP*(-1)^SW)
5060 TI$(2)="Presione cualquier tecla para continuar"
5070 RETURN
5500 REM Subrutina número 7.
5510 REM Inicialización de las variables del programa
5520 REM para otra ejecución.
5530 ERASE P#,N#
5540 SW=0
5550 SUP=0
5560 COTA$=""
5570 RAICES$=""
5580 VARIABLE$=""
5590 SIGNO$=""
5600 RETURN
6000 REM Subrutina número 8.
6010 REM Fin del programa.
6020 END

```

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido acotadas empleando el programa que acabamos de mostrar.

TABLA 7

ECUACIÓN: $P(x)=0$	RAÍCES VERDADERAS	INTERVALO SEGÚN EL MÉTODO
$x^3 - 5x + 1$	-2.330, 0.202, 2.128	(-3,3)
$x^3 - 7x - 7$	-1.692, -1.357, 3.049	(-2,4)
$x^4 - 41x^2 + 400$	-5, -4, 4, 5	(-6,6)
$3x^4 - 40x^3 + 130x^2 - 120x + 27$	0.333, 1, 3, 9	(0,10)
$x^4 - 40x^3 + 390x^2 - 1080x + 729$	1, 3, 9, 27	(0,28)
$x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	-2,1,3,4	(-3,5)

Es evidente que las cotas de las raíces que genera esta técnica son mejores que las que las obtenidas a través del método de Laguerre-Thibault, y, superiores a las que surgen al aplicar el procedimiento del mayor de los módulos de los coeficientes negativos de la ecuación, de modo que se presenta como mejor

candidato que el método de Laguerre-Thibault para ser punto de partida en el proceso de separación de raíces que refinará estas acotaciones.

Lo que hemos desarrollado acerca de los tres métodos vistos nos sugiere que un procedimiento general para acotar las raíces de una ecuación polinomial, debe aplicar, en su fase inicial, el método del mayor de los módulos de los coeficientes negativos de la ecuación, para generar valores máximo y mínimo que acoten dichas raíces en forma grosera —considerando estas cotas en valor absoluto, como el menor entero mayor que la cota que genera dicho método—. Luego, se debe aplicar el método de Newton, con el objeto de reducir el tamaño del intervalo acotador, ya que la evidencia empírica nos indica que éste es el más preciso de todos.

Las cotas generadas por el método del mayor de los módulos de los coeficientes negativos de la ecuación, harían el papel, en el método de Newton, de los valores (K) y (R) que introduce el usuario. Además, es evidente que las cotas que surgirán de este método combinado —mayor de los módulos-Newton—, son iguales a las que se obtienen al aplicar el método de Newton —únicamente—, el número de veces que sea necesario.

A continuación, mostraremos el desarrollo de dicho método, que denominaremos “Método mixto de acotación: mayor de los módulos de los coeficientes negativos de la ecuación-Newton”.

3.6.1.4. MÉTODO MIXTO DE ACOTACIÓN: MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN-NEWTON.

La implementación de un programa que realice la acotación de los ceros de una ecuación algebraica de raíces reales, por esta combinación de métodos, no es una tarea tan sencilla como lo han sido los métodos previamente desarrollados.

Básicamente, se enmarca en los mismos términos que el método del mayor de los módulos de los coeficientes negativos de la ecuación, pero empleando el menor entero mayor que las cotas generadas por éste, para obtener las cotas definitivas.

Este método es poco probable que lleve a la máquina a un ciclo sin fin o extremadamente largo (que sería el caso cuando las raíces de la ecuación se ubican en posiciones muy alejadas del origen del sistema de coordenadas).

El único problema que pudiera surgir consiste en que alguno de los valores de la fórmula del mayor de los módulos exceda la capacidad del compilador usado. Esto sólo se puede remediar empleando un compilador más poderoso.

A continuación, pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO MIXTO DE ACOTACIÓN: MAYOR DE LOS MÓDULOS DE LOS COEFICIENTES NEGATIVOS DE LA ECUACIÓN-NEWTON.

```
10 REM Programa número 4.
20 REM Cálculo de las cotas inferior y superior de
30 REM las raíces reales de una ecuación algebraica
```

```

40 REM de coeficientes reales, empleando el método mixto
50 REM de acotación (mayor de los módulos de los
60 REM coeficientes negativos de la ecuación-Newton).
70 REM Bloque número 1.
80 REM Presentación del programa.
90 KEY OFF
100 CLS
110 CLEAR
120 DIM TI$(5)
130 TI$(0)="Cálculo de las cotas inferior y superior de"
140 TI$(1)="las raíces reales de una ecuación algebraica"
150 TI$(2)="de coeficientes reales, empleando el método"
160 TI$(3)="mixto de acotación (mayor de los módulos de los"
170 TI$(4)="coeficientes negativos de la ecuación-Newton)"
180 TI$(5)="Presione cualquier tecla para continuar"
190 ETIQ=5
200 GOSUB 2000
210 GOSUB 2500
220 REM Bloque número 2.
230 REM Carga del grado del polinomio.
240 CLS
250 CLEAR
260 DIM TI$(5)
270 TI$(0)="Por favor, introduzca el grado del"
280 TI$(1)="polinomio (N) y pulse ``Enter'"
290 TI$(2)="para continuar"
300 ETIQ=2
310 GOSUB 2000
320 PRINT
330 PRINT
340 INPUT "N=",N
350 IF N>0 AND N=FIX(N) THEN 430
360 CLS
370 TI$(0)="La información introducida es incorrecta"
380 TI$(1)="Presione cualquier tecla para continuar"
390 ETIQ=1
400 GOSUB 2000
410 GOSUB 2500
420 GOTO 240
430 REM Bloque número 3.
440 REM Dimensiones del vector y matriz usados.
450 DIM P$(N),N$(N+1,N)
460 REM Bloque número 4.
470 REM Carga de los coeficientes del polinomio.
480 CLS
490 TI$(0)="Por favor, introduzca los coeficientes del"
500 TI$(1)="polinomio, en orden descendente y pulse"

```



```

510 TI$(2)="`Enter'" para continuar, en cada caso"
520 TI$(3)="Presione cualquier tecla para continuar"
530 ETIQ=3
540 GOSUB 2000
550 GOSUB 2500
560 CLS
570 FOR I=N TO 0 STEP -1
580 PRINT "Introduzca el coeficiente de X^";I;
590 INPUT "=",P#(I)
600 PRINT
610 PRINT
620 NEXT I
630 IF P#(N)<>0# AND P#(0)<>0# THEN 730
640 CLS
650 TI$(0)="El polinomio introducido puede simplificarse,"
660 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+" ,"
670 TI$(2)="ni el de X^ 0, pueden ser nulos"
680 TI$(3)="Presione cualquier tecla para continuar"
690 ETIQ=3
700 GOSUB 2000
710 GOSUB 2500
720 GOTO 240
730 REM Bloque número 5.
740 REM Inicialización del polinomio de trabajo.
750 FOR I=0 TO N
760 N#(0,I)=CDBL((-1)^I)*P#(I)
770 NEXT I
780 CLS
790 REM Bloque número 6.
800 REM Verificar que el coeficiente de X^N es
810 REM mayor que cero.
820 GOSUB 3000
830 REM Bloque número 7.
840 REM Examinar si cero es o no cota inferior de las
850 REM raíces del polinomio.
860 CONTROL$=""
870 GOSUB 3500
880 REM Bloque número 8.
890 REM Cálculo de una cota inferior de las raíces del
900 REM polinomio por el método del mayor de los
910 REM módulos de los coeficientes negativos de la
920 REM ecuación.
930 GOSUB 4000
940 REM Bloque número 9.
950 REM Refinamiento de la cota inferior de las raíces
960 REM del polinomio por el método de Newton.
970 SW=1

```

```

980 COTA$="inferior"
990 RAICES$="negativas"
1000 VARIABLE$="S"
1010 GOSUB 4500
1020 REM Bloque número 10.
1030 REM Reinicialización del polinomio.
1040 FOR I=0 TO N
1050 N#(0,I)=P#(I)
1060 NEXT I
1070 REM Bloque número 11.
1080 REM Verificar que el coeficiente de X^N es
1090 REM mayor que cero.
1100 GOSUB 3000
1110 REM Bloque número 12.
1120 REM Examinar si cero es o no cota superior de las
1130 REM raíces del polinomio.
1140 CONTROL$=""
1150 GOSUB 3500
1160 REM Bloque número 13.
1170 REM Cálculo de una cota superior de las raíces del
1180 REM polinomio por el método del mayor de los
1190 REM módulos de los coeficientes negativos de la
1200 REM ecuación.
1210 GOSUB 4000
1220 REM Bloque número 14.
1230 REM Refinamiento de la cota superior de las raíces
1240 REM del polinomio por el método de Newton.
1250 SW=0
1260 COTA$="superior"
1270 RAICES$="positivas"
1280 VARIABLE$="L"
1290 GOSUB 4500
1300 REM Bloque número 15.
1310 REM Fin del programa.
1320 CLS
1330 TI$(0)="¿Desea ejecutar nuevamente el programa?"
1340 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
1350 TI$(2)="en caso contrario, cualquier otra tecla para"
1360 TI$(3)="finalizar la ejecución del programa"
1370 ETIQ=3
1380 GOSUB 2000
1390 GOSUB 2500
1400 IF Z$<>"1" THEN KEY ON ELSE 230
1410 CLS
1420 GOSUB 6000
1430 STOP
1500 REM Subrutinas del programa.

```

```

2000 REM Subrutina número 1.
2010 REM Impresión centrada de las etiquetas del
2020 REM programa, por página.
2030 FOR I=0 TO ETIQ
2040 FOR J=1 TO 3
2050 PRINT
2060 NEXT J
2070 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
2080 NEXT I
2090 RETURN
2500 REM Subrutina número 2.
2510 REM Paralización del avance del programa mientras
2520 REM el usuario lee los mensajes.
2530 Z$=""
2540 WHILE Z$=""
2550 Z$=INKEY$
2560 WEND
2570 RETURN
3000 REM Subrutina número 3.
3010 REM Verificar que el coeficiente de  $X^N$  es
3020 REM mayor que cero.
3030 IF N#(0,N)>0# THEN 3070
3040 FOR I=0 TO N
3050 N#(0,I)=-N#(0,I)
3060 NEXT I
3070 RETURN
3500 REM Subrutina número 4.
3510 REM Examinar si cero es o no cota de las raíces
3520 REM del polinomio.
3530 FOR I=N-1 TO 0 STEP -1
3540 IF N#(0,I)<0# THEN 3590
3550 NEXT I
3560 SUP=0
3570 CONTROL$="CERO"
3580 GOTO 3600
3590 P=I
3600 RETURN
4000 REM Subrutina número 5.
4010 REM Método del mayor de los módulos de los
4020 REM coeficientes negativos de la ecuación.
4030 IF CONTROL$="CERO" THEN 4120
4040 M=0
4050 FOR I=1 TO N
4060 IF N#(0,M)<=N#(0,I) THEN 4080
4070 M=I
4080 NEXT I
4090 G#=ABS(N#(0,M))

```

```

4100 A#=N#(0,N)
4110 SUP=1+INT(1#+(G#/A#)^(1/(N-P)))
4120 RETURN
4500 REM Subrutina número 6.
4510 REM Método de Newton.
4520 FOR J=1 TO N+1
4530 N#(J,N)=N#(0,N)
4540 NEXT J
4550 FOR P=0 TO SUP
4560 GR=0
4570 FOR M=1 TO N
4580 FOR J=N-1 TO GR STEP -1
4590 N#(M,J)=N#(M,J+1)*CDBL(P)+N#(M-1,J)
4600 NEXT J
4610 GR=GR+1
4620 NEXT M
4630 J=0
4640 FOR M=1 TO N+1
4650 IF N#(M,J)<=0# THEN 4690
4660 J=J+1
4670 NEXT M
4680 GOTO 4700
4690 NEXT P
4700 L=P
4710 GOSUB 5000
4720 ETIQ=2
4730 GOSUB 2000
4740 GOSUB 2500
4750 RETURN
5000 REM Subrutina número 7.
5010 REM Mensajes de salida cuando se encuentra la cota.
5020 TI$(0)="Una cota "+COTA$+" de las raíces "+RAICES$
5030 TI$(0)=TI$(0)+" de"
5040 TI$(1)="la ecuación es "+VARIABLE$+"="+STR$(L*(-1)^SW)
5050 TI$(2)="Presione cualquier tecla para continuar"
5060 RETURN
5500 REM Subrutina número 8.
5510 REM Inicialización de las variables del programa
5520 REM para otra ejecución.
5530 ERASE P#,N#
5540 SW=0
5550 COTA$=""
5560 RAICES$=""
5570 VARIABLE$=""
5580 RETURN
6000 REM Subrutina número 9.
6010 REM Fin del programa.

```

6020 END

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido acotadas empleando el programa que acabamos de mostrar.

TABLA 8

ECUACIÓN: $P(x)=0$	RAÍCES VERDADERAS	INTERVALO SEGÚN EL MÉTODO
$x^3 - 5x + 1$	- 2.330, 0.202, 2.128	(- 3,3)
$x^3 - 7x - 7$	- 1.692, - 1.357, 3.049	(- 2,4)
$x^4 - 41x^2 + 400$	- 5, - 4, 4, 5	(- 6,6)
$3x^4 - 40x^3 + 130x^2 - 120x + 27$	0.333, 1, 3, 9	(0,10)
$x^4 - 40x^3 + 390x^2 - 1080x + 729$	1, 3, 9, 27	(0,28)
$x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	- 2,1,3,4	(- 3,5)

Es evidente que las cotas de las raíces que genera esta técnica son similares a las obtenidas a través del método de Newton. Además, al aplicar este procedimiento, no se necesita mayor información que la suministrada por el polinomio en sí, y el mismo, es eficiente en prácticamente todos los casos (salvo donde el método del mayor de los módulos de los coeficientes negativos de la ecuación falla, en el manejo de cantidades muy grandes o pequeñas).

Estos hechos nos permiten considerar que el problema de acotar los ceros reales de una ecuación algebraica de raíces reales, al aplicar el último método desarrollado, se puede considerar razonablemente resuelto.

A continuación, consideraremos la implementación en computadoras del proceso que sigue en nuestro desarrollo teórico, es decir, la separación de los ceros reales de una ecuación algebraica de raíces reales.

3.6.2. MÉTODOS QUE SEPARAN RAÍCES REALES.

Con respecto a estos métodos, en primer término, establecemos que sólo se realizará el desarrollo del método de Sturm, ya que los demás bisección y Rolle , son empleados en la implementación de éste , ya sea en sus aspectos teóricos o en la práctica .

En primer término, presentamos un listado de las variables más importantes que usarán los programas de separación de raíces.

VARIABLES MÁS IMPORTANTES QUE USARÁN LOS PROGRAMAS SOBRE SEPARACIÓN DE RAÍCES REALES.

VARIABLE	SIGNIFICADO
TI\$(5)	Vector alfanumérico donde se guardan las salidas del programa.
ETIQ	Variable numérica (de tipo precisión simple) que representa el número de etiquetas a imprimir, en un momento dado en la ejecución del programa.
N, GR, GRA, M y P	Variables numéricas (de tipo precisión simple) que representan los grados de los polinomios usados dentro del programa.
P#(N), N#(N), D#(N), C#(N-1), POLY#(N), A#(N-2) y M#(N)	Vectores numéricos (de tipo doble precisión), donde se guardan los polinomios original y los obtenidos.
INF, SUP, A# y B#	Variables numéricas (de tipo simple y doble precisión) que contienen, por parejas, respectivamente, los extremos del intervalo que acota a las raíces reales del polinomio.

VARIABLE	SIGNIFICADO
PO# y LI	Variables numéricas (de tipos doble y simple precisión), que contienen, respectivamente, el porcentaje máximo a que se reducirá la longitud de los subintervalos a obtener, y el número máximo de iteraciones a realizar para reducir la longitud de dichos subintervalos.
B#(N,N)	Matriz numérica (de tipo doble precisión) donde se guardan los coeficientes de los polinomios de la sucesión de Sturm relacionada a la ecuación de raíces simples obtenida.
Q#(N+1,N)	Matriz numérica (de tipo doble precisión), donde se calcula la cota buscada por el método de Newton.
LG#	Variable numérica (de tipo doble precisión) que contiene la longitud del intervalo que acota las raíces del polinomio.
E#(1,N)	Matriz numérica (de tipo doble precisión) que contiene, en su fila 0, los extremos de los subintervalos que genera el método, y, en su fila 1, el número de variaciones que presenta la sucesión de Sturm, para el punto en mención.
C y V	Variables numéricas (de tipo precisión simple) donde se cuenta el número de variaciones que va presentando la sucesión de Sturm.
R	Es un interruptor que nos permite determinar cuando uno de los extremos obtenidos es raíz de la ecuación original.
TOL#	Variable numérica (de tipo doble precisión) que guarda el valor calculado de la tolerancia máxima del error en que se puede incurrir al aproximar los intervalos.

VARIABLE	SIGNIFICADO
MA#	Variable numérica (de tipo doble precisión) que representa la longitud mayor de todos los subintervalos calculados en una iteración.
l#(N-1,1)	Matriz numérica (de tipo doble precisión) donde se refina el proceso de separación de raíces realizado por el método de Sturm, al aplicar a cada pareja l#(k,0), l#(k,1), $k = 0, 1, \dots, N - 1$ (que son los extremos de un subintervalo obtenido), el procedimiento de bisección de Bolzano.
Z\$	Variable alfanumérica, es un interruptor, controla el avance del programa, para que el usuario pueda leer las pantallas.

3.6.2.1. MÉTODO DE STURM.

La implementación de un programa que realice la separación de los ceros de una ecuación algebraica de raíces reales, por este método, no es una tarea fácil, como han sido los métodos previamente desarrollados. Esto se debe a que, si no es planteado en una forma adecuada, dicho método puede ser presa, fácilmente, de la acumulación del error por redondeo en los cálculos, lo que haría totalmente inútil el planteamiento.

Esta situación se presenta, básicamente, en los procesos que involucran la división de polinomios, es decir, en la reducción de una ecuación de raíces múltiples a otra que posea las mismas raíces, pero todas simples, y en la construcción de la sucesión de Sturm.

Sin embargo, existen ciertos resultados de la teoría de división de polinomios asociados que nos permiten controlar, parcialmente, esta situación. Dichas reglas, las

mencionamos en el apéndice de teoría de ecuaciones, teorema número **14**, y básicamente, consisten en que cada resto parcial de una división puede ser multiplicado por una constante no nula positiva, sin que se alteren las raíces del dividendo. La constante que requerimos, en nuestro caso, es tal que el término resultante en el cociente sea un entero.

Por otro lado, si los polinomios dados poseen coeficientes decimales, definitivamente que todos los resultados siguientes los contendrán, de modo que también debemos contemplar la opción de transformar la ecuación en otra de raíces iguales, pero con todos sus coeficientes enteros. Para lograr esto, multiplicaremos la ecuación por una potencia de 10, en función del mayor número de decimales que presenten los coeficientes de la ecuación original.

Pero generalmente esto ocasiona el aumento exorbitante (en valor absoluto) de los coeficientes de la ecuación, por lo que se procede a descomponer los mismos con el objeto de eliminar los factores comunes y disminuir, así, la magnitud de los mismos.

Es evidente que este método está sujeto al error por redondeo, de modo que pondremos en el mismo dos controles para impedir que la máquina entre en un ciclo sin fin. El primer control consistirá en acotar el número máximo de iteraciones que realizará el método, mientras que el otro, será acotar la longitud mínima que poseerá cada intervalo que genere el método; dicha longitud estará dada en función del intervalo originalmente dado, como un porcentaje de éste.

Además, las características del método nos indican que sólo puede aplicarse en polinomios que, al suprimir sus raíces múltiples, se reducen a ecuaciones con dos o más raíces reales simples, ya que, de lo contrario, las operaciones de división sintética que involucra no tendrían sentido.

Finalmente, si las raíces de la ecuación son excesivamente próximas (dentro de las limitaciones impuestas por el compilador que estamos usando, dos raíces se pueden considerar como muy próximas entre sí, si su diferencia en valor absoluto, es menor que 10^{-7}), el error por redondeo afectará definitivamente el proceso de supresión de raíces múltiples, por lo que el método podría llegar a resultados totalmente errados. Esta situación es, definitivamente, una falla del método, que sólo se puede salvar, parcialmente, con el empleo de un compilador más poderoso.

A continuación, pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO DE STURM

```

10 REM Programa número 5.
20 REM Cálculo de los subintervalos que separan
30 REM las raíces reales de una ecuación algebraica
40 REM de coeficientes reales, empleando el método de
50 REM Sturm.
60 REM Bloque número 1.
70 REM Presentación del programa.
80 KEY OFF
90 CLS
100 CLEAR
110 DIM TI$(5)
120 TI$(0)="Cálculo de los subintervalos que separan"
130 TI$(1)="las raíces reales de una ecuación algebraica"
140 TI$(2)="de coeficientes reales, empleando"
150 TI$(3)="el método de Sturm"
160 TI$(4)="Presione cualquier tecla para continuar"
170 ETIQ=4
180 GOSUB 4500
190 GOSUB 5000
200 REM Bloque número 2.
210 REM Carga del grado del polinomio.
220 CLS
230 CLEAR
240 DIM TI$(5)
250 TI$(0)="Por favor, introduzca el grado del"
260 TI$(1)="polinomio (N) (El método requiere que"
```

```

270 TI$(2)="el polinomio sea de grado mayor o igual"
280 TI$(3)="que dos o no funcionará correctamente)"
290 TI$(4)="y pulse ``Enter'' para continuar"
300 ETIQ=4
310 GOSUB 4500
320 PRINT
330 PRINT
340 INPUT "N=",N
350 IF N>=2 AND N=FIX(N) THEN 430
360 CLS
370 TI$(0)="La información introducida es incorrecta"
380 TI$(1)="Presione cualquier tecla para continuar"
390 ETIQ=1
400 GOSUB 4500
410 GOSUB 5000
420 GOTO 220
430 REM Bloque número 3.
440 REM Dimensiones de los vectores y matrices iniciales.
450 DIM B#(N,N),C#(N-1),D#(N),E#(1,N),I#(N-1,1)
460 DIM M#(N),N#(N),P#(N)
470 GR=N
480 REM Bloque número 4.
490 REM Carga de los coeficientes del polinomio
500 REM y cálculo del mayor número de decimales presente
510 REM en dichos coeficientes.
520 CLS
530 TI$(0)="Introduzca los coeficientes del"
540 TI$(1)="polinomio, en orden descendente y pulse"
550 TI$(2)="``Enter'' para continuar, en cada caso"
560 TI$(3)="Presione cualquier tecla para continuar"
570 ETIQ=3
580 GOSUB 4500
590 GOSUB 5000
600 CLS
610 FOR I=N TO 0 STEP -1
620 PRINT "Introduzca el coeficiente de X^";I;
630 INPUT "=",P#(I)
640 VAR$=STR$(P#(I))
650 LONG=LEN(VAR$)
660 FOR J=1 TO LONG
670 CARC$=MID$(VAR$,J,1)
680 IF CARC$="." THEN POT=LONG-J ELSE NEXT J
690 IF I=N THEN NU=POT
700 IF NU>POT THEN 720
710 NU=POT
720 PRINT
730 PRINT

```

```

740 NEXT I
750 IF P#(N)<>0# AND P#(0)<>0# THEN 850
760 CLS
770 TI$(0)="El polinomio introducido puede simplificarse,"
780 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+", "
790 TI$(2)="ni el de X^ 0, pueden ser nulos"
800 TI$(3)="Presione cualquier tecla para continuar"
810 ETIQ=3
820 GOSUB 4500
830 GOSUB 5000
840 GOTO 220
850 REM Bloque número 5.
860 REM Carga de los extremos del intervalo que acota
870 REM las raíces de la ecuación.
880 CLS
890 N=GR
900 TI$(0)="Introduzca los extremos"
910 TI$(1)="del intervalo que acota a las raíces"
920 TI$(2)="del polinomio"
930 ETIQ=2
940 GOSUB 4500
950 PRINT
960 PRINT
970 INPUT "Extremo inferior (INF) =",INF
980 PRINT
990 PRINT
1000 INPUT "Extremo superior (SUP) =",SUP
1010 IF INF<SUP THEN 1090
1020 CLS
1030 TI$(0)="La información introducida es incorrecta"
1040 TI$(1)="Presione cualquier tecla para continuar"
1050 ETIQ=1
1060 GOSUB 4500
1070 GOSUB 5000
1080 GOTO 850
1090 REM Bloque número 6.
1100 REM Carga de la tolerancia y número máximo de
1110 REM iteraciones que realizará el método.
1120 CLS
1130 TI$(0)="Introduzca la longitud"
1140 TI$(1)="máxima de los subintervalos que acotarán"
1150 TI$(2)="a las raíces del polinomio"
1160 ETIQ=2
1170 GOSUB 4500
1180 PRINT
1190 PRINT
1200 INPUT "Longitud ( $10^{-7} < PO < 10^0$ ) (PO) =  $10^$  ",PO#

```

```

1210 CLS
1220 TI$(0)="Introduzca el número máximo de iteraciones"
1230 TI$(1)="a realizar al reducir la longitud de los"
1240 TI$(2)="subintervalos que acotan las raíces"
1250 ETIQ=2
1260 GOSUB 4500
1270 PRINT
1280 PRINT
1290 INPUT "Número de iteraciones (LI) =",LI
1300 PO#=CDBL(10^PO#)
1310 IF PO#<=CDBL(10^-7) OR PO#>=1# THEN 1340
1320 IF LI<=0 THEN 1340
1330 IF LI=FIX(LI) THEN 1410
1340 CLS
1350 TI$(0)="La información introducida es incorrecta"
1360 TI$(1)="Presione cualquier tecla para continuar"
1370 ETIQ=1
1380 GOSUB 4500
1390 GOSUB 5000
1400 GOTO 1090
1410 REM Bloque número 7.
1420 REM Inicialización de los polinomios de trabajo y
1430 REM supresión de los decimales (si los hay), de los
1440 REM coeficientes del polinomio.
1450 CLS
1460 IF NU=0 THEN 1500
1470 FOR I=0 TO N
1480 P#(I)=P#(I)*CDBL(10^NU)
1490 NEXT I
1500 FOR I=0 TO N
1510 N#(I)=P#(I)
1520 NEXT I
1530 REM Bloque número 8.
1540 REM Simplificar (si es posible), los coeficientes
1550 REM del polinomio.
1560 FOR I=0 TO N
1570 M#(I)=N#(I)
1580 NEXT I
1590 M=N
1600 GOSUB 5500
1610 GOSUB 6000
1620 FOR I=0 TO N
1630 N#(I)=M#(I)
1640 NEXT I
1650 REM Bloque número 9.
1660 REM Calcular y simplificar la derivada del polinomio.
1670 M=N-1

```

```
1680 FOR I=0 TO M
1690 M#(I)=CDBL(I+1)*N#(I+1)
1700 NEXT I
1710 GOSUB 5500
1720 GOSUB 6000
1730 REM Bloque número 10.
1740 REM Reducir la ecuación original a una con raíces
1750 REM simples.
1760 FOR I=0 TO M
1770 D#(I)=M#(I)
1780 NEXT I
1790 P=N
1800 GOSUB 7000
1810 FOR K=P-2 TO 0 STEP -1
1820 IF N#(K)<>0 THEN 1920
1830 NEXT K
1840 GOSUB 7500
1850 P=N
1860 GOSUB 7000
1870 FOR I=P-M TO 0 STEP -1
1880 N#(I)=C#(I)
1890 NEXT I
1900 N=P-M
1910 GOTO 1960
1920 IF P=2 THEN 1950
1930 GOSUB 8000
1940 GOTO 1800
1950 GOSUB 7500
1960 REM Bloque número 11.
1970 REM Cálculo de los polinomios de Sturm.
1980 FOR J=0 TO N
1990 B#(0,J)=N#(J)
2000 NEXT J
2010 L=1
2020 FOR J=N-1 TO 0 STEP -1
2030 B#(L,J)=N#(J+1)*CDBL(J+1)
2040 M#(J)=B#(L,J)
2050 NEXT J
2060 M=N-1
2070 GOSUB 5500
2080 GOSUB 6000
2090 FOR I=0 TO M
2100 D#(I)=M#(I)
2110 NEXT I
2120 P=N
2130 GOSUB 7000
2140 GOSUB 8000
```

```
2150 L=L+1
2160 FOR J=P-1 TO 0 STEP -1
2170 D#(J)=-D#(J)
2180 B#(L,J)=D#(J)
2190 NEXT J
2200 IF L=N THEN 2220
2210 GOTO 2130
2220 REM Bloque número 12.
2230 REM Cálculo de los subintervalos que separan las
2240 REM raíces de la ecuación.
2250 A#=CDBL(INF)
2260 B#=CDBL(SUP)
2270 LG#=B#-A#
2280 E#(0,0)=A#
2290 E#(0,N)=B#
2300 R=0
2310 GOSUB 8500
2320 IF C=N THEN 2380
2330 GOSUB 12500
2340 ETIQ=3
2350 GOSUB 4500
2360 GOSUB 5000
2370 GOTO 3560
2380 E#(1,0)=CDBL(C)
2390 A#=B#
2400 R=R+1
2410 GOSUB 8500
2420 E#(1,N)=CDBL(C)
2430 R=R+1
2440 I=0
2450 J=1
2460 K=N
2470 GOSUB 9500
2480 IF F=0 THEN 2550
2490 D=0
2500 FOR P=0 TO N-1
2510 V=CINT(E#(1,P)-E#(1,P+1))
2520 IF V=0 THEN 2560
2530 IF V=1 THEN 2600
2540 IF V>1 THEN 2700
2550 GOTO 2800
2560 IF D<>0 THEN 2580
2570 T=P
2580 D=D+1
2590 GOTO 2680
2600 IF D=0 THEN 2680
2610 GOSUB 10000
```

```
2620 I=T+1
2630 J=T+2
2640 K=N
2650 GOSUB 9500
2660 IF F=0 THEN 2490
2670 GOTO 2490
2680 NEXT P
2690 GOTO 2800
2700 IF D=0 THEN 2730
2710 GOSUB 10000
2720 P=T
2730 E#(1,P+V)=E#(1,P+1)
2740 E#(0,P+V)=E#(0,P+1)
2750 I=P
2760 J=P+1
2770 K=P+V
2780 GOSUB 9500
2790 GOTO 2490
2800 FOR J=0 TO N-1
2810 I#(J,0)=E#(0,J)
2820 I#(J,1)=E#(0,J+1)
2830 NEXT J
2840 TOL#=LG#*PO#
2850 FOR J=0 TO N-1
2860 A#=I#(J,0)
2870 GOSUB 10500
2880 B#=S#
2890 A#=I#(J,1)
2900 GOSUB 10500
2910 IF B#*S#>=0# THEN 2940
2920 NEXT J
2930 GOTO 2990
2940 GOSUB 12500
2950 ETIQ=3
2960 GOSUB 4500
2970 GOSUB 5000
2980 GOTO 3560
2990 MA#=I#(0,1)-I#(0,0)
3000 FOR I=0 TO N-1
3010 IF MA#>I#(I,1)-I#(I,0) THEN 3030
3020 MA#=I#(I,1)-I#(I,0)
3030 NEXT I
3040 FOR J=1 TO LI
3050 FOR I=0 TO N-1
3060 A#=I#(I,0)
3070 GOSUB 10500
3080 B#=S#
```



```

3090 A#=(I#(I,0)+I#(I,1))/2#
3100 GOSUB 10500
3110 IF S#=0# THEN 3150
3120 IF B#*S#<0# THEN 3180
3130 I#(I,0)=A#
3140 GOTO 3190
3150 I#(I,0)=A#-MA#/3#
3160 I#(I,1)=A#+MA#/3#
3170 GOTO 3190
3180 I#(I,1)=A#
3190 NEXT I
3200 GOSUB 11000
3210 IF MA#<TOL# THEN 3370
3220 FOR I=0 TO N-1
3230 IF I#(I,0)<>I#(I,1) THEN 3260
3240 I#(I,0)=I#(I,0)-MA#/3#
3250 I#(I,1)=I#(I,1)+MA#/3#
3260 NEXT I
3270 NEXT J
3280 REM Bloque número 13.
3290 REM Salida cuando no se encuentran los subintervalos
3300 REM en el número establecido de iteraciones.
3310 GOSUB 11500
3320 ETIQ=3
3330 CLS
3340 GOSUB 4500
3350 GOSUB 5000
3360 GOTO 3560
3370 REM Bloque número 14.
3380 REM Salida cuando se encuentran los subintervalos
3390 REM en el número establecido de iteraciones.
3400 GOSUB 12000
3410 ETIQ=2
3420 CLS
3430 GOSUB 4500
3440 GOSUB 5000
3450 FOR K=0 TO N-1
3460 CLS
3470 TI$(0)="En el subintervalo"
3480 TI$(1)="("+STR$(I#(K,0))+", "+STR$(I#(K,1))+")"
3490 TI$(2)="se encuentra separada la raíz"
3500 TI$(3)="número"+STR$(K+1)+" de la ecuación."
3510 TI$(4)="Presione cualquier tecla para continuar"
3520 ETIQ=4
3530 GOSUB 4500
3540 GOSUB 5000
3550 NEXT K

```

```
3560 REM Bloque número 15.
3570 REM Fin del programa.
3580 CLS
3590 TI$(0)="¿Desea ejecutar nuevamente el programa?"
3600 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
3610 TI$(2)="en caso contrario, cualquier otra tecla para"
3620 TI$(3)="finalizar la ejecución del programa"
3630 ETIQ=3
3640 GOSUB 4500
3650 GOSUB 5000
3660 IF Z$<>"1" THEN 3770
3670 CLS
3680 TI$(0)="¿Desea realizar los cálculos sobre el mismo"
3690 TI$(1)="polinomio? Si es así, pulse ``1'' para"
3700 TI$(2)="hacerlo, y, en caso contrario, cualquier otra"
3710 TI$(3)="tecla para cargar todos los datos nuevamente"
3720 ETIQ=3
3730 GOSUB 4500
3740 GOSUB 5000
3750 IF Z$="1" THEN 850 ELSE GOSUB 13000
3760 GOTO 200
3770 KEY ON
3780 CLS
3790 GOSUB 13500
3800 STOP
4000 REM Subrutinas del programa.
4500 REM Subrutina número 1.
4510 REM Impresión centrada de las etiquetas del
4520 REM programa, por página.
4530 FOR I=0 TO ETIQ
4540 FOR J=1 TO 3
4550 PRINT
4560 NEXT J
4570 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
4580 NEXT I
4590 RETURN
5000 REM Subrutina número 2.
5010 REM Paralización del avance del programa mientras
5020 REM el usuario lee los mensajes.
5030 Z$=""
5040 WHILE Z$=""
5050 Z$=INKEY$
5060 WEND
5070 RETURN
5500 REM Subrutina número 3.
5510 REM Determinar el menor coeficiente (en valor
5520 REM absoluto) no nulo de un polinomio.
```

```

5530 FOR I=0 TO M
5540 IF ABS(M#(I))=0# THEN NEXT I
5550 ME#=ABS(M#(I))
5560 FOR I=0 TO M
5570 IF M#(I)=0# THEN 5590 ELSE IF ME#<=ABS(M#(I)) THEN 5590
5580 ME#=ABS(M#(I))
5590 NEXT I
5600 RETURN
6000 REM Subrutina número 4.
6010 REM Descomponer el menor coeficiente del polinomio
6020 REM en sus factores primos.
6030 B#=2#
6040 GOTO 6060
6050 GOSUB 6500
6060 IF ME#=1# THEN 6170
6070 IF ME#/B#=INT(ME#/B#) THEN 6050 ELSE B#=3#
6080 C#=INT(SQR(ME#)+1#)
6090 IF B#<C# THEN 6130
6100 B#=ME#
6110 GOSUB 6500
6120 GOTO 6170
6130 IF ME#/B#=INT(ME#/B#) THEN 6150 ELSE B#=B#+2#
6140 GOTO 6080
6150 GOSUB 6500
6160 GOTO 6080
6170 RETURN
6500 REM Subrutina número 5.
6510 REM Simplificar los coeficientes del polinomio
6520 REM si es posible.
6530 ME#=ME#/B#
6540 FOR I=0 TO M
6550 IF M#(I)/B#<>INT(M#(I)/B#) THEN 6600
6560 IF I<>M THEN NEXT I
6570 FOR I=0 TO M
6580 M#(I)=M#(I)/B#
6590 NEXT I
6600 RETURN
7000 REM Subrutina número 6.
7010 REM División de polinomios con el algoritmo
7020 REM de los enteros.
7030 FOR I=0 TO M+1
7040 N#(I)=N#(I)*ABS(D#(M))
7050 NEXT I
7060 FOR I=P-M TO 0 STEP -1
7070 C#(I)=N#(I+M)/D#(M)
7080 FOR J=0 TO M
7090 N#(I+J)=N#(I+J)-C#(I)*D#(J)

```

```
7100 NEXT J
7110 IF I=0 THEN 7160
7120 FOR J=0 TO M
7130 N#(J)=N#(J)*ABS(D#(M))
7140 NEXT J
7150 NEXT I
7160 RETURN
7500 REM Subrutina número 7.
7510 REM Simplificar nuevamente el polinomio original
7520 REM y preparar la división por su M.C.D.
7530 FOR J=0 TO N
7540 N#(J)=P#(J)
7550 M#(J)=P#(J)
7560 NEXT J
7570 L=M
7580 M=N
7590 GOSUB 5500
7600 GOSUB 6000
7610 FOR J=0 TO N
7620 N#(J)=M#(J)
7630 NEXT J
7640 M=L
7650 RETURN
8000 REM Subrutina número 8.
8010 REM Preparar el cálculo del siguiente candidato a
8020 REM M.C.D. del polinomio original.
8030 FOR I=P-2 TO 0 STEP -1
8040 M#(I)=N#(I)
8050 NEXT I
8060 FOR J=P-2 TO 0 STEP -1
8070 IF M#(J)<>0# THEN 8090
8080 NEXT J
8090 M=J
8100 GOSUB 5500
8110 GOSUB 6000
8120 FOR I=P-1 TO 0 STEP -1
8130 N#(I)=D#(I)
8140 NEXT I
8150 FOR I=P-2 TO 0 STEP -1
8160 D#(I)=M#(I)
8170 NEXT I
8180 P=P-1
8190 RETURN
8500 REM Subrutina número 9.
8510 REM Cálculo del número de variaciones de la sucesión
8520 REM de Sturm en el valor A.
8530 I=0
```

```

8540 FOR J=N TO 1 STEP -1
8550 N#(J)=B#(I,J)
8560 FOR M=J-1 TO 0 STEP -1
8570 N#(M)=A#*N#(M+1)+B#(I,M)
8580 NEXT M
8590 M#(I)=N#(0)
8600 I=I+1
8610 NEXT J
8620 IF M#(0)<>0# THEN 8830
8630 IF R<>0 THEN 8730
8640 CLS
8650 TI$(0)="El extremo inferior introducido "+STR$(A#)
8660 TI$(1)="es una raíz de la ecuación, ejecute"
8670 TI$(2)="nuevamente el programa con mejores intervalos"
8680 TI$(3)="Presione cualquier tecla para continuar"
8690 ETIQ=3
8700 GOSUB 4500
8710 GOSUB 5000
8720 GOTO 3770
8730 IF R<>1 THEN 8830
8740 CLS
8750 TI$(0)="El extremo superior introducido "+STR$(B#)
8760 TI$(1)="es una raíz de la ecuación, ejecute"
8770 TI$(2)="nuevamente el programa con mejores intervalos"
8780 TI$(3)="Presione cualquier tecla para continuar"
8790 ETIQ=3
8800 GOSUB 4500
8810 GOSUB 5000
8820 GOTO 3770
8830 M#(N)=B#(N,0)
8840 C=0
8850 FOR J=0 TO N-1
8860 IF M#(J)=0# THEN 8940
8870 K=J+1
8880 IF M#(K)=0# THEN 8980
8890 IF M#(J)>0# THEN IF M#(K)>0# THEN 8920
8900 IF M#(J)<0# THEN IF M#(K)<0# THEN 8920
8910 C=C+1
8920 IF K=N THEN 9000
8930 GOTO 8950
8940 IF J=0 THEN 8960
8950 J=K-1
8960 NEXT J
8970 GOTO 9000
8980 K=K+1
8990 IF K<=N THEN 8880
9000 RETURN

```

```

9500 REM Subrutina número 10.
9510 REM Verificar que el número de variaciones sea
9520 REM el correcto en los intervalos, y si es necesario,
9530 REM se reeligen los extremos del subintervalo.
9540 V=CINT(E#(1,I)-E#(1,K))
9550 F=0
9560 IF V=1 THEN 9840
9570 F=1
9580 L#=E#(0,K)-E#(0,I)
9590 LO#=L#/CDBL(V)
9600 E=J
9610 E#(0,E)=E#(0,E-1)+LO#
9620 A#=E#(0,E)
9630 GOSUB 8500
9640 E#(1,E)=CDBL(C)
9650 E=E+1
9660 IF E<=K-1 THEN 9610
9670 U=0
9680 A#=E#(0,U)
9690 GOSUB 10500
9700 IF S#<>0# THEN 9750
9710 E#(0,U)=E#(0,U)+(E#(0,U+1)-E#(0,U))/3#
9720 A#=E#(0,U)
9730 GOSUB 8500
9740 E#(1,U)=CDBL(C)
9750 U=U+1
9760 IF U<=N-1 THEN 9680
9770 A#=E#(0,N)
9780 GOSUB 10500
9790 IF S#<>0# THEN 9840
9800 E#(0,N)=E#(0,N)+1#/3#
9810 A#=E#(0,N)
9820 GOSUB 8500
9830 E#(1,N)=CDBL(C)
9840 RETURN
10000 REM Subrutina número 11.
10010 REM Substituye los extremos de los subintervalos
10020 REM que no poseen raíces, por otros que si las
10030 REM poseen.
10040 E#(1,T)=E#(1,P)
10050 E#(1,T+1)=E#(1,P+1)
10060 E#(0,T)=E#(0,P)
10070 E#(0,T+1)=E#(0,P+1)
10080 RETURN
10500 REM Subrutina número 12.
10510 REM Evaluación del polinomio reducido en el valor A
10520 REM para reducir la longitud de los subintervalos que

```

```

10530 REM separan a las raíces.
10540 N#(N)=B#(0,N)
10550 FOR M=N-1 TO 0 STEP -1
10560 N#(M)=A#*N#(M+1)+B#(0,M)
10570 NEXT M
10580 S#=N#(M+1)
10590 RETURN
11000 REM Subrutina número 13.
11010 REM Cálculo de la longitud del mayor de los
11020 REM subintervalos que separan las raíces.
11030 MA#=0#
11040 FOR I=0 TO N-1
11050 LO#=I#(I,1)-I#(I,0)
11060 IF MA#>=LO# THEN 11080
11070 MA#=LO#
11080 NEXT I
11090 RETURN
11500 REM Subrutina número 14.
11510 REM Mensajes de salida cuando no se encuentran
11520 REM los subintervalos.
11530 TI$(0)="Los subintervalos que acotan a las raíces"
11540 TI$(1)="simples de la ecuación, no se han podido"
11550 TI$(2)="calcular, en "+STR$(LI)+" iteraciones"
11560 TI$(3)="Presione cualquier tecla para continuar"
11570 RETURN
12000 REM Subrutina número 15.
12010 REM Mensajes de salida cuando se encuentran los
12020 REM los subintervalos.
12030 TI$(0)="Los subintervalos que acotan a las raíces"
12040 TI$(1)="simples de la ecuación, han sido calculados"
12050 TI$(2)="Presione cualquier tecla para continuar"
12060 RETURN
12500 REM Subrutina número 16.
12510 REM Mensajes de salida cuando la ecuación posee
12520 REM raíces complejas.
12530 TI$(0)="Los intervalos introducidos deben ampliarse"
12540 TI$(1)="o la ecuación posee raíces complejas, por lo"
12550 TI$(2)="que el método no le es aplicable"
12560 TI$(3)="Presione cualquier tecla para continuar"
12570 RETURN
13000 REM Subrutina número 17.
13010 REM Inicialización de las variables del programa
13020 REM para otra ejecución.
13030 ERASE B#,C#,D#,E#,I#,M#,N#,P#
13040 RETURN
13500 REM Subrutina número 18.
13510 REM Fin del programa.

```

13520 END

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido separadas empleando el programa que acabamos de mostrar.

TABLA 9

ECUACIÓN: $P(x)=0$	RAÍCES VERDADERAS	INTERVALO SEGÚN EL MÉTODO
$x^3 - 5x + 1$	- 2.3300587 0.20163968 2.1284191	(2.330186631944436, 2.330035927854929) (0.201491367669762, 0.2016420717592682) (2.128393856095689, 2.128544560185195)
$x^3 - 7x - 7$	- 1.6920215 - 1.3568959 3.0489173	(- 1.692030164930547, - 1.692020745924953) (- 1.356901945891195, - 1.356892526885601) (3.048894434799392, 3.049045138888898)
$x^4 - 41x^2 + 400$	- 5.0000000 - 4.0000000 4.0000000 5.0000000	(- 5.000019073486328, - 4.999971389770508) (- 4.000091552734375, - 3.999996185302734) (3.999835252761841, 4.000028967857361) (4.999986290931702, 5.000180006027222)
$3x^4 - 40x^3 + 130x^2 - 120x + 27$	0.33333333 1.0000000 3.0000000 9.0000000	(0.3333250681559245, 0.3333409627278646) (0.9999910990397136, 1.000022888183594) (2.99994945526123, 3.000116348266602) (8.999919891357422, 9.000086784362793)
$x^4 - 40x^3 + 390x^2 - 1080x + 729$	1.0000000 3.0000000 9.0000000 27.000000	(0.9999910990397136, 1.000022888183594) (2.999941507975261, 3.000005086263021) (8.999896049499512, 9.000062942504883) (26.99997425079346, 27.00014114379883)
$x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	-2.0000000 1.0000000 3.0000000 4.0000000	(-2.000045776367188, -1.999855041503906) (0.9999946311668115, 1.000001695421007) (2.999998022008826, 3.000005086263021) (3.999978524667246, 4.000006781684029)

Hay varias situaciones que debemos aclarar dentro del proceso que hemos realizado, para evitar malas interpretaciones de la acción del método de Sturm.

En primer término, todas las ecuaciones que se muestran en la tabla 9 se resolvieron empleando un juego uniforme de parámetros de control. Así, el intervalo que acotaría, en principio, las raíces de las ecuaciones fue $(-100,100)$ —por lo desarrollado en la sección referente a acotación de raíces, sabemos que el tamaño de este intervalo es excesivamente grande, con respecto a la ubicación de las raíces de la ecuación—. Además, todas las raíces fueron calculadas empleando como longitud máxima de los subintervalos que acotarán a las raíces de la ecuación, el valor mínimo que puede asumir esta variable dentro del programa, es decir, un error $\varepsilon = 0.0001 = 10^{-4}$, o sea, una exactitud en el cálculo muy elevada. Por otro lado se permitió que el método iterara un número elevado de veces (100), de modo que tuviera oportunidad de separar las raíces de las ecuaciones con la precisión pedida.

Los resultados que generó la aplicación del método fueron —en todos los casos— satisfactorios con respecto a los valores que consideramos exactos (los que surgen al aplicar el programa Eureka versión 1.0 sobre las mismas ecuaciones), de modo que podemos considerar que el programa es eficiente.

En otro orden de cosas, debe observarse que el programa del método de Sturm sólo genera intervalos que acotan a las raíces simples de la ecuación introducida, y el mismo no contempla un procedimiento dentro de él que nos permita determinar la multiplicidad de cada raíz, dicha situación se tratará de corregir en la etapa de aproximación de raíces, ya que sólo conociendo con exactitud la raíz, se puede tratar de establecer su multiplicidad en base al teorema número 4 del apéndice de teoría de ecuaciones. Esta situación se evidencia en la última ecuación de la tabla 9.

Finalmente, debemos observar que antes de la tabla 9, sólo se habían considerado las raíces en forma aproximada se consideraba solamente 3 decimales en ellas , ya que para los efectos de acotar las raíces, esto es suficiente. Sin embargo, desde la tabla 9 en adelante, incluiremos todas las cifras decimales que generen los programas ya que esto es necesario para separar y aproximar correctamente las raíces de las ecuaciones.

A continuación, consideraremos la implementación en computadora del método de Sturm con el método integrado de acotación, en un sólo programa, que denominaremos método mixto de separación: método mixto de acotación-Sturm.

3.6.2.2. MÉTODO MIXTO DE SEPARACIÓN: MÉTODO MIXTO DE ACOTACIÓN-STURM.

La implementación de un programa que reuna el método de Sturm con el método mixto de acotación que desarrollamos con anterioridad, no es excesivamente más difícil que la implementación individual de cada método, de modo que sus limitaciones, serán, básicamente, la unión de las limitaciones de ambos métodos.

A continuación, pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO MIXTO DE SEPARACIÓN:

MÉTODO MIXTO DE ACOTACIÓN-STURM

```
10 REM Programa número 6.
20 REM Cálculo de los subintervalos que separan
30 REM las raíces reales de una ecuación algebraica
40 REM de coeficientes reales, empleando el método
50 REM mixto de acotación-Sturm.
```

```

60 REM Bloque número 1.
70 REM Presentación del programa.
80 KEY OFF
90 CLS
100 CLEAR
110 DIM TI$(5)
120 TI$(0)="Cálculo de los subintervalos que separan"
130 TI$(1)="las raíces reales de una ecuación algebraica"
140 TI$(2)="de coeficientes reales, empleando"
150 TI$(3)="el método mixto de acotación-Sturm"
160 TI$(4)="Presione cualquier tecla para continuar"
170 ETIQ=4
180 GOSUB 5000
190 GOSUB 5500
200 REM Bloque número 2.
210 REM Carga del grado del polinomio.
220 CLS
230 CLEAR
240 DIM TI$(5)
250 TI$(0)="Por favor, introduzca el grado del"
260 TI$(1)="polinomio (N) (El método requiere que"
270 TI$(2)="el polinomio sea de grado mayor o igual"
280 TI$(3)="que dos o no funcionará correctamente)"
290 TI$(4)="y pulse ``Enter'' para continuar"
300 ETIQ=4
310 GOSUB 5000
320 PRINT
330 PRINT
340 INPUT "N=",N
350 IF N>=2 AND N=FIX(N) THEN 430
360 CLS
370 TI$(0)="La información introducida es incorrecta"
380 TI$(1)="Presione cualquier tecla para continuar"
390 ETIQ=1
400 GOSUB 5000
410 GOSUB 5500
420 GOTO 220
430 REM Bloque número 3.
440 REM Dimensiones de los vectores y matrices iniciales.
450 DIM B#(N,N),C#(N-1),D#(N),E#(1,N),I#(N-1,1)
460 DIM M#(N),N#(N),P#(N),POLY#(N),Q#(N+1,N)
470 GR=N
480 REM Bloque número 4.
490 REM Carga de los coeficientes del polinomio
500 REM y cálculo del mayor número de decimales presente
510 REM en dichos coeficientes.
520 CLS

```

```

530 TI$(0)="Introduzca los coeficientes del"
540 TI$(1)="polinomio, en orden descendente y pulse"
550 TI$(2)="`Enter'" para continuar, en cada caso"
560 TI$(3)="Presione cualquier tecla para continuar"
570 ETIQ=3
580 GOSUB 5000
590 GOSUB 5500
600 CLS
610 FOR I=N TO 0 STEP -1
620 PRINT "Introduzca el coeficiente de X^";I;
630 INPUT "=",POLY#(I)
640 VAR$=STR$(POLY#(I))
650 LONG=LEN(VAR$)
660 FOR J=1 TO LONG
670 CARC$=MID$(VAR$,J,1)
680 IF CARC$="." THEN POT=LONG-J ELSE NEXT J
690 IF I=N THEN NU=POT
700 IF NU>POT THEN 730
710 NU=POT
720 P#(I)=POLY#(I)
730 PRINT
740 PRINT
750 NEXT I
760 IF P#(N)<>0# AND P#(0)<>0# THEN 860
770 CLS
780 TI$(0)="El polinomio introducido puede simplificarse,"
790 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+" ,"
800 TI$(2)="ni el de X^ 0, pueden ser nulos"
810 TI$(3)="Presione cualquier tecla para continuar"
820 ETIQ=3
830 GOSUB 5000
840 GOSUB 5500
850 GOTO 220
860 REM Bloque número 5.
870 REM Carga de la tolerancia y número máximo de
880 REM iteraciones que realizará el método.
890 CLS
900 N=GR
910 TI$(0)="Introduzca la longitud"
920 TI$(1)="máxima de los subintervalos que acotarán"
930 TI$(2)="a las raíces del polinomio"
940 ETIQ=2
950 GOSUB 5000
960 PRINT
970 PRINT
980 INPUT "Longitud ( $10^{-7} < PO < 1$ ) ( $PO = 10^$  ",PO#
990 CLS

```

```

1000 TI$(0)="Introduzca el número máximo de iteraciones"
1010 TI$(1)="a realizar al reducir la longitud de los"
1020 TI$(2)="subintervalos que acotan las raíces"
1030 ETIQ=2
1040 GOSUB 5000
1050 PRINT
1060 PRINT
1070 INPUT "Número de iteraciones (LI) =",LI
1080 PO#=CDBL(10^PO#)
1090 IF PO#<=CDBL(10^-7) OR PO#>=1# THEN 1120
1100 IF LI<=0 THEN 1120
1110 IF LI=FIX(LI) THEN 1190
1120 CLS
1130 TI$(0)="La información introducida es incorrecta"
1140 TI$(1)="Presione cualquier tecla para continuar"
1150 ETIQ=1
1160 GOSUB 5000
1170 GOSUB 5500
1180 GOTO 860
1190 REM Bloque número 6.
1200 REM Inicialización del polinomio de trabajo.
1210 FOR I=0 TO N
1220 Q#(0,I)=CDBL((-1)^I)*POLY#(I)
1230 NEXT I
1240 CLS
1250 REM Bloque número 7.
1260 REM Verificar que el coeficiente de X^N es
1270 REM mayor que cero.
1280 GOSUB 6000
1290 REM Bloque número 8.
1300 REM Examinar si cero es o no cota inferior de las
1310 REM raíces del polinomio.
1320 CONTROL$=""
1330 GOSUB 6500
1340 REM Bloque número 9.
1350 REM Cálculo de una cota inferior de las raíces del
1360 REM polinomio por el método del mayor de los
1370 REM módulos de los coeficientes negativos de la
1380 REM ecuación.
1390 GOSUB 7000
1400 REM Bloque número 10.
1410 REM Refinamiento de la cota inferior de las raíces
1420 REM del polinomio por el método de Newton.
1430 SW=1
1440 GOSUB 7500
1450 REM Bloque número 11.
1460 REM Reinicialización del polinomio.

```

```
1470 FOR I=0 TO N
1480 Q#(0,I)=POLY#(I)
1490 NEXT I
1500 REM Bloque número 12.
1510 REM Verificar que el coeficiente de  $X^N$  es
1520 REM mayor que cero.
1530 GOSUB 6000
1540 REM Bloque número 13.
1550 REM Examinar si cero es o no cota superior de las
1560 REM raíces del polinomio.
1570 CONTROL$=""
1580 GOSUB 6500
1590 REM Bloque número 14.
1600 REM Cálculo de una cota superior de las raíces del
1610 REM polinomio por el método del mayor de los
1620 REM módulos de los coeficientes negativos de la
1630 REM ecuación.
1640 GOSUB 7000
1650 REM Bloque número 15.
1660 REM Refinamiento de la cota superior de las raíces
1670 REM del polinomio por el método de Newton.
1680 SW=0
1690 GOSUB 7500
1700 REM Bloque número 16.
1710 REM Inicialización de los polinomios de trabajo y
1720 REM supresión de los decimales (si los hay), de los
1730 REM coeficientes del polinomio.
1740 IF NU=0 THEN 1780
1750 FOR I=0 TO N
1760 P#(I)=POLY#(I)*CDBL(10^NU)
1770 NEXT I
1780 FOR I=0 TO N
1790 N#(I)=P#(I)
1800 NEXT I
1810 REM Bloque número 17.
1820 REM Simplificar (si es posible), los coeficientes
1830 REM del polinomio.
1840 FOR I=0 TO N
1850 M#(I)=N#(I)
1860 NEXT I
1870 M=N
1880 GOSUB 8000

1890 GOSUB 8500
1900 FOR I=0 TO N
1910 N#(I)=M#(I)
1920 NEXT I
```

```
1930 REM Bloque número 18.
1940 REM Calcular y simplificar la derivada del polinomio.
1950 M=N-1
1960 FOR I=0 TO M
1970 M#(I)=CDBL(I+1)*N#(I+1)
1980 NEXT I
1990 GOSUB 8000
2000 GOSUB 8500
2010 REM Bloque número 19.
2020 REM Reducir la ecuación original a una con raíces
2030 REM simples.
2040 FOR I=0 TO M
2050 D#(I)=M#(I)
2060 NEXT I
2070 P=N
2080 GOSUB 9500
2090 FOR K=P-2 TO 0 STEP -1
2100 IF N#(K)<>0# THEN 2200
2110 NEXT K
2120 GOSUB 10000
2130 P=N
2140 GOSUB 9500
2150 FOR I=P-M TO 0 STEP -1
2160 N#(I)=C#(I)
2170 NEXT I
2180 N=P-M
2190 GOTO 2240
2200 IF P=2 THEN 2230
2210 GOSUB 10500
2220 GOTO 2080
2230 GOSUB 10000
2240 REM Bloque número 20.
2250 REM Cálculo de los polinomios de Sturm.
2260 FOR J=0 TO N
2270 B#(0,J)=N#(J)
2280 NEXT J
2290 L=1
2300 FOR J=N-1 TO 0 STEP -1
2310 B#(L,J)=N#(J+1)*CDBL(J+1)
2320 M#(J)=B#(L,J)
2330 NEXT J
2340 M=N-1
2350 GOSUB 8000
2360 GOSUB 8500
2370 FOR I=0 TO M
2380 D#(I)=M#(I)
2390 NEXT I
```



```
2400 P=N
2410 GOSUB 9500
2420 GOSUB 10500
2430 L=L+1
2440 FOR J=P-1 TO 0 STEP -1
2450 D#(J)=-D#(J)
2460 B#(L,J)=D#(J)
2470 NEXT J
2480 IF L=N THEN 2500
2490 GOTO 2410
2500 REM Bloque número 21.
2510 REM Cálculo de los subintervalos que separan
2520 REM las raíces de la ecuación.
2530 A#=CDBL(INF)
2540 B#=CDBL(SUP)
2550 LG#=B#-A#
2560 E#(0,0)=A#
2570 E#(0,N)=B#
2580 R=0
2590 GOSUB 11000
2600 IF C=N THEN 2660
2610 GOSUB 15000
2620 ETIQ=3
2630 GOSUB 5000
2640 GOSUB 5500
2650 GOTO 3840
2660 E#(1,0)=CDBL(C)
2670 A#=B#
2680 R=R+1
2690 GOSUB 11000
2700 E#(1,N)=CDBL(C)
2710 R=R+1
2720 I=0
2730 J=1
2740 K=N
2750 GOSUB 12000
2760 IF F=0 THEN 2830
2770 D=0
2780 FOR P=0 TO N-1
2790 V=CINT(E#(1,P)-E#(1,P+1))
2800 IF V=0 THEN 2840
2810 IF V=1 THEN 2880
2820 IF V>1 THEN 2980
2830 GOTO 3080
2840 IF D<>0 THEN 2860
2850 T=P
2860 D=D+1
```

```
2870 GOTO 2960
2880 IF D=0 THEN 2960
2890 GOSUB 12500
2900 I=T+1
2910 J=T+2
2920 K=N
2930 GOSUB 12000
2940 IF F=0 THEN 2770
2950 GOTO 2770
2960 NEXT P
2970 GOTO 3080
2980 IF D=0 THEN 3010
2990 GOSUB 12500
3000 P=T
3010 E#(1,P+V)=E#(1,P+1)
3020 E#(0,P+V)=E#(0,P+1)
3030 I=P
3040 J=P+1
3050 K=P+V
3060 GOSUB 12000
3070 GOTO 2770
3080 FOR J=0 TO N-1
3090 I#(J,0)=E#(0,J)
3100 I#(J,1)=E#(0,J+1)
3110 NEXT J
3120 TOL#=LG#*PO#
3130 FOR J=0 TO N-1
3140 A#=I#(J,0)
3150 GOSUB 13000
3160 B#=S#
3170 A#=I#(J,1)
3180 GOSUB 13000
3190 IF B#*S#>=0# THEN 3220
3200 NEXT J
3210 GOTO 3270
3220 GOSUB 15000
3230 ETIQ=3
3240 GOSUB 5000
3250 GOSUB 5500
3260 GOTO 3840
3270 MA#=I#(0,1)-I#(0,0)
3280 FOR I=0 TO N-1
3290 IF MA#>I#(I,1)-I#(I,0) THEN 3310
3300 MA#=I#(I,1)-I#(I,0)
3310 NEXT I
3320 FOR J=1 TO LI
3330 FOR I=0 TO N-1
```

```

3340 A#=I#(I,0)
3350 GOSUB 13000
3360 B#=S#
3370 A#=(I#(I,0)+I#(I,1))/2#
3380 GOSUB 13000
3390 IF S#=0# THEN 3430
3400 IF B#*S#<0# THEN 3460
3410 I#(I,0)=A#
3420 GOTO 3470
3430 I#(I,0)=A#-MA#/3#
3440 I#(I,1)=A#+MA#/3#
3450 GOTO 3470
3460 I#(I,1)=A#
3470 NEXT I
3480 GOSUB 13500
3490 IF MA#<TOL# THEN 3650
3500 FOR I=0 TO N-1
3510 IF I#(I,0)<>I#(I,1) THEN 3540
3520 I#(I,0)=I#(I,0)-MA#/3#
3530 I#(I,1)=I#(I,1)+MA#/3#
3540 NEXT I
3550 NEXT J
3560 REM Bloque número 22.
3570 REM Salida cuando no se encuentran los subintervalos
3580 REM en el número establecido de iteraciones.
3590 GOSUB 14000
3600 ETIQ=3
3610 CLS
3620 GOSUB 5000
3630 GOSUB 5500
3640 GOTO 3840
3650 REM Bloque número 23.
3660 REM Salida cuando se encuentran los subintervalos
3670 REM en el número establecido de iteraciones.
3680 GOSUB 14500
3690 ETIQ=2
3700 CLS
3710 GOSUB 5000
3720 GOSUB 5500
3730 FOR K=0 TO N-1
3740 CLS
3750 TI$(0)="En el subintervalo"
3760 TI$(1)="("+STR$(I#(K,0))+", "+STR$(I#(K,1))+")"
3770 TI$(2)="se encuentra separada la raíz"
3780 TI$(3)="número"+STR$(K+1)+" de la ecuación."
3790 TI$(4)="Presione cualquier tecla para continuar"
3800 ETIQ=4

```

```

3810 GOSUB 5000
3820 GOSUB 5500
3830 NEXT K
3840 REM Bloque número 24.
3850 REM Fin del programa.
3860 CLS
3870 TI$(0)="¿Desea ejecutar nuevamente el programa?"
3880 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
3890 TI$(2)="en caso contrario, cualquier otra tecla para"
3900 TI$(3)="finalizar la ejecución del programa"
3910 ETIQ=3
3920 GOSUB 5000
3930 GOSUB 5500
3940 IF Z$<>"1" THEN 4050
3950 CLS
3960 TI$(0)="¿Desea realizar los cálculos sobre el mismo"
3970 TI$(1)="polinomio? Si es así, pulse ``1'' para"
3980 TI$(2)="hacerlo, y, en caso contrario, cualquier otra"
3990 TI$(3)="tecla para cargar todos los datos nuevamente"
4000 ETIQ=3
4010 GOSUB 5000
4020 GOSUB 5500
4030 IF Z$="1" THEN 860 ELSE GOSUB 15500
4040 GOTO 200
4050 KEY ON
4060 CLS
4070 GOSUB 16000
4080 STOP
4500 REM Subrutinas del programa.
5000 REM Subrutina número 1.
5010 REM Impresión centrada de las etiquetas del
5020 REM programa, por página.
5030 FOR I=0 TO ETIQ
5040 FOR J=1 TO 3
5050 PRINT
5060 NEXT J
5070 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
5080 NEXT I
5090 RETURN
5500 REM Subrutina número 2.
5510 REM Paralización del avance del programa mientras
5520 REM el usuario lee los mensajes.
5530 Z$=""
5540 WHILE Z$=""
5550 Z$=INKEY$
5560 WEND
5570 RETURN

```

```

6000 REM Subrutina número 3.
6010 REM Verificar que el coeficiente de X^N es
6020 REM mayor que cero.
6030 IF Q#(0,N)>0# THEN 6070
6040 FOR I=0 TO N
6050 Q#(0,I)=-Q#(0,I)
6060 NEXT I
6070 RETURN
6500 REM Subrutina número 4.
6510 REM Examinar si cero es o no cota de las raíces
6520 REM del polinomio.
6530 FOR I=N-1 TO 0 STEP -1
6540 IF Q#(0,I)<0# THEN 6590
6550 NEXT I
6560 SUP=0
6570 CONTROL$="CERO"
6580 GOTO 6600
6590 P=I
6600 RETURN
7000 REM Subrutina número 5.
7010 REM Método del mayor de los módulos de los
7020 REM coeficientes negativos de la ecuación.
7030 IF CONTROL$="CERO" THEN 7120
7040 M=0
7050 FOR I=1 TO N
7060 IF Q#(0,M)<=Q#(0,I) THEN 7080
7070 M=I
7080 NEXT I
7090 G#=ABS(Q#(0,M))
7100 A#=Q#(0,N)
7110 SUPR=1+INT(1#+(G#/A#)^(1/(N-P)))
7120 RETURN
7500 REM Subrutina número 6.
7510 REM Método de Newton.
7520 FOR J=1 TO N+1
7530 Q#(J,N)=Q#(0,N)
7540 NEXT J
7550 FOR P=0 TO SUPR
7560 GRA=0
7570 FOR M=1 TO N
7580 FOR J=N-1 TO GRA STEP -1
7590 Q#(M,J)=Q#(M,J+1)*CDBL(P)+Q#(M-1,J)
7600 NEXT J
7610 GRA=GRA+1
7620 NEXT M
7630 J=0
7640 FOR M=1 TO N+1

```

```

7650 IF Q#(M,J)<=0# THEN 7690

7660 J=J+1
7670 NEXT M
7680 GOTO 7700
7690 NEXT P
7700 IF SW=1 THEN INF=-P ELSE SUP=P
7710 RETURN
8000 REM Subrutina número 7.
8010 REM Determinar el menor coeficiente (en valor
8020 REM absoluto) no nulo de un polinomio.
8030 FOR I=0 TO M
8040 IF ABS(M#(I))=0# THEN NEXT I
8050 ME#=ABS(M#(I))
8060 FOR I=0 TO M
8070 IF M#(I)=0# THEN 8090 ELSE IF ME#<=ABS(M#(I)) THEN 8090
8080 ME#=ABS(M#(I))
8090 NEXT I
8100 RETURN
8500 REM Subrutina número 8.
8510 REM Descomponer el menor coeficiente del polinomio
8520 REM en sus factores primos.
8530 B#=2#
8540 GOTO 8560
8550 GOSUB 9000
8560 IF ME#=1# THEN 8670
8570 IF ME#/B#=INT(ME#/B#) THEN 8550 ELSE B#=3#
8580 C#=INT(SQR(ME#)+1#)
8590 IF B#<C# THEN 8630
8600 B#=ME#
8610 GOSUB 9000
8620 GOTO 8670
8630 IF ME#/B#=INT(ME#/B#) THEN 8650 ELSE B#=B#+2#
8640 GOTO 8580
8650 GOSUB 9000
8660 GOTO 8580
8670 RETURN
9000 REM Subrutina número 9.
9010 REM Simplificar los coeficientes del polinomio
9020 REM si es posible.
9030 ME#=ME#/B#
9040 FOR I=0 TO M
9050 IF M#(I)/B#<>INT(M#(I)/B#) THEN 9100
9060 IF I<>M THEN NEXT I
9070 FOR I=0 TO M
9080 M#(I)=M#(I)/B#
9090 NEXT I

```

```

9100 RETURN
9500 REM Subrutina número 10.
9510 REM División de polinomios con el algoritmo
9520 REM de los enteros.
9530 FOR I=0 TO M+1
9540 N#(I)=N#(I)*ABS(D#(M))
9550 NEXT I
9560 FOR I=P-M TO 0 STEP -1
9570 C#(I)=N#(I+M)/D#(M)
9580 FOR J=0 TO M
9590 N#(I+J)=N#(I+J)-C#(I)*D#(J)
9600 NEXT J
9610 IF I=0 THEN 9660
9620 FOR J=0 TO M
9630 N#(J)=N#(J)*ABS(D#(M))
9640 NEXT J
9650 NEXT I
9660 RETURN
10000 REM Subrutina número 11.
10010 REM Simplificar nuevamente el polinomio original
10020 REM y preparar su división por su M.C.D.
10030 FOR J=0 TO N
10040 N#(J)=P#(J)
10050 M#(J)=P#(J)
10060 NEXT J
10070 L=M
10080 M=N
10090 GOSUB 8000
10100 GOSUB 8500
10110 FOR J=0 TO N
10120 N#(J)=M#(J)
10130 NEXT J
10140 M=L
10150 RETURN
10500 REM Subrutina número 12.
10510 REM Preparar el cálculo del siguiente candidato a
10520 REM M.C.D. del polinomio original.
10530 FOR I=P-2 TO 0 STEP -1
10540 M#(I)=N#(I)
10550 NEXT I
10560 FOR J=P-2 TO 0 STEP -1
10570 IF M#(J)<>0# THEN 10590
10580 NEXT J
10590 M=J
10600 GOSUB 8000
10610 GOSUB 8500
10620 FOR I=P-1 TO 0 STEP -1

```

```

10630 N#(I)=D#(I)
10640 NEXT I
10650 FOR I=P-2 TO 0 STEP -1
10660 D#(I)=M#(I)
10670 NEXT I
10680 P=P-1
10690 RETURN
11000 REM Subrutina número 13.
11010 REM Cálculo del número de variaciones de la sucesión
11020 REM de Sturm en el valor A.
11030 I=0
11040 FOR J=N TO 1 STEP -1
11050 N#(J)=B#(I,J)
11060 FOR M=J-1 TO 0 STEP -1
11070 N#(M)=A#*N#(M+1)+B#(I,M)
11080 NEXT M
11090 M#(I)=N#(0)
11100 I=I+1
11110 NEXT J
11120 IF M#(0)<>0# THEN 11350
11130 IF R<>0 THEN 11240
11140 CLS
11150 TI$(0)="La cota inferior obtenida "+STR$(A#)
11160 TI$(1)="es una raíz de la ecuación, el programa"
11170 TI$(2)="no es capaz de calcular cotas para las"
11180 TI$(3)="raíces de la ecuación"
11190 TI$(4)="Presione cualquier tecla para continuar"
11200 ETIQ=4
11210 GOSUB 5000
11220 GOSUB 5500
11230 GOTO 4050
11240 IF R<>1 THEN 11350
11250 CLS
11260 TI$(0)="La cota superior obtenida "+STR$(B#)
11270 TI$(1)="es una raíz de la ecuación, el programa"
11280 TI$(2)="no es capaz de calcular cotas para las"
11290 TI$(3)="raíces de la ecuación"
11300 TI$(4)="Presione cualquier tecla para continuar"
11310 ETIQ=4
11320 GOSUB 5000
11330 GOSUB 5500
11340 GOTO 4050
11350 M#(N)=B#(N,0)
11360 C=0
11370 FOR J=0 TO N-1
11380 IF M#(J)=0# THEN 11460
11390 K=J+1

```



```

11400 IF M#(K)=0# THEN 11500
11410 IF M#(J)>0# THEN IF M#(K)>0# THEN 11440
11420 IF M#(J)<0# THEN IF M#(K)<0# THEN 11440
11430 C=C+1
11440 IF K=N THEN 11520
11450 GOTO 11470
11460 IF J=0 THEN 11480
11470 J=K-1
11480 NEXT J
11490 GOTO 11520
11500 K=K+1
11510 IF K<=N THEN 11400
11520 RETURN
12000 REM Subrutina número 14.
12010 REM Verificar que el número de variaciones sea el
12020 REM correcto en los intervalos, y, si es necesario,
12030 REM se reeligen los extremos del subintervalo.
12040 V=CINT(E#(1,I)-E#(1,K))
12050 F=0
12060 IF V=1 THEN 12340
12070 F=1
12080 L#=E#(0,K)-E#(0,I)
12090 LO#=L#/CDBL(V)
12100 E=J
12110 E#(0,E)=E#(0,E-1)+LO#
12120 A#=E#(0,E)
12130 GOSUB 11000
12140 E#(1,E)=CDBL(C)
12150 E=E+1
12160 IF E<=K-1 THEN 12110
12170 U=0
12180 A#=E#(0,U)
12190 GOSUB 13000
12200 IF S#<>0# THEN 12250
12210 E#(0,U)=E#(0,U)+(E#(0,U+1)-E#(0,U))/3#
12220 A#=E#(0,U)
12230 GOSUB 11000
12240 E#(1,U)=CDBL(C)
12250 U=U+1
12260 IF U<=N-1 THEN 12180
12270 A#=E#(0,N)
12280 GOSUB 13000
12290 IF S#<>0# THEN 12340
12300 E#(0,N)=E#(0,N)+1#/3#
12310 A#=E#(0,N)
12320 GOSUB 11000
12330 E#(1,N)=CDBL(C)

```

```

12340 RETURN
12500 REM Subrutina número 15.
12510 REM Substituye los extremos de los subintervalos
12520 REM que no poseen raíces, por otros que si las
12530 REM poseen.
12540 E#(1,T)=E#(1,P)
12550 E#(1,T+1)=E#(1,P+1)
12560 E#(0,T)=E#(0,P)
12570 E#(0,T+1)=E#(0,P+1)
12580 RETURN
13000 REM Subrutina número 16.
13010 REM Evaluación del polinomio reducido en el valor A
13020 REM para reducir la longitud de los subintervalos que
13030 REM separan a las raíces.
13040 N#(N)=B#(0,N)
13050 FOR M=N-1 TO 0 STEP -1
13060 N#(M)=A#*N#(M+1)+B#(0,M)
13070 NEXT M
13080 S#=N#(M+1)
13090 RETURN
13500 REM Subrutina número 17.
13510 REM Cálculo de la longitud del mayor de los
13520 REM subintervalos que separan las raíces.
13530 MA#=0#
13540 FOR I=0 TO N-1
13550 LO#=I#(I,1)-I#(I,0)
13560 IF MA#>=LO# THEN 13580
13570 MA#=LO#
13580 NEXT I
13590 RETURN
14000 REM Subrutina número 18.
14010 REM Mensajes de salida cuando no se encuentran
14020 REM los subintervalos.
14030 TI$(0)="Los subintervalos que acotan a las raíces"
14040 TI$(1)="simples de la ecuación no se han podido"
14050 TI$(2)="calcular, en "+STR$(LI)+" iteraciones"
14060 TI$(3)="Presione cualquier tecla para continuar"
14070 RETURN
14500 REM Subrutina número 19.
14510 REM Mensajes de salida cuando se encuentran
14520 REM los subintervalos.
14530 TI$(0)="Los subintervalos que acotan a las raíces"
14540 TI$(1)="simples de la ecuación, han sido calculados"
14550 TI$(2)="Presione cualquier tecla para continuar"
14560 RETURN
15000 REM Subrutina número 20.
15010 REM Mensajes de salida cuando la ecuación posee

```

```
15020 REM raíces complejas.
15030 TI$(0)="Los intervalos introducidos deben ampliarse"
15040 TI$(1)="o la ecuación posee raíces complejas, por lo"
15050 TI$(2)="que el método no le es aplicable"
15060 TI$(3)="Presione cualquier tecla para continuar"
15070 RETURN
15500 REM Subrutina número 21.
15510 REM Inicialización de las variables del programa
15520 REM para otra ejecución.
15530 ERASE B#,C#,D#,E#,I#,M#,N#,P#,POLY#,Q#
15540 RETURN
16000 REM Subrutina número 22.
16010 REM Fin del programa.
16020 END
```

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido separadas empleando el programa que acabamos de mostrar.

TABLA 10

ECUACIÓN: $P(x)=0$	RAÍCES VERDADERAS	INTERVALO SEGÚN EL MÉTODO
$x^3 - 5x + 1$	- 2.3300587 0.20163968 2.1284191	(- 2.330059051513672, - 2.330055236816406) (0.2016372680664062, 0.2016410827636719) (2.12841796875, 2.128421783447266)
$x^3 - 7x - 7$	- 1.6920215 - 1.3568959 3.0489173	(- 1.69202184677124, - 1.692021369934082) (- 1.356895923614502, - 1.356895446777344) (3.048914909362793, 3.048919677734375)
$x^4 - 41x^2 + 400$	- 5.0000000 - 4.0000000 4.0000000 5.0000000	(- 5.000000953674316, - 4.999999523162842) (- 4.000001907348633, - 3.999995470046997) (3.999998331069946, 4.00000011920929) (4.999999165534973, 5.000000953674316)
$3x^4 - 40x^3 + 130x^2 - 120x + 27$	0.33333333 1.0000000 3.0000000 9.0000000	(0.3333330154418945, 0.33333420753479) (0.9999990463256836, 1.000001430511475) (2.999996542930603, 3.000004291534424) (8.999996185302734, 9.000003933906555)
$x^4 - 40x^3 + 390x^2 - 1080x + 729$	1.0000000 3.0000000 9.0000000 27.0000000	(0.9999995231628418, 1.000002861022949) (2.999995231628418, 3.000001907348633) (8.999995708465576, 9.000017404556274) (26.99998378753662, 27.00000548362732)
$x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	-2.0000000 1.0000000 3.0000000 4.0000000	(- 2.000003090426697, - 1.999996909573303) (0.9999969095733031, 1.000003090426697) (2.999999999689559, 3.00000000015522) (3.999996909573303, 4.000003090426697)

Como en la tabla **9**, todas las ecuaciones fueron procesadas empleando como longitud máxima de los subintervalos que acotarán a las raíces de la ecuación, el valor mínimo que puede asumir esta variable dentro del programa, un error $\varepsilon = 0.0001 = 10^{-4}$, es decir, una exactitud en el cálculo muy elevada, y nuevamente se permitió que el método iterara un número elevado de veces (100), de modo que tuviera oportunidad de separar las raíces de las ecuaciones con la precisión pedida.

Al comparar las tablas **9** y **10**, es evidente que los intervalos que separan las raíces, generados con esta nueva técnica, son, básicamente, similares a las obtenidas a través del método de Sturm sólo. Sin embargo, con el método integrado no es necesario suministrar un intervalo que acote a las raíces, de modo que, éste es más eficiente en general.

Sin embargo, debemos resaltar que, en ambos casos, la construcción que hemos realizado falla cuando las raíces son extremadamente cercanas (distancia menor que 10^{-5}), de modo que no podemos considerar que este método es la culminación de este estudio. El mejoramiento de esta técnica puede y debe ser realizado en estudios más profundos acerca del tema.

Así, en base a las consideraciones que hemos realizado, asumiremos que el problema de separar los ceros reales de una ecuación algebraica de raíces reales, al aplicar el último método desarrollado, se encuentra razonablemente resuelto.

A continuación, consideraremos la implementación en computadoras del proceso que sigue en nuestro desarrollo teórico, es decir, la aproximación de los ceros reales de una ecuación algebraica de raíces reales.

3.6.3. MÉTODOS QUE APROXIMAN RAÍCES REALES.

Con respecto a estos métodos, en primer término, establecemos que sólo se realizará el desarrollo de los métodos de bisección de Bolzano y de Newton-Raphson-Fourier, ya que, en la práctica, son los que más fácilmente se pueden implementar para que operen con ecuaciones algebraicas.

Ambos métodos serán desarrollados siguiendo un esquema similar que consistirá en emplear la información conocida hasta el momento (la suministrada y la generada por el método mixto de separación), para generar la aproximación a la raíz simple. Luego, se aplicará el teorema número **4** del apéndice de teoría de ecuaciones para determinar la multiplicidad de cada raíz. Es evidente, entonces, que todo el proceso involucra un margen de error en que se puede incurrir, que se tratará de controlar en lo posible.

Es importante resaltar que, en los programas integrados, el número de iteraciones que empleará el método será el aplicado en las etapas de separación y aproximación, de modo que el mismo no debe ser un valor muy pequeño, que impida actuar al método de separación de raíces; sin embargo, la respuesta final será dada en función a la etapa de aproximación.

En primer término, presentamos un listado de las variables más importantes que usarán los programas de separación de raíces.

VARIABLES MÁS IMPORTANTES QUE USARÁN LOS PROGRAMAS SOBRE APROXIMACIÓN DE RAÍCES

VARIABLE	SIGNIFICADO
TI\$(5)	Vector alfanumérico donde se guardan las salidas del programa.
ETIQ	Variable numérica (de tipo precisión simple) que representa el número de etiquetas a imprimir, en un momento dado de la ejecución del programa.
N, GR, GRA, M y P	Variables numéricas (de tipo precisión simple) que representan los grados de los polinomios usados dentro del programa.
P#(N), N#(N), D#(N), M#(N), POLY#(N) y C#(N-1)	Vectores numéricos (de tipo doble precisión) donde se guardan los coeficientes del polinomio original y los obtenidos.
PO# y LI	Variables numéricas (de tipos doble y simple precisión, respectivamente) que contienen, en la secuencia descrita, la longitud máxima a que reducirá la longitud de los subintervalos a obtener, y el número máximo de iteraciones a realizar para reducir la longitud de dichos intervalos.
B#(N,N)	Matriz numérica (de tipo doble precisión) donde se guardan los coeficientes de los polinomios de la sucesión de Sturm relacionada a la ecuación de raíces simples obtenida.

VARIABLE	SIGNIFICADO
Q#(N+1,N)	Matriz numérica (de tipo doble precisión) donde se calculan las evaluaciones de la función y sus derivadas, en un punto del intervalo y en la raíz obtenida, para determinar su multiplicidad. Además, aquí se calcula la cota del error cometido en la última iteración en el método de Newton-Raphson-Fourier.
I#(N,1)	Matriz numérica (de tipo doble precisión) donde se almacenan los intervalos que acotarán a las raíces; cada pareja I#(K,0), I#(K,1) (para k = 0, 1, ..., N), son los extremos de un subintervalo, ya sea suministrado por el usuario, u obtenido por el método de Sturm o por el procedimiento de bisección.
IN	Variable numérica (de tipo precisión simple) que contiene el número de subintervalos que se calcularán.
Z\$	Variable alfanumérica; es un interruptor, controla el avance del programa, para que el usuario pueda leer las pantallas.
LG#	Variable numérica (de tipo doble precisión) que contiene la longitud del intervalo que acota las raíces del polinomio.
E#(1,N)	Matriz numérica (de tipo doble precisión) que contiene, en su fila 0, los extremos de los subintervalos que genera el método, y en su fila 1, el número de variaciones que presenta la sucesión de Sturm para el punto en mención.
C y V	Variables numéricas (de tipo precisión simple) donde se cuenta el número de variaciones que va presentando la sucesión de Sturm.

VARIABLE	SIGNIFICADO
R	Es un interruptor que nos permite determinar cuándo uno de los extremos obtenidos es raíz de la ecuación original.
TOL#	Variable numérica (de tipo doble precisión) que guarda el valor calculado de la tolerancia máxima del error en que se puede incurrir al aproximar los intervalos.
MA#	Variable numérica (de tipo doble precisión) que representa la longitud mayor de todos los subintervalos calculados en una iteración.
MULTI y MULT	Variables numéricas (de tipo precisión simple), que se emplean para determinar la multiplicidad de las raíces.
ER#	Variables numérica (de tipo precisión doble) donde se calcula el error relativo de cada aproximación.
ERRMAX#	Variable numérica (de tipo precisión doble) donde se calcula una cota del error cometido en la aproximación al realizar la última iteración.
IA#, FA#, IB# y FB#	Variables numéricas (de tipo precisión doble) donde se almacenan los extremos de los subintervalos que genera el método, así como los valores que surgen al evaluar los en la función.
PM# y FPM#	Variables numéricas (de tipo doble precisión), donde se calcula el punto medio del subintervalo y el valor que surge al evaluar dicho valor en la función.
H0#	Variable numérica (de tipo doble precisión), donde se guarda el extremo del intervalo introducido, que se empleará para iniciar el proceso iterativo, en el método de Newton-Raphson-Fourier.

VARIABLE	SIGNIFICADO
DES#	Variable numérica (de tipo doble precisión), donde se guarda el extremo del intervalo introducido que no se emplea como valor inicial, en el método de Newton-Raphson-Fourier.
HI# y HJ#	Variables numéricas (de tipo doble precisión), donde se guardan el valor aproximado hasta la iteración anterior, y el obtenido en la última iteración, respectivamente.
INF, SUP, A#, y B#	Variables numéricas (de tipos simple y doble precisión) que contienen, por parejas, los extremos del intervalo que acota a las raíces reales del polinomio.

A continuación, implementaremos en computadora el método de bisección de Bolzano.

3.6.3.1. MÉTODO DE BISECCIÓN DE BOLZANO.

La implementación de un programa que realice la aproximación de los ceros de una ecuación algebraica de raíces reales, presenta un cierto número de particularidades y limitaciones que, a continuación, detallaremos:

En primer término, hemos suprimido las raíces múltiples de la ecuación, para facilitar la acción del método sin embargo, por la construcción del método, éste puede manejarse sin problemas sobre las raíces múltiples. Esta acción la hemos realizado implementando nuevamente el esquema desarrollado en el método de Sturm, antes de aplicar, en sí, el método de Bisección de Bolzano. Por otro lado, dicha técnica limita la acción del programa que desarrollaremos a ecuaciones que se reduzcan a polinomios de, al menos, grado dos.

Posteriormente, el método se encargará de determinar la multiplicidad de cada raíz obtenida, al aplicar el criterio de las derivadas a éstas, considerando que dichas derivadas, al evaluarse, sean menores, en valor absoluto, que un múltiplo de la tolerancia del error que se introduzca como dato de entrada. Además, con respecto a dicha tolerancia, hay otras situaciones que aclarar.

Así, tenemos que la concepción teórica del método es solamente de tipo existencial y sumamente elemental; es evidente que la sucesión de valores que convergerá a la raíz buscada, asumiendo que ella existe, requerirá del cálculo de muchos elementos, antes de alcanzar una precisión elevada; sin embargo, hemos establecido que, en un caso extremo, se aproximarán las raíces con una precisión que varía dentro del intervalo $[10^{-17}, 10^{-8})$, en tal caso, el usuario debe complementar este valor con un número elevado de iteraciones, para lograr la convergencia del método.

También exigimos que se suministren los subintervalos que, se supone, contienen a las raíces del polinomio. Posteriormente, el programa verificará si en realidad ellos las contienen, y, en caso contrario, se informará al usuario la anomalía.

También el programa verificará que la suma de las multiplicidades de cada raíz totalice el grado del polinomio original, y, en caso de que esto no se cumpla, lo informará al usuario.

Finalmente, como en los programas anteriores, el usuario podrá, si lo desea, repetir la ejecución del programa o aplicarlo sobre la ecuación previamente introducida o sobre otra ecuación, según el caso.

A continuación, pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO DE BISECCIÓN DE BOLZANO

```

10 REM Programa número 7.
20 REM Cálculo de las raíces reales de una
30 REM ecuación algebraica de coeficientes reales,
40 REM empleando el método de bisección de Bolzano.
50 REM Bloque número 1.
60 REM Presentación del programa.
70 KEY OFF
80 CLS
90 CLEAR
100 DIM TI$(5)
110 TI$(0)="Cálculo de las raíces reales de una"
120 TI$(1)="ecuación algebraica de coeficientes"
130 TI$(2)="reales, empleando el método de"
140 TI$(3)="bisección de Bolzano"
150 TI$(4)="Presione cualquier tecla para continuar"
160 ETIQ=4
170 GOSUB 3500
180 GOSUB 4000
190 REM Bloque número 2.
200 REM Carga del grado del polinomio.
210 CLS
220 CLEAR
230 DIM TI$(5)
240 TI$(0)="Por favor, introduzca el grado del"
250 TI$(1)="polinomio (N) (El método requiere que"
260 TI$(2)="el polinomio sea de grado mayor o igual"
270 TI$(3)="que dos o no funcionará correctamente)"
280 TI$(4)="y pulse ``Enter'' para continuar"
290 ETIQ=4
300 GOSUB 3500
310 PRINT
320 PRINT
330 INPUT "N=",N
340 IF N>=2 AND N=FIX(N) THEN 420
350 CLS
360 TI$(0)="La información introducida es incorrecta"
370 TI$(1)="Presione cualquier tecla para continuar"
380 ETIQ=1
390 GOSUB 3500
400 GOSUB 4000
410 GOTO 210
420 REM Bloque número 3.
430 REM Dimensiones de los vectores y matrices iniciales.
440 DIM C#(N-1),D#(N),I#(N,1),M#(N),N#(N),P#(N)

```

```

450 DIM POLY#(N),Q#(N+1,N)
460 GR=N
470 REM Bloque número 4.
480 REM Carga de los coeficientes del polinomio
490 REM y cálculo del mayor número de decimales presente
500 REM en dichos coeficientes.
510 CLS
520 TI$(0)="Introduzca los coeficientes del"
530 TI$(1)="polinomio, en orden descendente y pulse"
540 TI$(2)="`Enter'" para continuar, en cada caso"
550 TI$(3)="Presione cualquier tecla para continuar"
560 ETIQ=3
570 GOSUB 3500
580 GOSUB 4000
590 CLS
600 FOR I=N TO 0 STEP -1
610 PRINT "Introduzca el coeficiente de X^";I;
620 INPUT "=",POLY#(I)
630 VAR$=STR$(POLY#(I))
640 LONG=LEN(VAR$)
650 FOR J=1 TO LONG
660 CARC$=MID$(VAR$,J,1)
670 IF CARC$="." THEN POT=LONG-J ELSE NEXT J
680 IF I=N THEN NU=POT
690 IF NU>POT THEN 720
700 NU=POT
710 P#(I)=POLY#(I)
720 PRINT
730 PRINT
740 NEXT I
750 IF P#(N)<>0# AND P#(0)<>0# THEN 850
760 CLS
770 TI$(0)="El polinomio introducido puede simplificarse,"
780 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+" ,"
790 TI$(2)="ni el de X^ 0, pueden ser nulos"
800 TI$(3)="Presione cualquier tecla para continuar"
810 ETIQ=3
820 GOSUB 3500
830 GOSUB 4000
840 GOTO 210
850 REM Bloque número 5.
860 REM Carga de la tolerancia y número máximo de
870 REM iteraciones que realizará el método.
880 CLS
890 N=GR
900 TI$(0)="Introduzca la cota máxima al error (PO)"
910 TI$(1)="que se permitirá en la aproximación de"

```

```

920 TI$(2)="las raíces del polinomio"
930 ETIQ=2
940 GOSUB 3500
950 PRINT
960 PRINT
970 INPUT "Valor ( $10^{-17} \leq PO < 10^{-8}$ ) (PO) =  $10^$  ",PO#
980 CLS
990 TI$(0)="Introduzca el número máximo de iteraciones"
1000 TI$(1)="a realizar (LI) al aplicar el método sobre"
1010 TI$(2)="los subintervalos que acotan a las raíces"
1020 ETIQ=2
1030 GOSUB 3500
1040 PRINT
1050 PRINT
1060 INPUT "Número de iteraciones (LI) =",LI
1070 PO#=CDBL( $10^{PO\#}$ )
1080 IF PO#<CDBL( $10^{-17}$ ) OR PO#>CDBL( $10^{-8}$ ) THEN 1110
1090 IF LI<=0 THEN 1110
1100 IF LI=FIX(LI) THEN 1180
1110 CLS
1120 TI$(0)="La información introducida es incorrecta"
1130 TI$(1)="Presione cualquier tecla para continuar"
1140 ETIQ=1
1150 GOSUB 3500
1160 GOSUB 4000
1170 GOTO 850
1180 REM Bloque número 6.
1190 REM Carga del número de subintervalos que acotan
1200 REM a las raíces simples del polinomio.
1210 CLS
1220 TI$(0)="Por favor, introduzca el número de"
1230 TI$(1)="subintervalos que acotan a las raíces"
1240 TI$(2)="simples del polinomio (IN), y pulse"
1250 TI$(3)="`Enter` para continuar"
1260 ETIQ=3
1270 GOSUB 3500
1280 PRINT
1290 PRINT
1300 PRINT "Número de subintervalos ( $0 < IN \leq N$ );";
1310 INPUT " (IN)=",IN
1320 IF IN>=1 AND IN=FIX(IN) AND IN<=N THEN 1400
1330 CLS
1340 TI$(0)="La información introducida es incorrecta"
1350 TI$(1)="Presione cualquier tecla para continuar"
1360 ETIQ=1
1370 GOSUB 3500
1380 GOSUB 4000

```

```

1390 GOTO 1210
1400 REM Bloque número 7.
1410 REM Dimensión de la matriz de subintervalos iniciales.
1420 ERASE I#
1430 DIM I#(IN,1)
1440 REM Bloque número 8.
1450 REM Carga de los subintervalos que acotan
1460 REM a las raíces simples del polinomio.
1470 CLS
1480 TI$(0)="Introduzca los extremos superior e inferior"
1490 TI$(1)="de los subintervalos que acotan las raíces"
1500 TI$(2)="de la ecuación (A, B) y pulse ``Enter'"
1510 TI$(3)="para continuar, en cada caso"
1520 TI$(4)="Presione cualquier tecla para continuar"
1530 ETIQ=4
1540 GOSUB 3500
1550 GOSUB 4000
1560 CLS
1570 FOR I=1 TO IN
1580 PRINT
1590 PRINT
1600 PRINT "Extremo inferior del ";
1610 PRINT "subintervalo número";I;":  A";I;
1620 INPUT "=",I#(I,0)
1630 PRINT
1640 PRINT
1650 PRINT "Extremo superior del ";
1660 PRINT "subintervalo número";I;":  B";I;
1670 INPUT "=",I#(I,1)
1680 NEXT I
1690 FOR I=1 TO IN
1700 IF I#(I,0)>I#(I,1) THEN 1760
1710 NEXT I
1720 FOR I=1 TO IN-1
1730 IF I#(I,1)>I#(I+1,0) THEN 1760
1740 NEXT I
1750 GOTO 1830
1760 CLS
1770 TI$(0)="La información introducida es incorrecta"
1780 TI$(1)="Presione cualquier tecla para continuar"
1790 ETIQ=1
1800 GOSUB 3500
1810 GOSUB 4000
1820 GOTO 1470
1830 REM Bloque número 9.
1840 REM Inicialización de los polinomios de trabajo y
1850 REM supresión de los decimales (si los hay), de los

```

```
1860 REM coeficientes del polinomio.
1870 CLS
1880 IF NU=0 THEN 1920
1890 FOR I=0 TO N
1900 P#(I)=POLY#(I)*CDBL(10^NU)
1910 NEXT I
1920 FOR I=0 TO N
1930 N#(I)=P#(I)
1940 NEXT I
1950 MULTI=0
1960 REM Bloque número 10.
1970 REM Simplificar (si es posible) los coeficientes
1980 REM del polinomio.
1990 FOR I=0 TO N
2000 M#(I)=N#(I)
2010 NEXT I
2020 M=N
2030 GOSUB 4500
2040 GOSUB 5000
2050 FOR I=0 TO N
2060 N#(I)=M#(I)
2070 NEXT I
2080 REM Bloque número 11.
2090 REM Calcular y simplificar la derivada del polinomio.
2100 M=N-1
2110 FOR I=0 TO M
2120 M#(I)=CDBL(I+1)*N#(I+1)
2130 NEXT I
2140 GOSUB 4500
2150 GOSUB 5000
2160 REM Bloque número 12.
2170 REM Reducir la ecuación original a una con raíces
2180 REM simples.
2190 FOR I=0 TO M
2200 D#(I)=M#(I)
2210 NEXT I
2220 P=N
2230 GOSUB 6000
2240 FOR K=P-2 TO 0 STEP -1
2250 IF N#(K)<>0# THEN 2350
2260 NEXT K
2270 GOSUB 6500
2280 P=N
2290 GOSUB 6000
2300 FOR I=P-M TO 0 STEP -1
2310 N#(I)=C#(I)
2320 NEXT I
```



```
2330 N=P-M
2340 GOTO 2390
2350 IF P=2 THEN 2380
2360 GOSUB 7000
2370 GOTO 2230
2380 GOSUB 6500
2390 REM Bloque número 13.
2400 REM Cálculo de las raíces del polinomio.
2410 MULTI=0
2420 FOR K=1 TO IN
2430 CLS
2440 ERRMAX#=0#
2450 GOSUB 7500
2460 MULTI=MULTI+MULT
2470 NEXT K
2480 IF MULTI=GR THEN 2570
2490 CLS
2500 TI$(0)="El programa no ha podido determinar"
2510 TI$(1)="correctamente la multiplicidad de cada"
2520 TI$(2)="raíz de la ecuación"
2530 TI$(3)="Presione cualquier tecla para continuar"
2540 ETIQ=3
2550 GOSUB 3500
2560 GOSUB 4000
2570 REM Bloque número 14.
2580 REM Fin del programa.
2590 CLS
2600 TI$(0)="¿Desea ejecutar nuevamente el programa?"
2610 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
2620 TI$(2)="en caso contrario, cualquier otra tecla para"
2630 TI$(3)="finalizar la ejecución del programa"
2640 ETIQ=3
2650 GOSUB 3500
2660 GOSUB 4000
2670 IF Z$<>"1" THEN 2780
2680 CLS
2690 TI$(0)="¿Desea realizar los cálculos sobre el mismo"
2700 TI$(1)="polinomio? Si es así, pulse ``1'' para"
2710 TI$(2)="hacerlo, y, en caso contrario, cualquier otra"
2720 TI$(3)="tecla para cargar todos los datos nuevamente"
2730 ETIQ=3
2740 GOSUB 3500
2750 GOSUB 4000
2760 IF Z$="1" THEN 850 ELSE GOSUB 11000
2770 GOTO 200
2780 KEY ON
2790 CLS
```

```
2800 GOSUB 11500
2810 STOP
3000 REM Subrutinas del programa.
3500 REM Subrutina número 1.
3510 REM Impresión centrada de las etiquetas del
3520 REM programa, por página.
3530 FOR I=0 TO ETIQ
3540 FOR J=1 TO 3
3550 PRINT
3560 NEXT J
3570 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
3580 NEXT I
3590 RETURN
4000 REM Subrutina número 2.
4010 REM Paralización del avance del programa mientras
4020 REM el usuario lee los mensajes.
4030 Z$=""
4040 WHILE Z$=""
4050 Z$=INKEY$
4060 WEND
4070 RETURN
4500 REM Subrutina número 3.
4510 REM Determinar el menor coeficiente (en valor
4520 REM absoluto) no nulo de un polinomio.
4530 FOR I=0 TO M
4540 IF ABS(M$(I))=0# THEN NEXT I
4550 ME#=ABS(M$(I))
4560 FOR I=0 TO M
4570 IF M$(I)=0# THEN 4590 ELSE IF ME#<=ABS(M$(I)) THEN 4590
4580 ME#=ABS(M$(I))
4590 NEXT I
4600 RETURN
5000 REM Subrutina número 4.
5010 REM Descomponer el menor coeficiente del polinomio
5020 REM en sus factores primos.
5030 B#=2#
5040 GOTO 5060
5050 GOSUB 5500
5060 IF ME#=1# THEN 5170
5070 IF ME#/B#=INT(ME#/B#) THEN 5050 ELSE B#=3#
5080 C#=INT(SQR(ME#)+1#)
5090 IF B#<C# THEN 5130
5100 B#=ME#
5110 GOSUB 5500
5120 GOTO 5170
5130 IF ME#/B#=INT(ME#/B#) THEN 5150 ELSE B#=B#+2#
5140 GOTO 5080
```

```
5150 GOSUB 5500
5160 GOTO 5080
5170 RETURN
5500 REM Subrutina número 5.
5510 REM Simplificar los coeficientes del polinomio
5520 REM si es posible.
5530 ME#=ME#/B#
5540 FOR I=0 TO M
5550 IF M#(I)/B#<>INT(M#(I)/B#) THEN 5600
5560 IF I<>M THEN NEXT I
5570 FOR I=0 TO M
5580 M#(I)=M#(I)/B#
5590 NEXT I
5600 RETURN
6000 REM Subrutina número 6.
6010 REM División de polinomios con el algoritmo
6020 REM de los enteros.
6030 FOR I=0 TO M+1
6040 N#(I)=N#(I)*ABS(D#(M))
6050 NEXT I
6060 FOR I=P-M TO 0 STEP -1
6070 C#(I)=N#(I+M)/D#(M)
6080 FOR J=0 TO M
6090 N#(I+J)=N#(I+J)-C#(I)*D#(J)
6100 NEXT J
6110 IF I=0 THEN 6160
6120 FOR J=0 TO M
6130 N#(J)=N#(J)*ABS(D#(M))
6140 NEXT J
6150 NEXT I
6160 RETURN
6500 REM Subrutina número 7.
6510 REM Simplificar nuevamente el polinomio original
6520 REM y preparar su división por su M.C.D.
6530 FOR J=0 TO N
6540 N#(J)=P#(J)
6550 M#(J)=P#(J)
6560 NEXT J
6570 L=M
6580 M=N
6590 GOSUB 4500
6600 GOSUB 5000
6610 FOR J=0 TO N
6620 N#(J)=M#(J)
6630 NEXT J
6640 M=L
6650 RETURN
```

```

7000 REM Subrutina número 8.
7010 REM Preparar el cálculo del siguiente candidato a
7020 REM M.C.D. del polinomio original.
7030 FOR I=P-2 TO 0 STEP -1
7040 M#(I)=N#(I)
7050 NEXT I
7060 FOR J=P-2 TO 0 STEP -1
7070 IF M#(J)<>0# THEN 7090
7080 NEXT J
7090 M=J
7100 GOSUB 4500
7110 GOSUB 5000
7120 FOR I=P-1 TO 0 STEP -1
7130 N#(I)=D#(I)
7140 NEXT I

7150 FOR I=P-2 TO 0 STEP -1
7160 D#(I)=M#(I)
7170 NEXT I
7180 P=P-1
7190 RETURN
7500 REM Subrutina número 9.
7510 REM Método de bisección de Bolzano.
7520 FOR I=0 TO N+1
7530 FOR J=0 TO N
7540 IF I=0 THEN Q#(I,J)=N#(J) ELSE Q#(I,J)=0#
7550 NEXT J
7560 NEXT I
7570 MULT=0
7580 IA#=I#(K,0)
7590 IB#=I#(K,1)
7600 FOR IT=1 TO LI
7610 P#=IA#
7620 Q#(1,N)=Q#(0,N)
7630 FOR J=N-1 TO 0 STEP -1
7640 Q#(1,J)=Q#(1,J+1)*P#+Q#(0,J)
7650 NEXT J
7660 FA#=Q#(1,0)
7670 PM#=(IA#+IB#)/2#
7680 P#=PM#
7690 FOR J=N-1 TO 0 STEP -1
7700 Q#(1,J)=Q#(1,J+1)*P#+Q#(0,J)
7710 NEXT J
7720 FPM#=Q#(1,0)
7730 P#=IB#
7740 FOR J=N-1 TO 0 STEP -1
7750 Q#(1,J)=Q#(1,J+1)*P#+Q#(0,J)

```

```

7760 NEXT J
7770 FB#=Q#(1,0)
7780 IF FPM#=0# THEN 7990
7790 IF FA#*FPM#>0# THEN 7820
7800 IB#=PM#
7810 GOTO 7840
7820 IF FPM#*FB#>0# THEN 7850
7830 IA#=PM#
7840 GOTO 7910
7850 CLS
7860 GOSUB 8500
7870 GOSUB 3500
7880 GOSUB 4000
7890 CLS
7900 GOTO 8020
7910 ER#=ABS((IB#-IA#)/PM#)
7920 IF ER#<PO# THEN 7990
7930 NEXT IT
7940 GOSUB 9000
7950 CLS
7960 GOSUB 3500
7970 GOSUB 4000
7980 GOTO 8020
7990 R#=PM#
8000 CLS
8010 GOSUB 9500
8020 RETURN
8500 REM Subrutina número 10.
8510 REM Mensaje de salida cuando no se encuentra la raíz.
8520 REM dentro del subintervalo introducido.
8530 TI$(0)="La"+STR$(K)+"-ésima raíz de la ecuación"
8540 TI$(1)="no se pudo calcular, ya que el subintervalo"
8550 TI$(2)="que la acota no cumple las condiciones de"
8560 TI$(3)="bisección, se proseguirá a calcular las demás"
8570 TI$(4)="Presione cualquier tecla para continuar"
8580 ETIQ=4
8590 RETURN
9000 REM Subrutina número 11.
9010 REM Mensaje de salida cuando no se encuentra la raíz.
9020 REM en el número establecido de iteraciones.
9030 TI$(0)="La"+STR$(K)+"-ésima raíz de la ecuación"
9040 TI$(1)="no se pudo calcular en"+STR$(LI)+" iteraciones"
9050 TI$(2)="se proseguirá a calcular las restantes"
9060 TI$(3)="Presione cualquier tecla para continuar"
9070 ETIQ=3
9080 RETURN
9500 REM Subrutina número 12.

```

```

9510 REM Cálculo de la multiplicidad de la raíz localizada.
9520 FOR I=0 TO GR
9530 Q#(0,I)=POLY#(I)
9540 NEXT I
9550 FOR J=1 TO GR+1
9560 Q#(J,GR)=Q#(0,GR)
9570 NEXT J
9580 GRA=0
9590 FOR M=1 TO GR
9600 FOR J=GR-1 TO GRA STEP -1
9610 Q#(M,J)=Q#(M,J+1)*R#+Q#(M-1,J)
9620 NEXT J
9630 GRA=GRA+1
9640 NEXT M
9650 FAC=1
9660 FOR P=2 TO GR+1
9670 FAC=FAC*P
9680 Q#(P,P-1)=Q#(P,P-1)*CDBL(FAC)
9690 NEXT P
9700 MU=1
9710 FOR M=2 TO GR+1
9720 IF ABS(Q#(M,MU))>10000#*PO# THEN 9770
9730 MU=MU+1
9740 NEXT M
9750 MULT=GR
9760 GOTO 9790
9770 MULT=MU
9780 CLS
9790 GOSUB 10000
9800 GOSUB 10500
9810 GOSUB 3500
9820 GOSUB 4000
9830 CLS
9840 RETURN
10000 REM Subrutina número 13.
10010 REM Cálculo de la cota del error cometido en la
10020 REM última iteración del método.
10030 ERRMAX#=ABS((I#(K,1)-I#(K,0))/(2^IT))
10040 RETURN
10500 REM Subrutina número 14.
10510 REM Mensajes de salida cuando se encuentra la
10520 REM raíz y su multiplicidad.
10530 TI$(0)="La"+STR$(K)+"-ésima raíz de la ecuación es"
10540 TI$(1)=STR$(R#)+", una cota del error cometido en la"
10550 TI$(2)="aproximación es:"+STR$(ERRMAX#)+", "
10560 TI$(3)="posee una multiplicidad igual a:"+STR$(MULT)
10570 TI$(4)="y se calculó en"+STR$(IT)+" iteraciones"

```

```
10580 TI$(5)="Presione cualquier tecla para continuar"
10590 ETIQ=5
10600 RETURN
11000 REM Subrutina número 15.
11010 REM Inicialización de las variables del programa
11020 REM para otra ejecución.
11030 ERASE C#,D#,I#,M#,N#,P#,POLY#,Q#
11040 RETURN
11500 REM Subrutina número 16.
11510 REM Fin del programa.
11520 END
```

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido aproximadas empleando el programa que acabamos de mostrar.

TABLA 11

ECUACIÓN	RAÍCES VERDADERAS	MULTIPLICIDAD VERDADERA	RAÍCES OBTENIDAS	MULTIPLICIDAD OBTENIDA	COTA DEL ERROR COMETIDO EN LA ÚLTIMA ITERACIÓN	ITERACIONES EMPLEADAS
1	-2.3300587	1	- 2.330058739567983	1	1.77635683940025E-015	31
	0.20163968	1	0.2016396757234046	1	1.110223024625157E-016	35
	2.1284191	1	2.128419063844577	1	1.77635683940025E-015	31
2	-1.6920215	1	- 1.692021471630096	1	8.881784197001252E-016	29
	- 1.3568959	1	- 1.35689586789221	1	8.881784197001252E-016	29
	3.0489173	1	3.048917339522304	1	2.220446049250313E-015	31
3	-5.0000000	1	- 5.000000000000001	1	2.664535259100376E-015	29
	- 4.0000000	1	- 4.000000000000001	1	2.997602166487923E-015	31
	4.0000000	1	4	1	6.661338147750939E-015	28
	5.0000000	1	5	1	3.33066907387547E-015	29
4	0.33333333	1	0.3333333333333335	1	2.775557561562891E-016	32
	1.0000000	1	1	1	5.551115123125783E-016	32
	3.0000000	1	3	1	3.608224830031759E-015	31
	9.0000000	1	8.999999999999996	1	7.216449660063517E-015	30
5	1.0000000	1	1	1	7.771561172376096E-016	32
	3.0000000	1	3	1	1.554312234475219E-15	32
	9.0000000	1	9.000000000000002	1	5.051514762044462E-015	32
	27.000000	1	26.999999999999999	1	2.020605904817785E-014	30
6	-2.0000000	2	-2	2	3.090426696772397E-006	1
	1.0000000	1	1	1	3.090426696827908E-006	1
	3.0000000	1	3	1	1.77635683940025E-015	18
	4.0000000	2	4	2	3.090426696772397E-006	1
ECUACIÓN #1 $x^3 - 5x + 1$	ECUACIÓN #2 $x^3 - 7x - 7$	ECUACIÓN #3 $x^4 - 41x^2 + 400$	ECUACIÓN #4 $3x^4 - 40x^3 + 130x^2 - 120x + 27$	ECUACIÓN #5 $x^4 - 40x^3 + 390x^2 - 1080x + 729$	ECUACIÓN #6 $x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	

Hay varias situaciones que debemos aclarar dentro del proceso que hemos realizado, para evitar malas interpretaciones de la acción del método de bisección de Bolzano.

En primer término, todas las ecuaciones que se muestran en la tabla **11** se resolvieron empleando un juego uniforme de parámetros de control. Así, todas las raíces fueron calculadas empleando, como cota máxima del error cometido en la aproximación de las raíces, un valor mínimo de los que puede asumir esta variable dentro del programa, es decir, un error $\varepsilon = 0.0000000000000001 = 10^{-15}$, o sea, una exactitud en el cálculo muy elevada. Por otro lado se permitió que el método iterara un número elevado de veces (100), de modo que tuviera oportunidad de aproximar las raíces de las ecuaciones con la precisión pedida.

Además, se emplearon como subintervalos iniciales, los generados por el método mixto de separación, contenidos en la tabla **10**, de modo que se emplearon unos valores iniciales muy buenos.

Los resultados que generó la aplicación del método fueron en todos los casos satisfactorios con respecto a los valores que consideramos exactos (los que surgen al aplicar el programa Eureka versión 1.0 sobre las mismas ecuaciones), de modo que podemos considerar que el programa es eficiente.

Además, en otro orden de cosas, debe observarse que el programa contiene una rutina, que no forma parte del método en sí, que le permitió determinar correctamente, en todos los casos, la multiplicidad de cada raíz.

A continuación, consideraremos la implementación en computadora del método de Newton-Raphson-Fourier.

3.6.3.2. MÉTODO DE NEWTON-RAPHSON-FOURIER.

La implementación de un programa que realice la aproximación de los ceros de una ecuación algebraica de raíces reales, por este método, presenta un cierto número de particularidades y limitaciones, que, a continuación, detallaremos:

En primer término, la etapa de supresión de las raíces múltiples de la ecuación, es obligatoria, ya que, en caso contrario, la sucesión que trate de aproximar una raíz de este tipo divergería, por efectos de la derivada tiende a cero y por tanto se anularía el denominador de la fórmula de iteración. En cambio, éste es un aspecto que no representa mayor problema en el método de bisección de Bolzano.

Así, pues, en este programa nuevamente adicionaremos el esquema desarrollado en el método de Sturm (convertir la ecuación original en otra con todas sus raíces simples), antes de aplicar, en sí, el método de Newton-Raphson-Fourier, y al igual que antes, dicha técnica limitará la acción del método a ecuaciones originales de, al menos, grado dos.

Nuevamente, el método será capaz de determinar la multiplicidad de cada raíz obtenida, al aplicar el criterio de las derivadas a éstas.

Por otro lado, como la concepción teórica del método es ahora de tipo constructivo al contrario del de bisección de Bolzano, podemos esperar que este programa sea más veloz que el anterior, en término del número de iteraciones empleadas para obtener las aproximaciones de las raíces. De este modo, nuevamente hemos establecido que, en un caso extremo, se aproximará las raíces con una precisión dentro del intervalo $[10^{-17}, 10^{-8})$, y, en consecuencia, este valor se debe complementar con un número elevado de iteraciones, para lograr la convergencia del método.

También exigimos que se suministren los subintervalos que, se supone, contienen a las raíces del polinomio. Posteriormente, el programa verificará si, en realidad, ellos las contienen, y, en caso contrario, se informará al usuario la anomalía.

También el programa verificará que la suma de las multiplicidades de cada raíz totalice el grado del polinomio original, y, en caso de que esto no se cumpla, lo informará al usuario.

Finalmente, como en los programas anteriores, el usuario podrá, si lo desea, repetir la ejecución del programa o aplicarlo sobre la ecuación previamente introducida o sobre otra ecuación, según el caso.

A continuación, pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO DE NEWTON-RAPHSON-FOURIER

```

10 REM Programa número 8.
20 REM Cálculo de las raíces reales de una
30 REM ecuación algebraica de coeficientes reales,
40 REM empleando el método de Newton-Raphson-Fourier.
50 REM Bloque número 1.
60 REM Presentación del programa.
70 KEY OFF
80 CLS
90 CLEAR
100 DIM TI$(5)
110 TI$(0)="Cálculo de las raíces reales de una"
120 TI$(1)="ecuación algebraica de coeficientes"
130 TI$(2)="reales, empleando el método de"
140 TI$(3)="Newton-Raphson-Fourier"
150 TI$(4)="Presione cualquier tecla para continuar"
160 ETIQ=4
170 GOSUB 3500
180 GOSUB 4000
190 REM Bloque número 2.
200 REM Carga del grado del polinomio.
```

```

210 CLS
220 CLEAR
230 DIM TI$(5)
240 TI$(0)="Por favor, introduzca el grado del"
250 TI$(1)="polinomio (N) (El método requiere que"
260 TI$(2)="el polinomio sea de grado mayor o igual"
270 TI$(3)="que dos o no funcionará correctamente)"
280 TI$(4)="y pulse ``Enter'' para continuar"
290 ETIQ=4
300 GOSUB 3500
310 PRINT
320 PRINT
330 INPUT "N=",N
340 IF N>=2 AND N=FIX(N) THEN 420
350 CLS
360 TI$(0)="La información introducida es incorrecta"
370 TI$(1)="Presione cualquier tecla para continuar"
380 ETIQ=1
390 GOSUB 3500
400 GOSUB 4000
410 GOTO 210
420 REM Bloque número 3.
430 REM Dimensiones de los vectores y matrices iniciales.
440 DIM C#(N-1),D#(N),I#(N,1),P#(N)
450 DIM M#(N),N#(N),POLY#(N),Q#(N+1,N)
460 GR=N
470 REM Bloque número 4.
480 REM Carga de los coeficientes del polinomio
490 REM y cálculo del mayor número de decimales presente
500 REM en dichos coeficientes.
510 CLS
520 TI$(0)="Introduzca los coeficientes del"
530 TI$(1)="polinomio, en orden descendente y pulse"
540 TI$(2)="``Enter'' para continuar, en cada caso"
550 TI$(3)="Presione cualquier tecla para continuar"
560 ETIQ=3
570 GOSUB 3500
580 GOSUB 4000
590 CLS
600 FOR I=N TO 0 STEP -1
610 PRINT "Introduzca el coeficiente de X^";I;
620 INPUT "=",POLY#(I)
630 VAR$=STR$(POLY#(I))
640 LONG=LEN(VAR$)
650 FOR J=1 TO LONG
660 CARC$=MID$(VAR$,J,1)
670 IF CARC$="." THEN POT=LONG-J ELSE NEXT J

```

```

680 IF I=N THEN NU=POT
690 IF NU>POT THEN 720
700 NU=POT
710 P#(I)=POLY#(I)
720 PRINT
730 PRINT
740 NEXT I
750 IF P#(N)<>0# AND P#(0)<>0# THEN 850
760 CLS
770 TI$(0)="El polinomio introducido puede simplificarse,"
780 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+", "
790 TI$(2)="ni el de X^ 0, pueden ser nulos"
800 TI$(3)="Presione cualquier tecla para continuar"
810 ETIQ=3
820 GOSUB 3500
830 GOSUB 4000
840 GOTO 210
850 REM Bloque número 5.
860 REM Carga de la tolerancia y número máximo de
870 REM iteraciones que realizará el método.
880 CLS
890 N=GR
900 TI$(0)="Introduzca la cota máxima al error (PO) "
910 TI$(1)="que se permitirá en la aproximación de"
920 TI$(2)="las raíces del polinomio"
930 ETIQ=2
940 GOSUB 3500
950 PRINT
960 PRINT
970 INPUT "Valor ( $10^{-17} \leq PO < 10^{-8}$ ) (PO) =  $10^$  ",PO#
980 CLS
990 TI$(0)="Introduzca el número máximo de iteraciones"
1000 TI$(1)="a realizar (LI) al aplicar el método sobre"
1010 TI$(2)="los subintervalos que acotan a las raíces"
1020 ETIQ=2
1030 GOSUB 3500
1040 PRINT
1050 PRINT
1060 INPUT "Número de iteraciones (LI) =",LI
1070 PO#=CDBL( $10^{PO\#}$ )
1080 IF PO#<CDBL( $10^{-17}$ ) OR PO#>CDBL( $10^{-8}$ ) THEN 1110
1090 IF LI<=0 THEN 1110
1100 IF LI=FIX(LI) THEN 1180
1110 CLS
1120 TI$(0)="La información introducida es incorrecta"
1130 TI$(1)="Presione cualquier tecla para continuar"
1140 ETIQ=1

```

```

1150 GOSUB 3500
1160 GOSUB 4000
1170 GOTO 850
1180 REM Bloque número 6.
1190 REM Carga del número de subintervalos que acotan
1200 REM a las raíces simples del polinomio.
1210 CLS
1220 TI$(0)="Por favor, introduzca el número de"
1230 TI$(1)="subintervalos que acotan a las raíces"
1240 TI$(2)="simples del polinomio (IN), y pulse"
1250 TI$(3)="`Enter` para continuar"
1260 ETIQ=3
1270 GOSUB 3500
1280 PRINT
1290 PRINT
1300 INPUT "Número de subintervalos ( $0 < IN \leq N$ ) (IN)=",IN
1310 IF IN>=1 AND IN=FIX(IN) AND IN<=N THEN 1390
1320 CLS
1330 TI$(0)="La información introducida es incorrecta"
1340 TI$(1)="Presione cualquier tecla para continuar"
1350 ETIQ=1
1360 GOSUB 3500
1370 GOSUB 4000
1380 GOTO 1210
1390 REM Bloque número 7.
1400 REM Dimensión de la matriz de subintervalos iniciales.
1410 ERASE I#
1420 DIM I#(IN,1)
1430 REM Bloque número 8.
1440 REM Carga de los subintervalos que acotan
1450 REM a las raíces simples del polinomio.
1460 CLS
1470 TI$(0)="Introduzca los extremos superior e inferior"
1480 TI$(1)="de los subintervalos que acotan las raíces"
1490 TI$(2)="de la ecuación (A, B) y pulse `Enter`"
1500 TI$(3)="para continuar, en cada caso"
1510 TI$(4)="Presione cualquier tecla para continuar"
1520 ETIQ=4
1530 GOSUB 3500
1540 GOSUB 4000
1550 CLS
1560 FOR I=1 TO IN
1570 PRINT
1580 PRINT
1590 PRINT "Extremo inferior del ";
1600 PRINT "subintervalo número";I;":  A";I;
1610 INPUT "=",I#(I,0)

```

```

1620 PRINT
1630 PRINT
1640 PRINT "Extremo superior del ";
1650 PRINT "subintervalo número";I;":  B";I;
1660 INPUT "=",I#(I,1)
1670 NEXT I
1680 FOR I=1 TO IN
1690 IF I#(I,0)>I#(I,1) THEN 1750
1700 NEXT I
1710 FOR I=1 TO IN-1
1720 IF I#(I,1)>I#(I+1,0) THEN 1750
1730 NEXT I
1740 GOTO 1820
1750 CLS
1760 TI$(0)="La información introducida es incorrecta"
1770 TI$(1)="Presione cualquier tecla para continuar"
1780 ETIQ=1
1790 GOSUB 3500
1800 GOSUB 4000
1810 GOTO 1460
1820 REM Bloque número 9.
1830 REM Inicialización de los polinomios de trabajo y
1840 REM supresión de los decimales (si los hay), de los
1850 REM coeficientes del polinomio.
1860 CLS
1870 IF NU=0 THEN 1910
1880 FOR I=0 TO N
1890 P#(I)=POLY#(I)*CDBL(10^NU)
1900 NEXT I
1910 FOR I=0 TO N
1920 N#(I)=P#(I)
1930 NEXT I
1940 MULTI=0
1950 REM Bloque número 10.
1960 REM Simplificar (si es posible), los coeficientes
1970 REM del polinomio.
1980 FOR I=0 TO N
1990 M#(I)=N#(I)
2000 NEXT I
2010 M=N
2020 GOSUB 4500
2030 GOSUB 5000
2040 FOR I=0 TO N
2050 N#(I)=M#(I)
2060 NEXT I
2070 REM Bloque número 11.
2080 REM Calcular y simplificar la derivada del polinomio.

```

```
2090 M=N-1
2100 FOR I=0 TO M
2110 M#(I)=CDBL(I+1)*N#(I+1)
2120 NEXT I
2130 GOSUB 4500
2140 GOSUB 5000
2150 REM Bloque número 12.
2160 REM Reducir la ecuación original a una con raíces
2170 REM simples.
2180 FOR I=0 TO M
2190 D#(I)=M#(I)
2200 NEXT I
2210 P=N
2220 GOSUB 6000
2230 FOR K=P-2 TO 0 STEP -1
2240 IF N#(K)<>0# THEN 2340
2250 NEXT K
2260 GOSUB 6500
2270 P=N
2280 GOSUB 6000
2290 FOR I=P-M TO 0 STEP -1
2300 N#(I)=C#(I)
2310 NEXT I
2320 N=P-M
2330 GOTO 2380
2340 IF P=2 THEN 2370
2350 GOSUB 7000
2360 GOTO 2220
2370 GOSUB 6500
2380 REM Bloque número 13.
2390 REM Cálculo de las raíces del polinomio.
2400 MULTI=0
2410 FOR K=1 TO IN
2420 CLS
2430 MULT=0
2440 MAXA#=0#
2450 MAXB#=0#
2460 DES#=0#
2470 ERRMAX#=0#
2480 D2MAX#=0#
2490 DFHJ#=0#
2500 GOSUB 7500
2510 MULTI=MULTI+MULT
2520 NEXT K
2530 IF MULTI=GR THEN 2620
2540 CLS
2550 TI$(0)="El programa no ha podido determinar"
```



```

2560 TI$(1)="correctamente la multiplicidad de cada"
2570 TI$(2)="raíz de la ecuación"
2580 TI$(3)="Presione cualquier tecla para continuar"
2590 ETIQ=3
2600 GOSUB 3500
2610 GOSUB 4000
2620 REM Bloque número 14.
2630 REM Fin del programa.
2640 CLS
2650 TI$(0)="¿Desea ejecutar nuevamente el programa?"
2660 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
2670 TI$(2)="en caso contrario, cualquier otra tecla para"
2680 TI$(3)="finalizar la ejecución del programa"
2690 ETIQ=3
2700 GOSUB 3500
2710 GOSUB 4000
2720 IF Z$<>"1" THEN 2830
2730 CLS
2740 TI$(0)="¿Desea realizar los cálculos sobre el mismo"
2750 TI$(1)="polinomio? Si es así, pulse ``1'' para"
2760 TI$(2)="hacerlo, y, en caso contrario, cualquier otra"
2770 TI$(3)="tecla para cargar todos los datos nuevamente"
2780 ETIQ=3
2790 GOSUB 3500
2800 GOSUB 4000
2810 IF Z$="1" THEN 850 ELSE GOSUB 11500
2820 GOTO 200
2830 KEY ON
2840 CLS
2850 GOSUB 12000
2860 STOP
3000 REM Subrutinas del programa.
3500 REM Subrutina número 1.
3510 REM Impresión centrada de las etiquetas del
3520 REM programa, por página.
3530 FOR I=0 TO ETIQ
3540 FOR J=1 TO 3
3550 PRINT
3560 NEXT J
3570 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
3580 NEXT I
3590 RETURN
4000 REM Subrutina número 2.
4010 REM Paralización del avance del programa mientras
4020 REM el usuario lee los mensajes.
4030 Z$=""
4040 WHILE Z$=""

```

```
4050 Z$=INKEY$
4060 WEND
4070 RETURN
4500 REM Subrutina número 3.
4510 REM Determinar el menor coeficiente (en valor
4520 REM absoluto) no nulo de un polinomio.
4530 FOR I=0 TO M
4540 IF ABS(M#(I))=0# THEN NEXT I
4550 ME#=ABS(M#(I))
4560 FOR I=0 TO M
4570 IF M#(I)=0# THEN 4590 ELSE IF ME#<=ABS(M#(I)) THEN 4590
4580 ME#=ABS(M#(I))
4590 NEXT I
4600 RETURN
5000 REM Subrutina número 4.
5010 REM Descomponer el menor coeficiente del polinomio
5020 REM en sus factores primos.
5030 B#=2#
5040 GOTO 5060
5050 GOSUB 5500
5060 IF ME#=1# THEN 5170
5070 IF ME#/B#=INT(ME#/B#) THEN 5050 ELSE B#=3#
5080 C#=INT(SQR(ME#)+1#)
5090 IF B#<C# THEN 5130
5100 B#=ME#
5110 GOSUB 5500
5120 GOTO 5170
5130 IF ME#/B#=INT(ME#/B#) THEN 5150 ELSE B#=B#+2#
5140 GOTO 5080
5150 GOSUB 5500
5160 GOTO 5080
5170 RETURN
5500 REM Subrutina número 5.
5510 REM Simplificar los coeficientes del polinomio
5520 REM si es posible.
5530 ME#=ME#/B#
5540 FOR I=0 TO M
5550 IF M#(I)/B#<>INT(M#(I)/B#) THEN 5600
5560 IF I<>M THEN NEXT I
5570 FOR I=0 TO M
5580 M#(I)=M#(I)/B#
5590 NEXT I
5600 RETURN
6000 REM Subrutina número 6.
6010 REM División de polinomios con el algoritmo
6020 REM de los enteros.
6030 FOR I=0 TO M+1
```

```

6040 N#(I)=N#(I)*ABS(D#(M))
6050 NEXT I
6060 FOR I=P-M TO 0 STEP -1
6070 C#(I)=N#(I+M)/D#(M)
6080 FOR J=0 TO M
6090 N#(I+J)=N#(I+J)-C#(I)*D#(J)
6100 NEXT J
6110 IF I=0 THEN 6160
6120 FOR J=0 TO M
6130 N#(J)=N#(J)*ABS(D#(M))
6140 NEXT J
6150 NEXT I
6160 RETURN
6500 REM Subrutina número 7.
6510 REM Simplificar nuevamente el polinomio original
6520 REM y preparar su división por su M.C.D.
6530 FOR J=0 TO N
6540 N#(J)=P#(J)
6550 M#(J)=P#(J)
6560 NEXT J
6570 L=M
6580 M=N
6590 GOSUB 4500
6600 GOSUB 5000
6610 FOR J=0 TO N
6620 N#(J)=M#(J)
6630 NEXT J
6640 M=L
6650 RETURN
7000 REM Subrutina número 8.
7010 REM Preparar el cálculo del siguiente candidato a
7020 REM M.C.D. del polinomio original.
7030 FOR I=P-2 TO 0 STEP -1
7040 M#(I)=N#(I)
7050 NEXT I
7060 FOR J=P-2 TO 0 STEP -1
7070 IF M#(J)<>0# THEN 7090
7080 NEXT J
7090 M=J
7100 GOSUB 4500
7110 GOSUB 5000
7120 FOR I=P-1 TO 0 STEP -1
7130 N#(I)=D#(I)
7140 NEXT I
7150 FOR I=P-2 TO 0 STEP -1
7160 D#(I)=M#(I)
7170 NEXT I

```

```
7180 P=P-1
7190 RETURN
7500 REM Subrutina número 9.
7510 REM Método de Newton-Raphson-Fourier.
7520 FOR I=0 TO N+1
7530 FOR J=0 TO N
7540 IF I=0 THEN Q#(I,J)=N#(J) ELSE Q#(I,J)=0#
7550 NEXT J
7560 NEXT I
7570 P#=I#(K,0)
7580 FOR J=1 TO 3
7590 Q#(J,N)=Q#(0,N)
7600 NEXT J
7610 GRA=0
7620 FOR M=1 TO 3
7630 FOR J=N-1 TO GRA STEP -1
7640 Q#(M,J)=Q#(M,J+1)*P#+Q#(M-1,J)
7650 NEXT J
7660 GRA=GRA+1
7670 NEXT M
7680 FA#=Q#(1,0)
7690 Q#(3,2)=Q#(3,2)*2#
7700 P#=I#(K,1)
7710 FOR J=1 TO 3
7720 Q#(J,N)=Q#(0,N)
7730 NEXT J
7740 GRA=0
7750 FOR M=1 TO 3
7760 FOR J=N-1 TO GRA STEP -1
7770 Q#(M,J)=Q#(M,J+1)*P#+Q#(M-1,J)
7780 NEXT J
7790 GRA=GRA+1
7800 NEXT M
7810 FB#=Q#(1,0)
7820 Q#(3,2)=Q#(3,2)*2#
7830 IF FA#*FB#<=0# THEN 7900
7840 GOSUB 9000
7850 CLS
7860 GOSUB 3500
7870 GOSUB 4000
7880 CLS
7890 GOTO 8270
7900 IF Q#(1,0)*Q#(3,2)>0# THEN 7940
7910 H0#=I#(K,1)
7920 DES#=I#(K,0)
7930 GOTO 7960
7940 H0#=I#(K,0)
```

```

7950 DES#=I#(K,1)
7960 GOSUB 8500
7970 HI#=H0#
7980 FOR IT=1 TO LI
7990 FOR I=0 TO N
8000 Q#(0,I)=N#(I)
8010 NEXT I
8020 FOR J=1 TO 2
8030 Q#(J,N)=Q#(0,N)
8040 NEXT J
8050 GRA=0
8060 FOR M=1 TO 2
8070 FOR J=N-1 TO GRA STEP -1
8080 Q#(M,J)=Q#(M,J+1)*HI#+Q#(M-1,J)
8090 NEXT J
8100 GRA=GRA+1
8110 NEXT M
8120 FHI#=Q#(1,0)
8130 DFHI#=Q#(2,1)
8140 HJ#=HI#-FHI#/DFHI#
8150 ER#=ABS((HJ#-HI#)/HJ#)
8160 IF ER#<PO# THEN 8240
8170 HI#=HJ#
8180 NEXT IT
8190 GOSUB 9500
8200 CLS
8210 GOSUB 3500
8220 GOSUB 4000
8230 GOTO 8270
8240 R#=HJ#
8250 CLS
8260 GOSUB 10000
8270 RETURN
8500 REM Subrutina número 10.
8510 REM Cálculo del máximo de la segunda derivada de
8520 REM la función en los extremos del intervalo.
8530 MAXA#=ABS(Q#(3,2))
8540 P#=I#(K,1)
8550 FOR J=1 TO 3
8560 Q#(J,N)=Q#(0,N)
8570 NEXT J
8580 GRA=0
8590 FOR M=1 TO 3
8600 FOR J=N-1 TO GRA STEP -1
8610 Q#(M,J)=Q#(M,J+1)*P#+Q#(M-1,J)
8620 NEXT J
8630 GRA=GRA+1

```

```

8640 NEXT M
8650 Q#(3,2)=Q#(3,2)*2#
8660 MAXB#=ABS(Q#(3,2))
8670 IF MAXA#>MAXB# THEN D2MAX#=MAXA# ELSE D2MAX#=MAXB#
8680 RETURN
9000 REM Subrutina número 11.
9010 REM Mensaje de salida cuando no se encuentra la raíz.
9020 REM dentro del subintervalo obtenido.
9030 TI$(0)="La"+STR$(K)+"-ésima raíz de la ecuación"
9040 TI$(1)="no se pudo calcular, ya que el subintervalo"
9050 TI$(2)="que la acota no cumple las condiciones de"
9060 TI$(3)="bisección, se proseguirá a calcular las demás"
9070 TI$(4)="Presione cualquier tecla para continuar"
9080 ETIQ=4
9090 RETURN
9500 REM Subrutina número 12.
9510 REM Mensaje de salida cuando no se encuentra la raíz
9520 REM en el número indicado de iteraciones.
9530 TI$(0)="La"+STR$(K)+"-ésima raíz de la ecuación"
9540 TI$(1)="no se pudo calcular en"+STR$(LI)+" iteraciones"
9550 TI$(2)="se proseguirá a calcular las restantes"
9560 TI$(3)="Presione cualquier tecla para continuar"
9570 ETIQ=3
9580 RETURN
10000 REM Subrutina número 13.
10010 REM Cálculo de la multiplicidad de la raíz localizada.
10020 FOR I=0 TO GR
10030 Q#(0,I)=POLY#(I)
10040 NEXT I
10050 FOR J=1 TO GR+1
10060 Q#(J,GR)=Q#(0,GR)
10070 NEXT J
10080 GRA=0
10090 FOR M=1 TO GR
10100 FOR J=GR-1 TO GRA STEP -1
10110 Q#(M,J)=Q#(M,J+1)*R#+Q#(M-1,J)
10120 NEXT J
10130 GRA=GRA+1
10140 NEXT M
10150 FAC=1
10160 FOR P=2 TO GR+1
10170 FAC=FAC*P
10180 Q#(P,P-1)=Q#(P,P-1)*CDBL(FAC)
10190 NEXT P
10200 MU=1
10210 FOR M=2 TO GR+1
10220 IF ABS(Q#(M,MU))>10#*PO# THEN 10270

```

```

10230 MU=MU+1
10240 NEXT M
10250 MULT=GR
10260 GOTO 10290
10270 MULT=MU
10280 CLS
10290 GOSUB 10500
10300 GOSUB 11000
10310 GOSUB 3500
10320 GOSUB 4000
10330 RETURN
10500 REM Subrutina número 14.
10510 REM Cálculo de la cota del error cometido en la
10520 REM última iteración del método.
10530 P#=HJ#
10540 FOR J=1 TO 2
10550 Q#(J,N)=Q#(0,N)
10560 NEXT J
10570 GRA=0
10580 FOR M=1 TO 2
10590 FOR J=N-1 TO GRA STEP -1
10600 Q#(M,J)=Q#(M,J+1)*P#+Q#(M-1,J)
10610 NEXT J
10620 GRA=GRA+1
10630 NEXT M
10640 DFHJ#=Q#(2,1)
10650 ERRMAX#=ABS(D2MAX#*(DES#-HJ#)^2/(2*DFHJ#))
10660 RETURN
11000 REM Subrutina número 15.
11010 REM Mensajes de salida cuando se encuentra la
11020 REM raíz y su multiplicidad.
11030 TI$(0)="La"+STR$(K)+"-ésima raíz de la ecuación es"
11040 TI$(1)=STR$(R#)+", una cota del error cometido en la"
11050 TI$(2)="aproximación es:"+STR$(ERRMAX#)+", "
11060 TI$(3)="posee una multiplicidad igual a:"+STR$(MULT)
11070 TI$(4)="y se calculó en"+STR$(IT)+" iteraciones"
11080 TI$(5)="Presione cualquier tecla para continuar"
11090 ETIQ=5
11100 RETURN
11500 REM Subrutina número 16.
11510 REM Inicialización de las variables del programa
11520 REM para otra ejecución.
11530 ERASE C#,D#,I#,M#,N#,P#,POLY#,Q#
11540 RETURN
12000 REM Subrutina número 17.
12010 REM Fin del programa.
12020 END

```

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido aproximadas empleando el programa que acabamos de mostrar.

TABLA 12

ECUACIÓN	RAÍCES VERDADERAS	MULTIPLICIDAD VERDADERA	RAÍCES OBTENIDAS	MULTIPLICIDAD OBTENIDA	COTA DEL ERROR COMETIDO EN LA ÚLTIMA ITERACIÓN	ITERACIONES EMPLEADAS
1	-2.3300587	1	- 2.330058739567982	1	6.026246206847896E-014	3
	0.20163968	1	0.2016396757234047	1	7.188620323501215E-013	3
	2.1284191	1	2.128419063844577	1	5.497576391633488E-012	3
2	-1.6920215	1	- 1.692021471630096	1	4.496188772395555E-013	3
	- 1.3568959	1	- 1.356895867892209	1	4.889169509606158E-013	3
	3.0489173	1	3.048917339522305	1	2.39411277094094E-012	3
3	-5.0000000	1	- 5	1	1.101498849693314E-012	3
	- 4.0000000	1	- 4	1	1.567530027164235E-011	3
	4.0000000	1	4	1	2.127680968995136E-012	3
	5.0000000	1	5	1	1.101499717055094E-012	3
4	0.33333333	1	0.3333333333333333	1	2.011380448445825E-013	3
	1.0000000	1	1	1	1.79056000983491E-012	3
	3.0000000	1	3	1	1.304557451098407E-011	3
	9.0000000	1	9	1	6.299381232498747E-012	3
5	1.0000000	1	1	1	1.508535196299097E-013	3
	3.0000000	1	3	1	1.061075128718739E-012	3
	9.0000000	1	9	1	7.152251551353829E-011	3
	27.0000000	1	27	1	4.080036551834702E-012	3
6	-2.0000000	2	-2	2	7.231913484693509E-013	3
	1.0000000	1	1	1	6.715375907332756E-013	3
	3.0000000	1	3	1	1.578170607672866E-022	2
	4.0000000	2	4	2	6.295673559948161E-014	3
ECUACIÓN #1 $x^3 - 5x + 1$	ECUACIÓN #2 $x^3 - 7x - 7$	ECUACIÓN #3 $x^4 - 41x^2 + 400$	ECUACIÓN #4 $3x^4 - 40x^3 + 130x^2 - 120x + 27$	ECUACIÓN #5 $x^4 - 40x^3 + 390x^2 - 1080x + 729$	ECUACIÓN #6 $x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	

Debemos resaltar que, al ejecutar el programa del método de Newton-Raphson-Fourier, hemos empleado el mismo juego de parámetros fijos que se empleó al ejecutar el programa del método de bisección de Bolzano. Así, nuevamente, hemos calculado las raíces usando como cota máxima del error cometido en la aproximación de las raíces $\varepsilon = 0.000000000000001 = 10^{-15}$, y se permitió que el método iterara un número elevado de veces (100), para que pudiera aproximar las raíces de las ecuaciones con la precisión pedida.

Además, el empleo de un patrón común en las tablas **11** y **12**, nos permite resaltar las semejanzas y diferencias de la acción de ambos métodos.

Así, en primer término, ambos métodos han determinado aproximaciones muy buenas de las raíces de las ecuaciones cuando no las raíces exactas, así como del valor de su multiplicidad que en todos los casos resultó exacta; sin embargo, el método de Newton-Raphson-Fourier presentó una ligera ventaja sobre el método de bisección de Bolzano.

Por otro lado, es evidente que el método de Newton-Raphson-Fourier requirió sólo una fracción del número de iteraciones que emplea el método de bisección de Bolzano, de modo que el método de Newton-Raphson-Fourier resulta ser el más rápido de los dos, requiriendo, como promedio, la décima parte de las iteraciones empleadas por el otro. Así, en términos de la velocidad relativa de ambos métodos, el de Newton-Raphson-Fourier resultó más veloz.

Sin embargo, al comparar las cotas máximas del error cometido al aproximar la raíz exacta con el valor que surge de la última iteración, resultó que el método de bisección de Bolzano generó, en casi todos los casos, valores superiores que los dados por el método de Newton-Raphson-Fourier. Es más, casi siempre en los casos en que bisección originó la cota más grande, fue cuando éste realizó un número

menor de iteraciones que el de Newton-Raphson-Fourier (al realizar el proceso de bisección, encontraba que el punto medio de intervalo era, exactamente, la raíz que se deseaba aproximar). Este comportamiento, aparentemente incorrecto, puede atribuirse a varios factores:

En primer término, el método de bisección de Bolzano, acota continuamente a la raíz aproximada con un intervalo, de modo que, existencialmente, la restringe a un intervalo de la recta real. En cambio, Newton-Raphson-Fourier no la acota, y se atiene a las condiciones que verifica la función dentro del intervalo que contiene a las raíces.

Esto último es de gran trascendencia, ya que, por construcción, hemos asumido que los intervalos que suministrará el usuario serán buenos valores iniciales, y, como único control sobre éstos, hemos establecido que básicamente, verifiquen la condición de bisección de Bolzano. Esto es suficiente para el método de bisección de Bolzano y, de hecho, es todo lo que necesita para calcular las raíces.

En cambio, Newton-Raphson-Fourier requiere, además, que la función no cambie de concavidad dentro del intervalo, y el método asume de hecho que esto es así, pero no lo verifica, de modo que si introducimos un intervalo donde efectivamente no se cumpla la condición de concavidad, el método podría diverger. Además, el método de Newton-Raphson-Fourier asume que los valores máximo y mínimo de la función se dan en los extremos de intervalo introducido. Estas suposiciones son razonablemente buenas, para valores iniciales como los introducidos, sin embargo, no son exactas, de modo que puede resultar afectado el cálculo de la cota por ellas.

Además, el método de bisección de Bolzano involucra menos operaciones aritméticas y más simples que el de Newton, de modo que tiene menos posibilidades de que se propague el error por redondeo de una iteración a otra.

Sin embargo, este es un inconveniente menor, con respecto a los resultados realmente obtenidos, de modo que no debemos darle mayor importancia, y, en términos generales, podemos considerar que ambos métodos generan buenos resultados, y que el problema está resuelto satisfactoriamente (dentro de las limitaciones dadas).

A continuación, pues, consideraremos la construcción de dos programas de computadoras que implementen ambos métodos conjuntamente con el método mixto de separación de raíces, con el objeto de crear un sistema que, basado en todo lo anteriormente construido, reduzca a un mínimo la información inicial que debe suministrar el usuario, con el objeto de crear, finalmente, una aplicación de uso general.

El primer programa se implementará sobre el método de bisección de Bolzano, se denominará como método mixto **1** de aproximación (método mixto de separación-Bolzano), en tanto que el implementado sobre el método de Newton-Raphson se denominará método mixto **2** de aproximación (método mixto de separación-Newton-Raphson-Fourier).

3.6.3.3. MÉTODO MIXTO 1 DE APROXIMACIÓN: MÉTODO MIXTO DE SEPARACIÓN-BOLZANO.

La implementación de un programa que reuna al método mixto de separación con el método de bisección de Bolzano, visto con anterioridad, no es excesivamente más difícil que la implementación individual de cada método, de modo que sus limitaciones, serán básicamente la unión de las limitaciones de ambos métodos.

A continuación pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO MIXTO 1 DE APROXIMACIÓN: MÉTODO MIXTO DE SEPARACIÓN-BOLZANO

```

10 REM Programa número 9.
20 REM Cálculo de las raíces reales de una
30 REM ecuación algebraica de coeficientes reales,
40 REM empleando el método mixto de
50 REM separación-Bolzano.
60 REM Bloque número 1.
70 REM Presentación del programa.
80 KEY OFF
90 CLS
100 CLEAR
110 DIM TI$(5)
120 TI$(0)="Cálculo de las raíces reales de una"
130 TI$(1)="ecuación algebraica de coeficientes"
140 TI$(2)="reales, empleando el método mixto de"
150 TI$(3)="separación-Bolzano"
160 TI$(4)="Presione cualquier tecla para continuar"
170 ETIQ=4
180 GOSUB 5000
190 GOSUB 5500
200 REM Bloque número 2.
210 REM Carga del grado del polinomio.
220 CLS
230 CLEAR
240 DIM TI$(5)
250 TI$(0)="Por favor, introduzca el grado del"
260 TI$(1)="polinomio (N) (El método requiere que"
270 TI$(2)="el polinomio sea de grado mayor o igual"
280 TI$(3)="que dos o no funcionará correctamente)"
290 TI$(4)="y pulse ``Enter'' para continuar"
300 ETIQ=4
310 GOSUB 5000
320 PRINT
330 PRINT
340 INPUT "N=",N
350 IF N>=2 AND N=FIX(N) THEN 430
360 CLS
370 TI$(0)="La información introducida es incorrecta"
380 TI$(1)="Presione cualquier tecla para continuar"
390 ETIQ=1
400 GOSUB 5000
410 GOSUB 5500
420 GOTO 220

```

```

430 REM Bloque número 3.
440 REM Dimensiones de los vectores y matrices iniciales.
450 DIM B#(N,N),C#(N-1),D#(N),E#(1,N),I#(N,1)
460 DIM M#(N),N#(N),P#(N),POLY#(N),Q#(N+1,N)
470 GR=N
480 REM Bloque número 4.
490 REM Carga de los coeficientes del polinomio
500 REM y cálculo del mayor número de decimales presentes
510 REM en dichos coeficientes.
520 CLS
530 TI$(0)="Introduzca los coeficientes del"
540 TI$(1)="polinomio, en orden descendente y pulse"
550 TI$(2)=""`Enter' para continuar, en cada caso"
560 TI$(3)="Presione cualquier tecla para continuar"
570 ETIQ=3
580 GOSUB 5000
590 GOSUB 5500
600 CLS
610 FOR I=N TO 0 STEP -1
620 PRINT "Introduzca el coeficiente de X^";I;
630 INPUT "=",POLY#(I)
640 VAR$=STR$(POLY#(I))
650 LONG=LEN(VAR$)
660 FOR J=1 TO LONG
670 CARC$=MID$(VAR$,J,1)
680 IF CARC$="." THEN POT=LONG-J ELSE NEXT J
690 IF I=N THEN NU=POT
700 IF NU>POT THEN 730
710 NU=POT
720 P#(I)=POLY#(I)
730 PRINT
740 PRINT
750 NEXT I
760 IF P#(N)<>0# AND P#(0)<>0# THEN 860
770 CLS
780 TI$(0)="El polinomio introducido puede simplificarse,"
790 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+" ,"
800 TI$(2)="ni el de X^ 0, pueden ser nulos"
810 TI$(3)="Presione cualquier tecla para continuar"
820 ETIQ=3
830 GOSUB 5000
840 GOSUB 5500
850 GOTO 220
860 REM Bloque número 5.
870 REM Carga de la tolerancia y número máximo de
880 REM iteraciones que realizará el método.
890 CLS

```

```

900 N=GR
910 TI$(0)="Introduzca la cota máxima al error (PO)"
920 TI$(1)="que se permitirá en la aproximación de"
930 TI$(2)="las raíces del polinomio"
940 ETIQ=2
950 GOSUB 5000
960 PRINT
970 PRINT
980 INPUT "Valor ( $10^{-17} \leq PO < 10^{-8}$ ) (PO) =  $10^$  ",PO#
990 CLS
1000 TI$(0)="Introduzca el número máximo de iteraciones"
1010 TI$(1)="a realizar (LI) al aplicar el método sobre"
1020 TI$(2)="los subintervalos que acotan a las raíces"
1030 ETIQ=2
1040 GOSUB 5000
1050 PRINT
1060 PRINT
1070 INPUT "Número de iteraciones (LI) =",LI
1080 PO#=CDBL( $10^{PO\#}$ )
1090 IF PO#<CDBL( $10^{-17}$ ) OR PO#>CDBL( $10^{-8}$ ) THEN 1120
1100 IF LI<=0 THEN 1120
1110 IF LI=FIX(LI) THEN 1190
1120 CLS
1130 TI$(0)="La información introducida es incorrecta"
1140 TI$(1)="Presione cualquier tecla para continuar"
1150 ETIQ=1
1160 GOSUB 5000
1170 GOSUB 5500
1180 GOTO 860
1190 REM Bloque número 6.
1200 REM Inicialización del polinomio de trabajo.
1210 FOR I=0 TO N
1220 Q#(0,I)=CDBL((-1)^I)*POLY#(I)
1230 NEXT I
1240 CLS
1250 REM Bloque número 7.
1260 REM Verificar que el coeficiente de  $X^N$  es
1270 REM mayor que cero.
1280 GOSUB 6000
1290 REM Bloque número 8.
1300 REM Examinar si cero es o no cota inferior de las
1310 REM raíces del polinomio.
1320 CONTROL$=""
1330 GOSUB 6500
1340 REM Bloque número 9.
1350 REM Cálculo de una cota inferior de las raíces del
1360 REM polinomio por el método del mayor de los

```

```
1370 REM módulos de los coeficientes negativos de la
1380 REM ecuación.
1390 GOSUB 7000
1400 REM Bloque número 10.
1410 REM Refinamiento de la cota inferior de las raíces
1420 REM del polinomio por el método de Newton.
1430 SW=1
1440 GOSUB 7500
1450 REM Bloque número 11.
1460 REM Reinicialización del polinomio.
1470 FOR I=0 TO N
1480 Q#(0,I)=POLY#(I)
1490 NEXT I
1500 REM Bloque número 12.
1510 REM Verificar que el coeficiente de  $X^N$  es
1520 REM mayor que cero.
1530 GOSUB 6000
1540 REM Bloque número 13.
1550 REM Examinar si cero es o no cota superior de las
1560 REM raíces del polinomio.
1570 CONTROL$=""
1580 GOSUB 6500
1590 REM Bloque número 14.
1600 REM Cálculo de una cota superior de las raíces del
1610 REM polinomio por el método del mayor de los
1620 REM módulos de los coeficientes negativos de la
1630 REM ecuación.
1640 GOSUB 7000
1650 REM Bloque número 15.
1660 REM Refinamiento de la cota superior de las raíces
1670 REM del polinomio por el método de Newton.
1680 SW=0
1690 GOSUB 7500
1700 REM Bloque número 16.
1710 REM Reinicialización de los polinomios de trabajo y
1720 REM supresión de los decimales (si los hay) de los
1730 REM coeficientes del polinomio.
1740 IF NU=0 THEN 1780
1750 FOR I=0 TO N
1760 P#(I)=POLY#(I)*CDBL(10^NU)
1770 NEXT I
1780 FOR I=0 TO N
1790 N#(I)=P#(I)
1800 NEXT I
1810 REM Bloque número 17.
1820 REM Simplificar (si es posible) los coeficientes
1830 REM del polinomio.
```



```
1840 FOR I=0 TO N
1850 M#(I)=N#(I)
1860 NEXT I
1870 M=N
1880 GOSUB 8000
1890 GOSUB 8500
1900 FOR I=0 TO N
1910 N#(I)=M#(I)
1920 NEXT I
1930 REM Bloque número 18.
1940 REM Calcular y simplificar la derivada del polinomio.
1950 M=N-1
1960 FOR I=0 TO M
1970 M#(I)=CDBL(I+1)*N#(I+1)
1980 NEXT I
1990 GOSUB 8000
2000 GOSUB 8500
2010 REM Bloque número 19.
2020 REM Reducir la ecuación original a una con raíces
2030 REM simples.
2040 FOR I=0 TO M
2050 D#(I)=M#(I)
2060 NEXT I
2070 P=N
2080 GOSUB 9500
2090 FOR K=P-2 TO 0 STEP -1
2100 IF N#(K)<>0# THEN 2200
2110 NEXT K
2120 GOSUB 10000
2130 P=N
2140 GOSUB 9500
2150 FOR I=P-M TO 0 STEP -1
2160 N#(I)=C#(I)
2170 NEXT I
2180 N=P-M
2190 GOTO 2240
2200 IF P=2 THEN 2230
2210 GOSUB 10500
2220 GOTO 2080
2230 GOSUB 10000
2240 REM Bloque número 20.
2250 REM Cálculo de los polinomios de Sturm.
2260 FOR J=0 TO N
2270 B#(0,J)=N#(J)
2280 NEXT J
2290 L=1
2300 FOR J=N-1 TO 0 STEP -1
```

```
2310 B#(L,J)=N#(J+1)*CDBL(J+1)
2320 M#(J)=B#(L,J)
2330 NEXT J
2340 M=N-1
2350 GOSUB 8000
2360 GOSUB 8500
2370 FOR I=0 TO M
2380 D#(I)=M#(I)
2390 NEXT I
2400 P=N
2410 GOSUB 9500
2420 GOSUB 10500
2430 L=L+1
2440 FOR J=P-1 TO 0 STEP -1
2450 D#(J)=-D#(J)
2460 B#(L,J)=D#(J)
2470 NEXT J
2480 IF L=N THEN 2500
2490 GOTO 2410
2500 REM Bloque número 21.
2510 REM Cálculo de los subintervalos que separan
2520 REM las raíces de la ecuación.
2530 A#=CDBL(INF)
2540 B#=CDBL(SUP)
2550 LG#=B#-A#
2560 E#(0,0)=A#
2570 E#(0,N)=B#
2580 R=0
2590 GOSUB 11000
2600 IF C=N THEN 2660
2610 GOSUB 14500
2620 ETIQ=2
2630 GOSUB 5000
2640 GOSUB 5500
2650 GOTO 3830
2660 E#(1,0)=CDBL(C)
2670 A#=B#
2680 R=R+1
2690 GOSUB 11000
2700 E#(1,N)=CDBL(C)
2710 R=R+1
2720 I=0
2730 J=1
2740 K=N
2750 GOSUB 12000
2760 IF F=0 THEN 2830
2770 D=0
```

```
2780 FOR P=0 TO N-1
2790 V=CINT(E#(1,P)-E#(1,P+1))
2800 IF V=0 THEN 2840
2810 IF V=1 THEN 2880
2820 IF V>1 THEN 2980
2830 GOTO 3080
2840 IF D<>0 THEN 2860
2850 T=P
2860 D=D+1
2870 GOTO 2960
2880 IF D=0 THEN 2960
2890 GOSUB 12500
2900 I=T+1
2910 J=T+2
2920 K=N
2930 GOSUB 12000
2940 IF F=0 THEN 2770
2950 GOTO 2770
2960 NEXT P
2970 GOTO 3080
2980 IF D=0 THEN 3010
2990 GOSUB 12500
3000 P=T
3010 E#(1,P+V)=E#(1,P+1)
3020 E#(0,P+V)=E#(0,P+1)
3030 I=P
3040 J=P+1
3050 K=P+V
3060 GOSUB 12000
3070 GOTO 2770
3080 FOR J=0 TO N-1
3090 I#(J,0)=E#(0,J)
3100 I#(J,1)=E#(0,J+1)
3110 NEXT J
3120 TOL#=LG#/CDBL(10^6)
3130 FOR J=0 TO N-1
3140 A#=I#(J,0)
3150 GOSUB 13000
3160 B#=S#
3170 A#=I#(J,1)
3180 GOSUB 13000
3190 IF B#*S#>=0# THEN 3220
3200 NEXT J
3210 GOTO 3270
3220 GOSUB 14500
3230 ETIQ=2
3240 GOSUB 5000
```

```

3250 GOSUB 5500
3260 GOTO 3830
3270 MA#=I#(0,1)-I#(0,0)
3280 FOR I=0 TO N-1
3290 IF MA#=>I#(I,1)-I#(I,0) THEN 3310
3300 MA#=I#(I,1)-I#(I,0)
3310 NEXT I
3320 FOR J=1 TO LI
3330 FOR I=0 TO N-1
3340 A#=I#(I,0)
3350 GOSUB 13000
3360 B#=S#
3370 A#=(I#(I,0)+I#(I,1))/2#
3380 GOSUB 13000
3390 IF S#=0# THEN 3430
3400 IF B#*S#<0# THEN 3460
3410 I#(I,0)=A#
3420 GOTO 3470
3430 I#(I,0)=A#-MA#/3#
3440 I#(I,1)=A#+MA#/3#
3450 GOTO 3470
3460 I#(I,1)=A#
3470 NEXT I
3480 GOSUB 13500
3490 IF MA#<TOL# THEN 3650
3500 FOR I=0 TO N-1
3510 IF I#(I,0)<>I#(I,1) THEN 3540
3520 I#(I,0)=I#(I,0)-MA#/3#
3530 I#(I,1)=I#(I,1)+MA#/3#
3540 NEXT I
3550 NEXT J
3560 REM Bloque número 22.
3570 REM Salida cuando no se encuentran los subintervalos
3580 REM en el número establecido de iteraciones.
3590 GOSUB 14000
3600 ETIQ=3
3610 CLS
3620 GOSUB 5000
3630 GOSUB 5500
3640 GOTO 3830
3650 REM Bloque número 23.
3660 REM Cálculo de las raíces del polinomio.
3670 IN=N
3680 MULTI=0
3690 FOR K=0 TO IN-1
3695 CLS
3700 ERRMAX#=0#

```

```

3710 GOSUB 15000
3720 MULTI=MULTI+MULT
3730 NEXT K
3740 IF MULTI=GR THEN 3830
3750 CLS
3760 TI$(0)="El programa no ha podido determinar"
3770 TI$(1)="correctamente la multiplicidad de cada"
3780 TI$(2)="raíz de la ecuación"
3790 TI$(3)="Presione cualquier tecla para continuar"
3800 ETIQ=3
3810 GOSUB 5000
3820 GOSUB 5500
3830 REM Bloque número 24.
3840 REM Fin del programa.
3850 CLS
3860 TI$(0)="¿Desea ejecutar nuevamente el programa?"
3870 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
3880 TI$(2)="en caso contrario, cualquier otra tecla para"
3890 TI$(3)="finalizar la ejecución del programa"
3900 ETIQ=3
3910 GOSUB 5000
3920 GOSUB 5500
3930 IF Z$<>"1" THEN 4040
3940 CLS
3950 TI$(0)="¿Desea realizar los cálculos sobre el mismo"
3960 TI$(1)="polinomio? Si es así, pulse ``1'' para"
3970 TI$(2)="hacerlo, y, en caso contrario, cualquier otra"
3980 TI$(3)="tecla para cargar todos los datos nuevamente"
3990 ETIQ=3
4000 GOSUB 5000
4010 GOSUB 5500
4020 IF Z$="1" THEN 860 ELSE GOSUB 18500
4030 GOTO 210
4040 KEY ON
4050 CLS
4060 GOSUB 19000
4070 STOP
4500 REM Subrutinas del programa.
5000 REM Subrutina número 1.
5010 REM Impresión centrada de las etiquetas del
5020 REM programa, por página.
5030 FOR I=0 TO ETIQ
5040 FOR J=1 TO 3
5050 PRINT
5060 NEXT J
5070 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
5080 NEXT I

```

```

5090 RETURN
5500 REM Subrutina número 2.
5510 REM Paralización del avance del programa mientras
5520 REM el usuario lee los mensajes.
5530 Z$=""
5540 WHILE Z$=""
5550 Z$=INKEY$
5560 WEND
5570 RETURN
6000 REM Subrutina número 3.
6010 REM Verificar que el coeficiente de  $X^N$  es
6020 REM mayor que cero.
6030 IF Q#(0,N)>0# THEN 6070
6040 FOR I=0 TO N
6050 Q#(0,I)=-Q#(0,I)
6060 NEXT I
6070 RETURN
6500 REM Subrutina número 4.
6510 REM Examinar si cero es o no cota de las raíces
6520 REM del polinomio.
6530 FOR I=N-1 TO 0 STEP -1
6540 IF Q#(0,I)<0# THEN 6590
6550 NEXT I
6560 SUP=0
6570 CONTROL$="CERO"
6580 GOTO 6600
6590 P=I
6600 RETURN
7000 REM Subrutina número 5.
7010 REM Método del mayor de los módulos de los
7020 REM coeficientes negativos de la ecuación.
7030 IF CONTROL$="CERO" THEN 7120
7040 M=0
7050 FOR I=1 TO N
7060 IF Q#(0,M)<=Q#(0,I) THEN 7080
7070 M=I
7080 NEXT I
7090 G#=ABS(Q#(0,M))
7100 A#=Q#(0,N)
7110 SUPR=1+INT(1#+(G#/A#)^(1/(N-P)))
7120 RETURN
7500 REM Subrutina número 6.
7510 REM Método de Newton.
7520 FOR J=1 TO N+1
7530 Q#(J,N)=Q#(0,N)
7540 NEXT J
7550 FOR P=0 TO SUPR

```

```

7560 GRA=0
7570 FOR M=1 TO N
7580 FOR J=N-1 TO GRA STEP -1
7590 Q#(M,J)=Q#(M,J+1)*CDBL(P)+Q#(M-1,J)
7600 NEXT J
7610 GRA=GRA+1
7620 NEXT M
7630 J=0
7640 FOR M=1 TO N+1
7650 IF Q#(M,J)<=0# THEN 7690
7660 J=J+1
7670 NEXT M
7680 GOTO 7700
7690 NEXT P
7700 IF SW=1 THEN INF=-P ELSE SUP=P
7710 RETURN
8000 REM Subrutina número 7.
8010 REM Determinar el menor coeficiente (en valor
8020 REM absoluto) no nulo de un polinomio.
8030 FOR I=0 TO M
8040 IF ABS(M#(I))=0# THEN NEXT I
8050 ME#=ABS(M#(I))
8060 FOR I=0 TO M
8070 IF M#(I)=0# THEN 8090 ELSE IF ME#<=ABS(M#(I)) THEN 8090
8080 ME#=ABS(M#(I))
8090 NEXT I
8100 RETURN
8500 REM Subrutina número 8.
8510 REM Descomponer el menor coeficiente del polinomio
8520 REM en sus factores primos.
8530 B#=2#
8540 GOTO 8560
8550 GOSUB 9000
8560 IF ME#=1# THEN 8670
8570 IF ME#/B#=INT(ME#/B#) THEN 8550 ELSE B#=3#
8580 C#=INT(SQR(ME#)+1#)
8590 IF B#<C# THEN 8630
8600 B#=ME#
8610 GOSUB 9000
8620 GOTO 8670
8630 IF ME#/B#=INT(ME#/B#) THEN 8650 ELSE B#=B#+2#
8640 GOTO 8580
8650 GOSUB 9000
8660 GOTO 8580
8670 RETURN
9000 REM Subrutina número 9.
9010 REM Simplificar los coeficientes del polinomio

```

```
9020 REM si es posible.
9030 ME#=ME#/B#
9040 FOR I=0 TO M
9050 IF M#(I)/B#<>INT(M#(I)/B#) THEN 9100
9060 IF I<>M THEN NEXT I
9070 FOR I=0 TO M
9080 M#(I)=M#(I)/B#
9090 NEXT I
9100 RETURN
9500 REM Subrutina número 10.
9510 REM División de polinomios con el algoritmo
9520 REM de los enteros.
9530 FOR I=0 TO M+1
9540 N#(I)=N#(I)*ABS(D#(M))
9550 NEXT I
9560 FOR I=P-M TO 0 STEP -1
9570 C#(I)=N#(I+M)/D#(M)
9580 FOR J=0 TO M
9590 N#(I+J)=N#(I+J)-C#(I)*D#(J)
9600 NEXT J
9610 IF I=0 THEN 9660
9620 FOR J=0 TO M
9630 N#(J)=N#(J)*ABS(D#(M))
9640 NEXT J
9650 NEXT I
9660 RETURN
10000 REM Subrutina número 11.
10010 REM Simplificar nuevamente el polinomio original
10020 REM y preparar su división por su M.C.D.
10030 FOR J=0 TO N
10040 N#(J)=P#(J)
10050 M#(J)=P#(J)
10060 NEXT J
10070 L=M
10080 M=N
10090 GOSUB 8000
10100 GOSUB 8500
10110 FOR J=0 TO N
10120 N#(J)=M#(J)
10130 NEXT J
10140 M=L
10150 RETURN
10500 REM Subrutina número 12.
10510 REM Preparar el cálculo del siguiente candidato a
10520 REM M.C.D. del polinomio original.
10530 FOR I=P-2 TO 0 STEP -1
10540 M#(I)=N#(I)
```



```

10550 NEXT I
10560 FOR J=P-2 TO 0 STEP -1
10570 IF M#(J)<>0# THEN 10590
10580 NEXT J
10590 M=J
10600 GOSUB 8000
10610 GOSUB 8500
10620 FOR I=P-1 TO 0 STEP -1
10630 N#(I)=D#(I)
10640 NEXT I
10650 FOR I=P-2 TO 0 STEP -1
10660 D#(I)=M#(I)
10670 NEXT I
10680 P=P-1
10690 RETURN
11000 REM Subrutina número 13.
11010 REM Cálculo del número de variaciones de la sucesión
11020 REM de Sturm en el valor A.
11030 I=0
11040 FOR J=N TO 1 STEP -1
11050 N#(J)=B#(I,J)
11060 FOR M=J-1 TO 0 STEP -1
11070 N#(M)=A#*N#(M+1)+B#(I,M)
11080 NEXT M
11090 M#(I)=N#(0)
11100 I=I+1
11110 NEXT J
11120 IF M#(0)<>0# THEN 11350
11130 IF R<>0 THEN 11240
11140 CLS
11150 TI$(0)="La cota inferior obtenida "+STR$(A#)
11160 TI$(1)="es una raíz de la ecuación; el programa"
11170 TI$(2)="no es capaz de calcular cotas para las"
11180 TI$(3)="raíces de la ecuación"
11190 TI$(4)="Presione cualquier tecla para continuar"
11200 ETIQ=4
11210 GOSUB 5000
11220 GOSUB 5500
11230 GOTO 4040
11240 IF R<>1 THEN 11350
11250 CLS
11260 TI$(0)="La cota superior obtenida "+STR$(B#)
11270 TI$(1)="es una raíz de la ecuación; el programa"
11280 TI$(2)="no es capaz de calcular cotas para las"
11290 TI$(3)="raíces de la ecuación"
11300 TI$(4)="Presione cualquier tecla para continuar"
11310 ETIQ=4

```

```

11320 GOSUB 5000
11330 GOSUB 5500
11340 GOTO 4040
11350 M#(N)=B#(N,0)
11360 C=0
11370 FOR J=0 TO N-1
11380 IF M#(J)=0# THEN 11460
11390 K=J+1
11400 IF M#(K)=0# THEN 11500
11410 IF M#(J)>0# THEN IF M#(K)>0# THEN 11440
11420 IF M#(J)<0# THEN IF M#(K)<0# THEN 11440
11430 C=C+1
11440 IF K=N THEN 11520
11450 GOTO 11470
11460 IF J=0 THEN 11480
11470 J=K-1
11480 NEXT J
11490 GOTO 11520
11500 K=K+1
11510 IF K<=N THEN 11400
11520 RETURN
12000 REM Subrutina número 14.
12010 REM Verificar que el número de variaciones sea
12020 REM el correcto en los intervalos, y si es necesario,
12030 REM se reeligen los extremos del subintervalo.
12040 V=CINT(E#(1,I)-E#(1,K))
12050 F=0
12060 IF V=1 THEN 12340
12070 F=1
12080 L#=E#(0,K)-E#(0,I)
12090 LO#=L#/CDBL(V)
12100 E=J
12110 E#(0,E)=E#(0,E-1)+LO#
12120 A#=E#(0,E)
12130 GOSUB 11000
12140 E#(1,E)=CDBL(C)
12150 E=E+1
12160 IF E<=K-1 THEN 12110
12170 U=0
12180 A#=E#(0,U)
12190 GOSUB 13000
12200 IF S#<>0# THEN 12250
12210 E#(0,U)=E#(0,U)+(E#(0,U+1)-E#(0,U))/3#
12220 A#=E#(0,U)
12230 GOSUB 11000
12240 E#(1,U)=CDBL(C)
12250 U=U+1

```

```

12260 IF U<=N-1 THEN 12180
12270 A#=E#(0,N)
12280 GOSUB 13000
12290 IF S#<>0# THEN 12340
12300 E#(0,N)=E#(0,N)+1#/3#
12310 A#=E#(0,N)
12320 GOSUB 11000
12330 E#(1,N)=CDBL(C)
12340 RETURN
12500 REM Subrutina número 15.
12510 REM Substituye los extremos de los subintervalos
12520 REM que no poseen raíces, por otros que si las
12530 REM poseen.
12540 E#(1,T)=E#(1,P)
12550 E#(1,T+1)=E#(1,P+1)
12560 E#(0,T)=E#(0,P)
12570 E#(0,T+1)=E#(0,P+1)
12580 RETURN
13000 REM Subrutina número 16.
13010 REM Evaluación del polinomio reducido en el valor A
13020 REM para reducir la longitud de los subintervalos que
13030 REM separan a las raíces.
13040 N#(N)=B#(0,N)
13050 FOR M=N-1 TO 0 STEP -1
13060 N#(M)=A#*N#(M+1)+B#(0,M)
13070 NEXT M
13080 S#=N#(M+1)
13090 RETURN
13500 REM Subrutina número 17.
13510 REM Cálculo de la longitud del mayor de los
13520 REM subintervalos que separan las raíces.
13530 MA#=0#
13540 FOR I=0 TO N-1
13550 LO#=I#(I,1)-I#(I,0)
13560 IF MA#>=LO# THEN 13580
13570 MA#=LO#
13580 NEXT I
13590 RETURN
14000 REM Subrutina número 18.
14010 REM Mensajes de salida cuando no se encuentran
14020 REM los subintervalos.
14030 TI$(0)="Los subintervalos que acotan a las raíces"
14040 TI$(1)="simples de la ecuación no se han podido"
14050 TI$(2)="calcular, en "+STR$(LI)+" iteraciones"
14060 TI$(3)="Presione cualquier tecla para continuar"
14070 RETURN
14500 REM Subrutina número 19.

```

```

14510 REM Mensajes de salida cuando la ecuación posee
14520 REM raíces complejas.
14530 TI$(0)="La ecuación posee raíces complejas, por lo"
14540 TI$(1)="que el método no le es aplicable"
14550 TI$(2)="Presione cualquier tecla para continuar"
14560 RETURN
15000 REM Subrutina número 20.
15010 REM Método de bisección de Bolzano.
15020 FOR I=0 TO N+1
15030 FOR J=0 TO N
15040 IF I=0 THEN Q#(I,J)=B#(0,J) ELSE Q#(I,J)=0#
15050 NEXT J
15060 NEXT I
15070 MULT=0
15080 IA#=I#(K,0)
15090 IB#=I#(K,1)
15100 FOR IT=1 TO LI
15110 P#=IA#
15120 Q#(1,N)=Q#(0,N)
15130 FOR J=N-1 TO 0 STEP -1
15140 Q#(1,J)=Q#(1,J+1)*P#+Q#(0,J)
15150 NEXT J
15160 FA#=Q#(1,0)
15170 PM#=(IA#+IB#)/2#
15180 P#=PM#
15190 FOR J=N-1 TO 0 STEP -1
15200 Q#(1,J)=Q#(1,J+1)*P#+Q#(0,J)
15210 NEXT J
15220 FPM#=Q#(1,0)
15230 P#=IB#
15240 FOR J=N-1 TO 0 STEP -1
15250 Q#(1,J)=Q#(1,J+1)*P#+Q#(0,J)
15260 NEXT J
15270 FB#=Q#(1,0)
15280 IF FPM#=0# THEN 15480
15290 IF FA#*FPM#>0# THEN 15320
15300 IB#=PM#
15310 GOTO 15340
15320 IF FPM#*FB#>0# THEN 15350
15330 IA#=PM#
15340 GOTO 15400
15350 CLS
15360 GOSUB 16000
15370 GOSUB 5500
15380 CLS
15390 GOTO 15510
15400 ER#=ABS((IB#-IA#)/PM#)

```

```

15410 IF ER#<PO# THEN 15480
15420 NEXT IT
15430 GOSUB 16500
15440 CLS
15450 GOSUB 5000
15460 GOSUB 5500
15470 GOTO 15510
15480 R#=PM#
15490 CLS
15500 GOSUB 17000
15510 RETURN
16000 REM Subrutina número 21.
16010 REM Mensaje de salida cuando no se encuentra la raíz
16020 REM dentro de subintervalo calculado.
16030 TI$(0)="La"+STR$(K+1)+"-ésima raíz de la ecuación"
16040 TI$(1)="no se pudo calcular, ya que el subintervalo"
16050 TI$(2)="que la acota no cumple las condiciones de"
16060 TI$(3)="Bisección, se procederá a calcular las demás"
16070 ETIQ=4
16080 RETURN
16500 REM Subrutina número 22.
16510 REM Mensaje de salida cuando no se encuentra la raíz
16520 REM dentro del número establecido de iteraciones.
16530 TI$(0)="La"+STR$(K+1)+"-ésima raíz de la ecuación"
16540 TI$(1)="no pudo calcularse en"+STR$(LI)+" iteraciones"
16550 TI$(2)="se procederá a calcular las restantes"
16560 TI$(3)="Presione cualquier tecla para continuar"
16570 ETIQ=3
16580 RETURN
17000 REM Subrutina número 23.
17010 REM Cálculo de la multiplicidad de la raíz localizada.
17020 FOR I=0 TO GR
17030 Q#(0,I)=POLY#(I)
17040 NEXT I
17050 MUL=0
17060 FOR J=1 TO GR+1
17070 Q#(J,GR)=Q#(0,GR)
17080 NEXT J
17090 GRA=0
17100 FOR M=1 TO GR
17110 FOR J=GR-1 TO GRA STEP -1
17120 Q#(M,J)=Q#(M,J+1)*R#+Q#(M-1,J)
17130 NEXT J
17140 GRA=GRA+1
17150 NEXT M
17160 FAC=1
17170 FOR P=2 TO GR+1

```

```

17180 FAC=FAC*P
17190 Q#(P,P-1)=Q#(P,P-1)*CDBL(FAC)
17200 NEXT P
17210 MU=1
17220 FOR M=2 TO GR+1
17230 IF ABS(Q#(M,MU))>10000#*PO# THEN 17280
17240 MU=MU+1
17250 NEXT M
17260 MULT=GR
17270 GOTO 17290
17280 MULT=MU
17290 CLS
17300 GOSUB 17500
17310 GOSUB 18000
17320 GOSUB 5000
17330 GOSUB 5500
17340 CLS
17350 RETURN
17500 REM Subrutina número 24.
17510 REM Cálculo de la cota del error cometido en la
17520 REM última iteración del método.
17530 ERRMAX#=ABS((I#(K,1)-I#(K,0))/(2^IT))
17540 RETURN
18000 REM Subrutina número 25.
18010 REM Mensajes de salida cuando se encuentra la
18020 REM raíz y su multiplicidad.
18030 TI$(0)="La"+STR$(K+1)+"-ésima raíz de la ecuación es"
18040 TI$(1)=STR$(R#)+", una cota del error cometido en la"
18050 TI$(2)="aproximación es:"+STR$(ERRMAX#)+", "
18060 TI$(3)="posee una multiplicidad igual a:"+STR$(MULT)
18070 TI$(4)="y se calculó en"+STR$(IT)+" iteraciones"
18080 TI$(5)="Presione cualquier tecla para continuar"
18090 ETIQ=5
18100 RETURN
18500 REM Subrutina número 26.
18510 REM Inicialización de las variables del programa
18520 REM para otra ejecución.
18530 ERASE B#,C#,D#,E#,I#,M#,N#,P#,POLY#,Q#
18540 RETURN
19000 REM Subrutina número 27.
19010 REM Fin del programa.
19020 END

```

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido aproximadas empleando el programa que acabamos de mostrar.

TABLA 13

ECUACIÓN	RAÍCES VERDADERAS	MULTIPLICIDAD VERDADERA	RAÍCES OBTENIDAS	MULTIPLICIDAD OBTENIDA	COTA DEL ERROR COMETIDO EN LA ÚLTIMA ITERACIÓN	ITERACIONES EMPLEADAS
1	-2.3300587	1	- 2.330058739567983	1	1.77635683940025E-015	31
	0.20163968	1	0.2016396757234046	1	1.110223024625157E-016	35
	2.1284191	1	2.128419063844577	1	1.77635683940025E-015	31
2	-1.6920215	1	- 1.692021471630096	1	8.881784197001252E-016	29
	- 1.3568959	1	- 1.35689586789221	1	8.881784197001252E-016	29
	3.0489173	1	3.048917339522304	1	2.220446049250313E-015	31
3	-5.0000000	1	- 5.000000000000001	1	2.664535259100376E-015	29
	- 4.0000000	1	- 4.000000000000001	1	2.997602166487923E-015	31
	4.0000000	1	4	1	6.661338147750939E-015	28
	5.0000000	1	5	1	3.33066907387547E-015	29
4	0.33333333	1	0.3333333333333335	1	2.775557561562891E-016	32
	1.0000000	1	1	1	5.551115123125783E-016	32
	3.0000000	1	3	1	3.608224830031759E-015	31
	9.0000000	1	8.999999999999996	1	7.216449660063517E-015	30
5	1.0000000	1	1	1	7.771561172376096E-016	32
	3.0000000	1	3	1	1.554312234475219E-15	32
	9.0000000	1	9.000000000000002	1	5.051514762044462E-015	32
	27.000000	1	26.999999999999999	1	2.020605904817785E-014	30
6	-2.0000000	2	-2	2	3.090426696772397E-006	1
	1.0000000	1	1	1	3.090426696827908E-006	1
	3.0000000	1	3	1	1.77635683940025E-015	18
	4.0000000	2	4	2	3.090426696772397E-006	1
ECUACIÓN #1 $x^3 - 5x + 1$	ECUACIÓN #2 $x^3 - 7x - 7$	ECUACIÓN #3 $x^4 - 41x^2 + 400$	ECUACIÓN #4 $3x^4 - 40x^3 + 130x^2 - 120x + 27$	ECUACIÓN #5 $x^4 - 40x^3 + 390x^2 - 1080x + 729$	ECUACIÓN #6 $x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	

Al comparar las tablas **11** y **13**, es evidente que se obtuvieron resultados iguales. Esto ocurre así, porque la sección de separación de raíces del programa del método mixto 1 de aproximación fue programada para que, por defecto, use como longitud máxima de los subintervalos que acotarán a las raíces de la ecuación, al valor $\varepsilon = 0.0001 = 10^{-4}$, que fue el empleado para calcular la tabla **11**, de modo que los intervalos que generó internamente el método, son los mismos que se emplearon como valores iniciales para calcular la tabla **11** (se muestran en la tabla **10**).

De este modo, resulta que ambos programas generan respuestas similares a las obtenidas por el programa del método de bisección de Bolzano. Sin embargo, con la última técnica no es necesario suministrar los subintervalos que acotan a las raíces, de modo que dicha técnica resulta ser más eficiente en general.

A continuación, implementaremos en computadora el método mixto **2** de aproximación (método mixto de separación-Newton-Raphson-Fourier).

3.6.3.4. MÉTODO MIXTO 2 DE APROXIMACIÓN: MÉTODO MIXTO DE SEPARACIÓN-NEWTON-RAPHSON-FOURIER.

La implementación de un programa que reuna al método mixto de separación con el método de Newton-Raphson-Fourier al igual que ocurrió en el caso del método de bisección de Bolzano, no resulta excesivamente más difícil que la implementación individual de cada método, de modo que, similarmente, sus limitaciones serán básicamente la unión de las limitaciones de ambos métodos. Así, pues, no haremos más comentarios respecto al tema.

A continuación pues, presentamos el listado del programa que implementamos.

PROGRAMA DEL MÉTODO MIXTO 2 DE APROXIMACIÓN: MÉTODO MIXTO DE SEPARACIÓN-NEWTON-RAPHSON- FOURIER

```

10 REM Programa número 10.
20 REM Cálculo de las raíces reales de una
30 REM ecuación algebraica de coeficientes reales,
40 REM empleando el método mixto de
50 REM separación-Newton-Raphson-Fourier.
60 REM Bloque número 1.
70 REM Presentación del programa.
80 KEY OFF
90 CLS
100 CLEAR
110 DIM TI$(5)
120 TI$(0)="Cálculo de las raíces reales de una"
130 TI$(1)="ecuación algebraica de coeficientes"
140 TI$(2)="reales, empleando el método mixto de"
150 TI$(3)="separación-Newton-Raphson-Fourier"
160 TI$(4)="Presione cualquier tecla para continuar"
170 ETIQ=4
180 GOSUB 5000
190 GOSUB 5500
200 REM Bloque número 2.
210 REM Carga del grado del polinomio.
220 CLS
230 CLEAR
240 DIM TI$(5)
250 TI$(0)="Por favor, introduzca el grado del"
260 TI$(1)="polinomio (N) (El método requiere que"
270 TI$(2)="el polinomio sea de grado mayor o igual"
280 TI$(3)="que dos o no funcionará correctamente)"
290 TI$(4)="y pulse ``Enter'' para continuar"
300 ETIQ=4
310 GOSUB 5000
320 PRINT
330 PRINT
340 INPUT "N=",N
350 IF N>=2 AND N=FIX(N) THEN 430
360 CLS
370 TI$(0)="La información introducida es incorrecta"
380 TI$(1)="Presione cualquier tecla para continuar"
390 ETIQ=1
400 GOSUB 5000

```

```

410 GOSUB 5500
420 GOTO 220
430 REM Bloque número 3.
440 REM Dimensiones de los vectores y matrices iniciales.
450 DIM B#(N,N),C#(N-1),D#(N),E#(1,N),I#(N,1)
460 DIM M#(N),N#(N),P#(N),POLY#(N),Q#(N+1,N)
470 GR=N
480 REM Bloque número 4.
490 REM Carga de los coeficientes del polinomio
500 REM y cálculo del mayor número de decimales presentes
510 REM en dichos coeficientes.
520 CLS
530 TI$(0)="Introduzca los coeficientes del"
540 TI$(1)="polinomio, en orden descendente y pulse"
550 TI$(2)="`Enter` para continuar, en cada caso"
560 TI$(3)="Presione cualquier tecla para continuar"
570 ETIQ=3
580 GOSUB 5000
590 GOSUB 5500
600 CLS
610 FOR I=N TO 0 STEP -1
620 PRINT "Introduzca el coeficiente de X^";I;
630 INPUT "=",POLY#(I)
640 VAR$=STR$(POLY#(I))
650 LONG=LEN(VAR$)
660 FOR J=1 TO LONG
670 CARC$=MID$(VAR$,J,1)
680 IF CARC$="." THEN POT=LONG-J ELSE NEXT J
690 IF I=N THEN NU=POT
700 IF NU>POT THEN 730
710 NU=POT
720 P#(I)=POLY#(I)
730 PRINT
740 PRINT
750 NEXT I
760 IF P#(N)<>0# AND P#(0)<>0# THEN 860
770 CLS
780 TI$(0)="El polinomio introducido puede simplificarse,"
790 TI$(1)="ya que ni el coeficiente de X^"+STR$(N)+", "
800 TI$(2)="ni el de X^ 0, pueden ser nulos"
810 TI$(3)="Presione cualquier tecla para continuar"
820 ETIQ=3
830 GOSUB 5000
840 GOSUB 5500
850 GOTO 220
860 REM Bloque número 5.
870 REM Carga de la tolerancia y número máximo de

```

```

880 REM iteraciones que realizará el método.
890 CLS
900 N=GR
910 TI$(0)="Introduzca la cota máxima al error (PO)"
920 TI$(1)="que se permitirá en la aproximación de"
930 TI$(2)="las raíces del polinomio"
940 ETIQ=2
950 GOSUB 5000
960 PRINT
970 PRINT
980 INPUT "Valor ( $10^{-17} \leq PO < 10^{-8}$ ) (PO) =  $10^$  ",PO#
990 CLS
1000 TI$(0)="Introduzca el número máximo de iteraciones"
1010 TI$(1)="a realizar (LI) al aplicar el método sobre"
1020 TI$(2)="los subintervalos que acotan las raíces"
1030 ETIQ=2
1040 GOSUB 5000
1050 PRINT
1060 PRINT
1070 INPUT "Número de iteraciones (LI) =",LI
1080 PO#=CDBL( $10^{PO\#}$ )
1090 IF PO#<CDBL( $10^{-17}$ ) OR PO#>=CDBL( $10^{-8}$ ) THEN 1120
1100 IF LI<=0 THEN 1120
1110 IF LI=FIX(LI) THEN 1190
1120 CLS
1130 TI$(0)="La información introducida es incorrecta"
1140 TI$(1)="Presione cualquier tecla para continuar"
1150 ETIQ=1
1160 GOSUB 5000
1170 GOSUB 5500
1180 GOTO 860
1190 REM Bloque número 6.
1200 REM Inicialización del polinomio de trabajo.
1210 FOR I=0 TO N
1220 Q#(0,I)=CDBL((-1)^I)*POLY#(I)
1230 NEXT I
1240 CLS
1250 REM Bloque número 7.
1260 REM Verificar que el coeficiente de  $X^N$  es
1270 REM mayor que cero.
1280 GOSUB 6000
1290 REM Bloque número 8.
1300 REM Examinar si cero es o no cota inferior de las
1310 REM raíces del polinomio.
1320 CONTROL$=""
1330 GOSUB 6500
1340 REM Bloque número 9.

```

```
1350 REM Cálculo de una cota inferior de las raíces del
1360 REM polinomio por el método del mayor de los
1370 REM módulos de los coeficientes negativos de la
1380 REM ecuación.
1390 GOSUB 7000
1400 REM Bloque número 10.
1410 REM Refinamiento de la cota inferior de las raíces
1420 REM del polinomio por el método de Newton.
1430 SW=1
1440 GOSUB 7500
1450 REM Bloque número 11.
1460 REM Reinicialización del polinomio.
1470 FOR I=0 TO N
1480 Q#(0,I)=POLY#(I)
1490 NEXT I
1500 REM Bloque número 12.
1510 REM Verificar que el coeficiente de  $X^N$  es
1520 REM mayor que cero.
1530 GOSUB 6000
1540 REM Bloque número 13.
1550 REM Examinar si cero es o no cota superior de las
1560 REM raíces del polinomio.
1570 CONTROL$=""
1580 GOSUB 6500
1590 REM Bloque número 14.
1600 REM Cálculo de una cota superior de las raíces del
1610 REM polinomio por el método del mayor de los
1620 REM módulos de los coeficientes negativos de la
1630 REM ecuación.
1640 GOSUB 7000
1650 REM Bloque número 15.
1660 REM Refinamiento de la cota superior de las raíces
1670 REM del polinomio por el método de Newton.
1680 SW=0
1690 GOSUB 7500
1700 REM Bloque número 16.
1710 REM Reinicialización de los polinomios de trabajo y
1720 REM supresión de los decimales (si los hay) de los
1730 REM coeficientes del polinomio.
1740 IF NU=0 THEN 1780
1750 FOR I=0 TO N
1760 P#(I)=POLY#(I)*CDBL(10^NU)
1770 NEXT I
1780 FOR I=0 TO N
1790 N#(I)=P#(I)
1800 NEXT I
1810 REM Bloque número 17.
```

```
1820 REM Simplificar (si es posible) los coeficientes
1830 REM del polinomio.
1840 FOR I=0 TO N
1850 M#(I)=N#(I)
1860 NEXT I
1870 M=N
1880 GOSUB 8000
1890 GOSUB 8500
1900 FOR I=0 TO N
1910 N#(I)=M#(I)
1920 NEXT I
1930 REM Bloque número 18.
1940 REM Calcular y simplificar la derivada del polinomio.
1950 M=N-1
1960 FOR I=0 TO M
1970 M#(I)=CDBL(I+1)*N#(I+1)
1980 NEXT I
1990 GOSUB 8000
2000 GOSUB 8500
2010 REM Bloque número 19.
2020 REM Reducir la ecuación original a una con raíces
2030 REM simples.
2040 FOR I=0 TO M
2050 D#(I)=M#(I)
2060 NEXT I
2070 P=N
2080 GOSUB 9500
2090 FOR K=P-2 TO 0 STEP -1
2100 IF N#(K)<>0# THEN 2200
2110 NEXT K
2120 GOSUB 10000
2130 P=N
2140 GOSUB 9500
2150 FOR I=P-M TO 0 STEP -1
2160 N#(I)=C#(I)
2170 NEXT I
2180 N=P-M
2190 GOTO 2240
2200 IF P=2 THEN 2230
2210 GOSUB 10500
2220 GOTO 2080
2230 GOSUB 10000
2240 REM Bloque número 20.
2250 REM Cálculo de los polinomios de Sturm.
2260 FOR J=0 TO N
2270 B#(0,J)=N#(J)
2280 NEXT J
```

```
2290 L=1
2300 FOR J=N-1 TO 0 STEP -1
2310 B#(L,J)=N#(J+1)*CDBL(J+1)
2320 M#(J)=B#(L,J)
2330 NEXT J
2340 M=N-1
2350 GOSUB 8000
2360 GOSUB 8500
2370 FOR I=0 TO M
2380 D#(I)=M#(I)
2390 NEXT I
2400 P=N
2410 GOSUB 9500
2420 GOSUB 10500
2430 L=L+1
2440 FOR J=P-1 TO 0 STEP -1
2450 D#(J)=-D#(J)
2460 B#(L,J)=D#(J)
2470 NEXT J
2480 IF L=N THEN 2500
2490 GOTO 2410
2500 REM Bloque número 21.
2510 REM Cálculo de los subintervalos que separan
2520 REM las raíces de la ecuación.
2530 A#=CDBL(INF)
2540 B#=CDBL(SUP)
2550 LG#=B#-A#
2560 E#(0,0)=A#
2570 E#(0,N)=B#
2580 R=0
2590 GOSUB 11000
2600 IF C=N THEN 2660
2610 GOSUB 14500
2620 ETIQ=2
2630 GOSUB 5000
2640 GOSUB 5500
2650 GOTO 3900
2660 E#(1,0)=CDBL(C)
2670 A#=B#
2680 R=R+1
2690 GOSUB 11000
2700 E#(1,N)=CDBL(C)
2710 R=R+1
2720 I=0
2730 J=1
2740 K=N
2750 GOSUB 12000
```

```
2760 IF F=0 THEN 2830
2770 D=0
2780 FOR P=0 TO N-1
2790 V=CINT(E#(1,P)-E#(1,P+1))
2800 IF V=0 THEN 2840
2810 IF V=1 THEN 2880
2820 IF V>1 THEN 2980
2830 GOTO 3080
2840 IF D<>0 THEN 2860
2850 T=P
2860 D=D+1
2870 GOTO 2960
2880 IF D=0 THEN 2960
2890 GOSUB 12500
2900 I=T+1
2910 J=T+2
2920 K=N
2930 GOSUB 12000
2940 IF F=0 THEN 2770
2950 GOTO 2770
2960 NEXT P
2970 GOTO 3080
2980 IF D=0 THEN 3010
2990 GOSUB 12500
3000 P=T
3010 E#(1,P+V)=E#(1,P+1)
3020 E#(0,P+V)=E#(0,P+1)
3030 I=P
3040 J=P+1
3050 K=P+V
3060 GOSUB 12000
3070 GOTO 2770
3080 FOR J=0 TO N-1
3090 I#(J,0)=E#(0,J)
3100 I#(J,1)=E#(0,J+1)
3110 NEXT J
3120 TOL#=LG#/CDBL(10^6)
3130 FOR J=0 TO N-1
3140 A#=I#(J,0)
3150 GOSUB 13000
3160 B#=S#
3170 A#=I#(J,1)
3180 GOSUB 13000
3190 IF B#*S#>=0# THEN 3220
3200 NEXT J
3210 GOTO 3270
3220 GOSUB 14500
```



```
3230 ETIQ=2
3240 GOSUB 5000
3250 GOSUB 5500
3260 GOTO 3900
3270 MA#=I#(0,1)-I#(0,0)
3280 FOR I=0 TO N-1
3290 IF MA#>I#(I,1)-I#(I,0) THEN 3310
3300 MA#=I#(I,1)-I#(I,0)
3310 NEXT I
3320 FOR J=1 TO LI
3330 FOR I=0 TO N-1
3340 A#=I#(I,0)
3350 GOSUB 13000
3360 B#=S#
3370 A#=(I#(I,0)+I#(I,1))/2#
3380 GOSUB 13000
3390 IF S#=0# THEN 3430
3400 IF B#*S#<0# THEN 3460
3410 I#(I,0)=A#
3420 GOTO 3470
3430 I#(I,0)=A#-MA#/3#
3440 I#(I,1)=A#+MA#/3#
3450 GOTO 3470
3460 I#(I,1)=A#
3470 NEXT I
3480 GOSUB 13500
3490 IF MA#<TOL# THEN 3650
3500 FOR I=0 TO N-1
3510 IF I#(I,0)<>I#(I,1) THEN 3540
3520 I#(I,0)=I#(I,0)-MA#/3#
3530 I#(I,1)=I#(I,1)+MA#/3#
3540 NEXT I
3550 NEXT J
3560 REM Bloque número 22.
3570 REM Salida cuando no se encuentran los subintervalos
3580 REM en el número establecido de iteraciones.
3590 GOSUB 14000
3600 ETIQ=3
3610 CLS
3620 GOSUB 5000
3630 GOSUB 5500
3640 GOTO 3900
3650 REM Bloque número 23.
3660 REM Cálculo de las raíces de polinomio.
3670 IN=N
3680 MULTI=0
3690 FOR K=0 TO IN-1
```

```
3700 CLS
3710 MULT=0
3720 MAXA#=0#
3730 MAXB#=0#
3740 DES#=0#
3750 ERRMAX#=0#
3760 D2MAX#=0#
3770 DFHJ#=0#
3780 GOSUB 15000
3790 MULTI=MULTI+MULT
3800 NEXT K
3810 IF MULTI=GR THEN 3900
3820 CLS
3830 TI$(0)="El programa no ha podido determinar"
3840 TI$(1)="correctamente la multiplicidad de cada"
3850 TI$(2)="raíz de la ecuación"
3860 TI$(3)="Presione cualquier tecla para continuar"
3870 ETIQ=3
3880 GOSUB 5000
3890 GOSUB 5500
3900 REM Bloque número 24.
3910 REM Fin del programa.
3920 CLS
3930 TI$(0)="¿Desea ejecutar nuevamente el programa?"
3940 TI$(1)="si es así, pulse ``1'' para realizarlo, y,"
3950 TI$(2)="en caso contrario, cualquier otra tecla para"
3960 TI$(3)="finalizar la ejecución del programa"
3970 ETIQ=3
3980 GOSUB 5000
3990 GOSUB 5500
4000 IF Z$<>"1" THEN 4110
4010 CLS
4020 TI$(0)="¿Desea realizar los cálculos sobre el mismo"
4030 TI$(1)="polinomio? Si es así, pulse ``1'' para"
4040 TI$(2)="hacerlo, y, en caso contrario, cualquier otra"
4050 TI$(3)="tecla para cargar todos los datos nuevamente"
4060 ETIQ=3
4070 GOSUB 5000
4080 GOSUB 5500
4090 IF Z$="1" THEN 860 ELSE GOSUB 19000
4100 GOTO 210
4110 KEY ON
4120 CLS
4130 GOSUB 19500
4140 STOP
4500 REM Subrutinas del programa.
5000 REM Subrutina número 1.
```

```

5010 REM Impresión centrada de las etiquetas del
5020 REM programa, por página.
5030 FOR I=0 TO ETIQ
5040 FOR J=1 TO 3
5050 PRINT
5060 NEXT J
5070 PRINT TAB(INT((80-LEN(TI$(I)))/2));TI$(I)
5080 NEXT I
5090 RETURN
5500 REM Subrutina número 2.
5510 REM Paralización del avance del programa mientras
5520 REM el usuario lee los mensajes.
5530 Z$=""
5540 WHILE Z$=""
5550 Z$=INKEY$
5560 WEND
5570 RETURN
6000 REM Subrutina número 3.
6010 REM Verificar que el coeficiente de  $X^N$  es
6020 REM mayor que cero.
6030 IF Q#(0,N)>0# THEN 6070
6040 FOR I=0 TO N
6050 Q#(0,I)=-Q#(0,I)
6060 NEXT I
6070 RETURN
6500 REM Subrutina número 4.
6510 REM Examinar si cero es o no cota de las raíces
6520 REM del polinomio.
6530 FOR I=N-1 TO 0 STEP -1
6540 IF Q#(0,I)<0# THEN 6590
6550 NEXT I
6560 SUP=0
6570 CONTROL$="CERO"
6580 GOTO 6600
6590 P=I
6600 RETURN
7000 REM Subrutina número 5.
7010 REM Método del mayor de los módulos de los
7020 REM coeficientes negativos de la ecuación.
7030 IF CONTROL$="CERO" THEN 7120
7040 M=0
7050 FOR I=1 TO N
7060 IF Q#(0,M)<=Q#(0,I) THEN 7080
7070 M=I
7080 NEXT I
7090 G#=ABS(Q#(0,M))
7100 A#=Q#(0,N)

```

```

7110 SUPR=1+INT(1#+(G#/A#)^(1/(N-P)))
7120 RETURN
7500 REM Subrutina número 6.
7510 REM Método de Newton.
7520 FOR J=1 TO N+1
7530 Q#(J,N)=Q#(0,N)
7540 NEXT J
7550 FOR P=0 TO SUPR
7560 GRA=0
7570 FOR M=1 TO N
7580 FOR J=N-1 TO GRA STEP -1
7590 Q#(M,J)=Q#(M,J+1)*CDBL(P)+Q#(M-1,J)
7600 NEXT J
7610 GRA=GRA+1
7620 NEXT M
7630 J=0
7640 FOR M=1 TO N+1
7650 IF Q#(M,J)<=0# THEN 7690
7660 J=J+1
7670 NEXT M
7680 GOTO 7700
7690 NEXT P
7700 IF SW=1 THEN INF=-P ELSE SUP=P
7710 RETURN
8000 REM Subrutina número 7.
8010 REM Determinar el menor coeficiente (en valor
8020 REM absoluto) no nulo de un polinomio.
8030 FOR I=0 TO M
8040 IF ABS(M#(I))=0# THEN NEXT I
8050 ME#=ABS(M#(I))
8060 FOR I=0 TO M
8070 IF M#(I)=0# THEN 8090 ELSE IF ME#<=ABS(M#(I)) THEN 8090
8080 ME#=ABS(M#(I))
8090 NEXT I
8100 RETURN
8500 REM Subrutina número 8.
8510 REM Descomponer el menor coeficiente del polinomio
8520 REM en sus factores primos.
8530 B#=2#
8540 GOTO 8560
8550 GOSUB 9000
8560 IF ME#=1# THEN 8670
8570 IF ME#/B#=INT(ME#/B#) THEN 8550 ELSE B#=3#
8580 C#=INT(SQR(ME#)+1#)
8590 IF B#<C# THEN 8630
8600 B#=ME#
8610 GOSUB 9000

```

```

8620 GOTO 8670
8630 IF ME#/B#=INT(ME#/B#) THEN 8650 ELSE B#=B#+2#
8640 GOTO 8580
8650 GOSUB 9000
8660 GOTO 8580
8670 RETURN
9000 REM Subrutina número 9.
9010 REM Simplificar los coeficientes del polinomio,
9020 REM si es posible.
9030 ME#=ME#/B#
9040 FOR I=0 TO M
9050 IF M#(I)/B#<>INT(M#(I)/B#) THEN 9100
9060 IF I<>M THEN NEXT I
9070 FOR I=0 TO M
9080 M#(I)=M#(I)/B#
9090 NEXT I
9100 RETURN
9500 REM Subrutina número 10.
9510 REM División de polinomios con el algoritmo
9520 REM de los enteros.
9530 FOR I=0 TO M+1
9540 N#(I)=N#(I)*ABS(D#(M))
9550 NEXT I
9560 FOR I=P-M TO 0 STEP -1
9570 C#(I)=N#(I+M)/D#(M)
9580 FOR J=0 TO M
9590 N#(I+J)=N#(I+J)-C#(I)*D#(J)
9600 NEXT J
9610 IF I=0 THEN 9660
9620 FOR J=0 TO M
9630 N#(J)=N#(J)*ABS(D#(M))
9640 NEXT J
9650 NEXT I
9660 RETURN
10000 REM Subrutina número 11.
10010 REM Simplificar nuevamente el polinomio original
10020 REM y preparar su división por su M.C.D.
10030 FOR J=0 TO N
10040 N#(J)=P#(J)
10050 M#(J)=P#(J)
10060 NEXT J
10070 L=M
10080 M=N
10090 GOSUB 8000
10100 GOSUB 8500
10110 FOR J=0 TO N
10120 N#(J)=M#(J)

```

```

10130 NEXT J
10140 M=L
10150 RETURN
10500 REM Subrutina número 12.
10510 REM Preparar el cálculo del siguiente candidato a
10520 REM M.C.D. del polinomio original.
10530 FOR I=P-2 TO 0 STEP -1
10540 M#(I)=N#(I)
10550 NEXT I
10560 FOR J=P-2 TO 0 STEP -1
10570 IF M#(J)<>0# THEN 10590
10580 NEXT J
10590 M=J
10600 GOSUB 8000
10610 GOSUB 8500
10620 FOR I=P-1 TO 0 STEP -1
10630 N#(I)=D#(I)
10640 NEXT I
10650 FOR I=P-2 TO 0 STEP -1
10660 D#(I)=M#(I)
10670 NEXT I
10680 P=P-1
10690 RETURN
11000 REM Subrutina número 13.
11010 REM Cálculo del número de variaciones de la sucesión
11020 REM de Sturm en el valor A.
11030 I=0
11040 FOR J=N TO 1 STEP -1
11050 N#(J)=B#(I,J)
11060 FOR M=J-1 TO 0 STEP -1
11070 N#(M)=A#*N#(M+1)+B#(I,M)
11080 NEXT M
11090 M#(I)=N#(0)
11100 I=I+1
11110 NEXT J
11120 IF M#(0)<>0# THEN 11350
11130 IF R<>0 THEN 11240
11140 CLS
11150 TI$(0)="La cota inferior obtenida "+STR$(A#)
11160 TI$(1)="es una raíz de la ecuación; el programa"
11170 TI$(2)="no es capaz de calcular cotas para las"
11180 TI$(3)="raíces de la ecuación"
11190 TI$(4)="Presione cualquier tecla para continuar"
11200 ETIQ=4
11210 GOSUB 5000
11220 GOSUB 5500
11230 GOTO 4110

```

```

11240 IF R<>1 THEN 11350
11250 CLS
11260 TI$(0)="La cota superior obtenida "+STR$(B#)
11270 TI$(1)="es una raíz de la ecuación; el programa"
11280 TI$(2)="no es capaz de calcular cotas para las"
11290 TI$(3)="raíces de la ecuación"
11300 TI$(4)="Presione cualquier tecla para continuar"
11310 ETIQ=4
11320 GOSUB 5000
11330 GOSUB 5500
11340 GOTO 4110
11350 M#(N)=B#(N,0)
11360 C=0
11370 FOR J=0 TO N-1
11380 IF M#(J)=0# THEN 11460
11390 K=J+1
11400 IF M#(K)=0# THEN 11500
11410 IF M#(J)>0# THEN IF M#(K)>0# THEN 11440
11420 IF M#(J)<0# THEN IF M#(K)<0# THEN 11440
11430 C=C+1
11440 IF K=N THEN 11520
11450 GOTO 11470
11460 IF J=0 THEN 11480
11470 J=K-1
11480 NEXT J
11490 GOTO 11520
11500 K=K+1
11510 IF K<=N THEN 11400
11520 RETURN
12000 REM Subrutina número 14.
12010 REM Verificar que el número de variaciones sea el
12020 REM correcto en los intervalos, y, si es necesario,
12030 REM se reeligen los extremos del subintervalo.
12040 V=CINT(E#(1,I)-E#(1,K))
12050 F=0
12060 IF V=1 THEN 12340
12070 F=1
12080 L#=E#(0,K)-E#(0,I)
12090 LO#=L#/CDBL(V)
12100 E=J
12110 E#(0,E)=E#(0,E-1)+LO#
12120 A#=E#(0,E)
12130 GOSUB 11000
12140 E#(1,E)=CDBL(C)
12150 E=E+1
12160 IF E<=K-1 THEN 12110
12170 U=0

```

```

12180 A#=E#(0,U)
12190 GOSUB 13000
12200 IF S#<>0# THEN 12250
12210 E#(0,U)=E#(0,U)+(E#(0,U+1)-E#(0,U))/3#
12220 A#=E#(0,U)
12230 GOSUB 11000
12240 E#(1,U)=CDBL(C)
12250 U=U+1
12260 IF U<=N-1 THEN 12180
12270 A#=E#(0,N)
12280 GOSUB 13000
12290 IF S#<>0# THEN 12340
12300 E#(0,N)=E#(0,N)+1#/3#
12310 A#=E#(0,N)
12320 GOSUB 11000
12330 E#(1,N)=CDBL(C)
12340 RETURN
12500 REM Subrutina número 15.
12510 REM Substituye los extremos de los subintervalos
12520 REM que no poseen raíces, por otros que si las
12530 REM poseen.
12540 E#(1,T)=E#(1,P)
12550 E#(1,T+1)=E#(1,P+1)
12560 E#(0,T)=E#(0,P)
12570 E#(0,T+1)=E#(0,P+1)
12580 RETURN
13000 REM Subrutina número 16.
13010 REM Evaluación del polinomio reducido en el valor A
13020 REM para reducir la longitud de los subintervalos que
13030 REM separan a las raíces.
13040 N#(N)=B#(0,N)
13050 FOR M=N-1 TO 0 STEP -1
13060 N#(M)=A#*N#(M+1)+B#(0,M)
13070 NEXT M
13080 S#=N#(M+1)
13090 RETURN
13500 REM Subrutina número 17.
13510 REM Cálculo de la longitud del mayor de los
13520 REM subintervalos que separan las raíces.
13530 MA#=0#
13540 FOR I=0 TO N-1
13550 LO#=I#(I,1)-I#(I,0)
13560 IF MA#>=LO# THEN 13580
13570 MA#=LO#
13580 NEXT I
13590 RETURN
14000 REM Subrutina número 18.

```



```

14010 REM Mensajes de salida cuando no se encuentran
14020 REM los subintervalos.
14030 TI$(0)="Los subintervalos que acotan a las raíces"
14040 TI$(1)="simples de la ecuación, no se han podido"
14050 TI$(2)="calcular, en "+STR$(LI)+" iteraciones"
14060 TI$(3)="Presione cualquier tecla para continuar"
14070 RETURN
14500 REM Subrutina número 19.
14510 REM Mensajes de salida cuando la ecuación posee
14520 REM raíces complejas.
14530 TI$(0)="La ecuación posee raíces complejas, por lo"
14540 TI$(1)="que el método no le es aplicable"
14550 TI$(2)="Presione cualquier tecla para continuar"
14560 RETURN
15000 REM Subrutina número 20.
15010 REM Método de Newton-Raphson-Fourier.
15020 FOR I=0 TO N+1
15030 FOR J=0 TO N
15040 IF I=0 THEN Q#(I,J)=B#(0,J) ELSE Q#(I,J)=0#
15050 NEXT J
15060 NEXT I
15070 P#=I#(K,0)
15080 FOR J=1 TO 3
15090 Q#(J,N)=Q#(0,N)
15100 NEXT J
15110 GRA=0
15120 FOR M=1 TO 3
15130 FOR J=N-1 TO GRA STEP -1
15140 Q#(M,J)=Q#(M,J+1)*P#+Q#(M-1,J)
15150 NEXT J
15160 GRA=GRA+1
15170 NEXT M
15180 FA#=Q#(1,0)
15190 Q#(3,2)=Q#(3,2)*2#
15200 P#=I#(K,1)
15210 FOR J=1 TO 3
15220 Q#(J,N)=Q#(0,N)
15230 NEXT J
15240 GRA=0
15250 FOR M=1 TO 3
15260 FOR J=N-1 TO GRA STEP -1
15270 Q#(M,J)=Q#(M,J+1)*P#+Q#(M-1,J)
15280 NEXT J
15290 GRA=GRA+1
15300 NEXT M
15310 FB#=Q#(1,0)
15320 Q#(3,2)=Q#(3,2)*2#

```

```

15330 IF FA#*FB#<=0# THEN 15400
15340 GOSUB 16500
15350 CLS
15360 GOSUB 5000
15370 GOSUB 5500
15380 CLS
15390 GOTO 15770
15400 IF Q#(1,0)*Q#(3,2)>0# THEN 15440
15410 H0#=I#(K,1)
15420 DES#=I#(K,0)
15430 GOTO 15460
15440 H0#=I#(K,0)
15450 DES#=I#(K,1)
15460 GOSUB 16000
15470 HI#=H0#
15480 FOR IT=1 TO LI
15490 FOR I=0 TO N
15500 Q#(0,I)=B#(0,I)
15510 NEXT I
15520 FOR J=1 TO 2
15530 Q#(J,N)=Q#(0,N)
15540 NEXT J
15550 GRA=0
15560 FOR M=1 TO 2
15570 FOR J=N-1 TO GRA STEP -1
15580 Q#(M,J)=Q#(M,J+1)*HI#+Q#(M-1,J)
15590 NEXT J
15600 GRA=GRA+1
15610 NEXT M
15620 FHI#=Q#(1,0)
15630 DFHI#=Q#(2,1)
15640 HJ#=HI#-FHI#/DFHI#
15650 ER#=ABS((HJ#-HI#)/HJ#)
15660 IF ER#<PO# THEN 15740
15670 HI#=HJ#
15680 NEXT IT
15690 GOSUB 17000
15700 CLS
15710 GOSUB 5000
15720 GOSUB 5500
15730 GOTO 15770
15740 R#=HJ#
15750 CLS
15760 GOSUB 17500
15770 RETURN
16000 REM Subrutina número 21.
16010 REM Cálculo del máximo de la segunda derivada de la

```

```

16020 REM la función en los extremos del intervalo.
16030 MAXA#=ABS(Q#(3,2))
16040 P#=I#(K,1)
16050 FOR J=1 TO 3
16060 Q#(J,N)=Q#(0,N)
16070 NEXT J
16080 GRA=0
16090 FOR M=1 TO 3
16100 FOR J=N-1 TO GRA STEP -1
16110 Q#(M,J)=Q#(M,J+1)*P#+Q#(M-1,J)
16120 NEXT J
16130 GRA=GRA+1
16140 NEXT M
16150 Q#(3,2)=Q#(3,2)*2#
16160 MAXB#=ABS(Q#(3,2))
16170 IF MAXA#>MAXB# THEN D2MAX#=MAXA# ELSE D2MAX#=MAXB#
16180 RETURN
16500 REM Subrutina número 22.
16510 REM Mensaje de salida cuando no se encuentra la raíz.
16520 REM dentro del subintervalo obtenido.
16530 TI$(0)="La"+STR$(K)+"-ésima raíz de la ecuación"
16540 TI$(1)="no se pudo calcular, ya que el subintervalo"
16550 TI$(2)="que la acota no cumple las condiciones de"
16560 TI$(3)="bisección; se procederá a calcular las demás"
16570 TI$(4)="Presione cualquier tecla para continuar"
16580 ETIQ=4
16590 RETURN
17000 REM Subrutina número 23.
17010 REM Mensaje de salida cuando no se encuentra la raíz
17020 REM en el número indicado de iteraciones.
17030 TI$(0)="La"+STR$(K+1)+"-ésima raíz de la ecuación"
17040 TI$(1)="no pudo calcularse en"+STR$(LI)+" iteraciones"
17050 TI$(2)="se procederá a calcular las restantes"
17060 TI$(3)="Presione cualquier tecla para continuar"
17070 ETIQ=3
17080 RETURN
17500 REM Subrutina número 24.
17510 REM Cálculo de la multiplicidad de la raíz localizada.
17520 FOR I=0 TO GR
17530 Q#(0,I)=POLY#(I)
17540 NEXT I
17550 FOR J=1 TO GR+1
17560 Q#(J,GR)=Q#(0,GR)
17570 NEXT J
17580 GRA=0
17590 FOR M=1 TO GR
17600 FOR J=GR-1 TO GRA STEP -1

```

```

17610 Q#(M,J)=Q#(M,J+1)*R#+Q#(M-1,J)
17620 NEXT J
17630 GRA=GRA+1
17640 NEXT M
17650 FAC=1
17660 FOR P=2 TO GR+1
17670 FAC=FAC*P
17680 Q#(P,P-1)=Q#(P,P-1)*CDBL(FAC)
17690 NEXT P
17700 MU=1
17710 FOR M=2 TO GR+1
17720 IF ABS(Q#(M,MU))>10#*PO# THEN 17770
17730 MU=MU+1
17740 NEXT M
17750 MULT=GR
17760 GOTO 17780
17770 MULT=MU
17780 CLS
17790 GOSUB 18000
17800 GOSUB 18500
17810 GOSUB 5000
17820 GOSUB 5500
17830 RETURN
18000 REM Subrutina número 25.
18010 REM Cálculo de la cota del error cometido en la
18020 REM última iteración del método.
18030 P#=HJ#
18040 FOR J=1 TO 2
18050 Q#(J,N)=Q#(0,N)
18060 NEXT J
18070 GRA=0
18080 FOR M=1 TO 2
18090 FOR J=N-1 TO GRA STEP -1
18100 Q#(M,J)=Q#(M,J+1)*P#+Q#(M-1,J)
18110 NEXT J
18120 GRA=GRA+1
18130 NEXT M
18140 DFHJ#=Q#(2,1)
18150 ERRMAX#=ABS(D2MAX#*(DES#-HJ#)^2/(2*DFHJ#))
18160 RETURN
18500 REM Subrutina número 26.
18510 REM Mensajes de salida cuando se encuentra la
18520 REM raíz y su multiplicidad.
18530 TI$(0)="La"+STR$(K+1)+"-ésima raíz de la ecuación es"
18540 TI$(1)=STR$(R#)+", una cota del error cometido en la"
18550 TI$(2)="aproximación es:"+STR$(ERRMAX#)+", "
18560 TI$(3)="posee una multiplicidad igual a:"+STR$(MULT)

```

```
18570 TI$(4)="y se calculó en"+STR$(IT)+" iteraciones"
18580 TI$(5)="Presione cualquier tecla para continuar"
18590 ETIQ=5
18600 RETURN
19000 REM Subrutina número 27.
19010 REM Inicialización de las variables del programa
19020 REM para otra ejecución.
19030 ERASE B#,C#,D#,E#,I#,M#,N#,P#,POLY#,Q#
19040 RETURN
19500 REM Subrutina número 28.
19510 REM Fin del programa.
19520 END
```

A continuación, presentamos una tabla que contiene un grupo de ecuaciones polinomiales de raíces reales; éstas han sido aproximadas empleando el programa que acabamos de mostrar.

TABLA 14

ECUACIÓN	RAÍCES VERDADERAS	MULTIPLICIDAD VERDADERA	RAÍCES OBTENIDAS	MULTIPLICIDAD OBTENIDA	COTA DEL ERROR COMETIDO EN LA ÚLTIMA ITERACIÓN	ITERACIONES EMPLEADAS
1	-2.3300587	1	- 2.330058739567982	1	6.026246206847896E-014	3
	0.20163968	1	0.2016396757234047	1	7.188620323501215E-013	3
	2.1284191	1	2.128419063844577	1	5.497576391633488E-012	3
2	-1.6920215	1	- 1.692021471630096	1	4.496188772395555E-013	3
	- 1.3568959	1	- 1.356895867892209	1	4.889169509606158E-013	3
	3.0489173	1	3.048917339522305	1	2.39411277094094E-012	3
3	-5.0000000	1	- 5	1	1.101498849693314E-012	3
	- 4.0000000	1	- 4	1	1.567530027164235E-011	3
	4.0000000	1	4	1	2.127680968995136E-012	3
	5.0000000	1	5	1	1.101499717055094E-012	3
4	0.33333333	1	0.3333333333333333	1	2.011380448445825E-013	3
	1.0000000	1	1	1	1.79056000983491E-012	3
	3.0000000	1	3	1	1.304557451098407E-011	3
	9.0000000	1	9	1	6.299381232498747E-012	3
5	1.0000000	1	1	1	1.508535196299097E-013	3
	3.0000000	1	3	1	1.061075128718739E-012	3
	9.0000000	1	9	1	7.152251551353829E-011	3
	27.0000000	1	27	1	4.080036551834702E-012	3
6	-2.0000000	2	-2	2	7.231913484693509E-013	3
	1.0000000	1	1	1	6.715375907332756E-013	3
	3.0000000	1	3	1	1.578170607672866E-022	2
	4.0000000	2	4	2	6.295673559948161E-014	3
ECUACIÓN #1 $x^3 - 5x + 1$	ECUACIÓN #2 $x^3 - 7x - 7$	ECUACIÓN #3 $x^4 - 41x^2 + 400$	ECUACIÓN #4 $3x^4 - 40x^3 + 130x^2 - 120x + 27$	ECUACIÓN #5 $x^4 - 40x^3 + 390x^2 - 1080x + 729$	ECUACIÓN #6 $x^6 - 8x^5 + 7x^4 + 68x^3 - 100x^2 - 160x + 192$	

Al comparar las tablas **12** y **14**, es evidente que se obtuvieron resultados iguales. Esto ocurre así, porque, similarmente a lo hecho con el método de bisección de Bolzano, la sección de separación de raíces del programa del método mixto **2** de aproximación fue programada para que, por defecto, use como porcentaje de la longitud máxima de los subintervalos que acotarán a las raíces de la ecuación, el valor $\varepsilon = 0.0001 = 10^{-4}$, que fue el empleado para calcular la tabla **12**, de modo que los intervalos que generó internamente el método son los mismos que se emplearon como valores iniciales para calcular la tabla **12**.

De este modo, resulta que ambos programas generan respuestas similares a las obtenidas por el programa del método de Newton-Raphson-Fourier. Sin embargo, con la última técnica no es necesario suministrar los subintervalos que acotan a las raíces, de modo que dicha técnica resulta ser más eficiente en general.

3.7. VENTAJAS, DESVENTAJAS Y LIMITACIONES DE LOS MÉTODOS DESARROLLADOS, EN LA SOLUCIÓN DE PROBLEMAS PARTICULARES.

Recapitulando lo realizado hasta el momento, en este trabajo hemos desarrollado dos programas de computadoras que implementan los dos métodos desarrollados dentro de nuestra exposición, que consideramos más aptos para solucionar nuestro problema, en aplicaciones prácticas que pueden ser usadas para resolver, dentro de sus limitaciones, ecuaciones polinomiales con raíces reales. A continuación, procederemos a dejar claramente establecidas cuales son las limitaciones, desventajas y ventajas que caracterizan a dichos programas.

En primer término, con respecto a las limitaciones de los programas, tenemos que recalcar que las limitaciones de cada uno de los métodos que se han desarrollado en forma individual se van sumando con las de los demás al integrar los programas en uno solo.

De este modo, en primer término, todos los métodos tienen su acción limitada a ecuaciones que no lleven a la máquina a realizar operaciones con números que excedan la capacidad aritmética del compilador empleado para generar el programa. El compilador que empleamos en la creación de nuestros programas tiene la capacidad de operar con números que sean menores, en valor absoluto, que 10^{10} , empleando dentro de sus cálculos un máximo de 16 cifras significativas, que el usuario puede manejar dentro de las entradas o salidas de información —normalmente para efectos de cálculo, el compilador maneja algunas cifras significativas adicionales—. Para efectos prácticos, estos son unos límites bastante amplios, pero, de todos modos, la acción de los programas, bajo ningún concepto, puede revasarlos.

Por otro lado, ambos métodos tienen su acción limitada —con convergencia garantizada—, exclusivamente, sobre ecuaciones polinomiales de raíces reales que, al menos, posean dos raíces simples —un caso relativamente común—, en caso contrario, el programa abortará, ya que tratará de realizar operaciones no permitidas dentro del lenguaje **BASIC**. Sin embargo, esta situación no es excesivamente problemática, pero debe tomarse siempre en cuenta.

Además, si las raíces de las ecuaciones son excesivamente próximas (dentro de las limitaciones impuestas por el compilador que estamos usando, dos raíces se pueden considerar como muy próximas entre sí, si su diferencia en valor absoluto, es menor que 10^{-7}), el error por redondeo afectará

definitivamente el proceso de separación de raíces, al reducir la ecuación a una que no tenga raíces múltiples; por lo que el método podría llegar a resultados totalmente errados. Esta situación es, definitivamente, una falla del método, que sólo se puede salvar, parcialmente, con el empleo de un compilador más poderoso.

Finalmente, existe la posibilidad de que los métodos desarrollados obtengan la multiplicidad de las raíces erróneamente y que no puedan determinar la falla, debido a que los programas verifican únicamente que la suma de las multiplicidades de las raíces obtenidas coincida con el grado del polinomio original. Esto puede ocurrir cuando una de las multiplicidades difiere —en un factor $k \in \mathbb{Z}$ — de la multiplicidad verdadera; y en las multiplicidades de otras raíces obtenidas se compensa dicha diferencia. Esta deficiencia ocurre por efecto de realizar el cálculo de las multiplicidades de las raíces obtenidas, en forma aproximada.

Con respecto a las desventajas de los programas, es evidente que, por su construcción, no son capaces de proporcionar, en general, respuestas exactas, de modo que toda la información que ellos calculan, debe ser sujeta a un análisis cuidadoso por parte del usuario, con el objeto de verificar y —si es posible— rectificar la respuesta. Así, es falsa la opinión generalizada de que las computadoras siempre generan respuestas correctas y exactas.

Por otro lado, los programas que hemos desarrollado tienen la ventaja de brindarnos la posibilidad de poder obtener las raíces reales de ecuaciones que, de otro modo, no habría forma de determinar, de manera que, indiscutiblemente, estos programas presentan una solución práctica, aunque no absoluta, del problema que originalmente nos planteamos, es decir, la solución de ecuaciones algebraicas reales que posean ceros únicamente reales.

3.8. COMENTARIOS FINALES.

De todo lo tratado hasta el momento, no debe deducirse la inexistencia de métodos iterativos relativamente eficientes para calcular las raíces complejas de ecuaciones algebraicas. Simplemente, ellos no forman parte del contenido principal de este trabajo.

Sin embargo, éstos son tan importantes como los métodos desarrollados para obtener raíces reales, de modo que finalizaremos nuestro trabajo realizando una breve referencia general de los métodos más conocidos que se pueden emplear para la obtención de raíces complejas, sin adentraremos en sus detalles, mencionando algunos por sus nombres propios, y dando algunas ideas acerca de cómo se aplican, así como algunas referencias bibliográficas.

Este proceder lo justificamos en función de que muchos de los teoremas que sustentan estos métodos sólo pueden probarse empleando, ampliamente, la teoría de funciones de variable compleja avanzada. Pero, nuestro interés en citarlos se fundamenta en que: quienes lean este trabajo, puedan conocer de su existencia, de modo que, en el futuro, el tema sea desarrollado y se profundice; tal vez enmarcado en otro trabajo de graduación.

Los métodos a los que nos referimos no son entorpecidos tanto por la presencia de raíces múltiples o complejas; sin embargo, son, por lo general, de convergencia lenta.

Iniciamos nuestra exposición, estableciendo que: salvo contadas excepciones, todos los métodos desarrollados de acotación, separación y aproximación de raíces, poseen algún equivalente que permite realizar una tarea similar, en el problema de encontrar las raíces complejas de un polinomio.

Así, podemos hacer referencia a la fórmula de Montel, que establece que: todas raíces, reales y complejas, de un polinomio de coeficientes complejos:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

donde $a_n \neq 0$ y los $a_i \in \mathbb{C}$, con $i = 0, 1, \dots, n$, están ubicadas dentro de la bola cerrada de centro en el origen, y radio:

$$r = 2 \max(|b_0|, |b_1|, \dots, |b_n|)$$

donde:

$$b_k = \sqrt[k]{b_k}, \quad k = 0, 1, \dots, n$$

Los interesados en la demostración de este resultado pueden hallarla en **SAGASTUME BERRA y FERNÁNDEZ [29]**, página 452 y siguientes.

Con respecto al problema de la separación de los ceros complejos, tenemos, dentro de la teoría de funciones de variable compleja, el teorema del argumento, cuya aplicación permite determinar el número de ceros que se encuentran dentro de una curva cerrada simple —por ejemplo, una circunferencia—. Además, el teorema de Rouché, permite establecer cuándo dos funciones poseen el mismo número de ceros, dentro de una curva cerrada simple, y en la que dichas funciones cumplen ciertas condiciones.

Estos dos teoremas, al ser aplicados en forma combinada, permiten establecer el número de ceros de una función que están ubicados dentro de un anillo preestablecido, de modo que, en teoría, dejan entreabierto la posibilidad

de desarrollar un algoritmo de separación de ceros complejos. Los detalles de estos teoremas se pueden encontrar en **SPIEGEL [31]**, página 119 y siguientes.

Con respecto al problema de aproximar raíces complejas, tenemos que existe una gran diversidad de métodos que pueden ser empleados. Unos, no son más que la extensión de los métodos ya desarrollados en el problema de encontrar raíces reales, al caso complejo (en su mayoría, no tienen una deducción muy complicada, más que todo, es tediosa); en cambio, existen métodos que sólo su formulación geométrica, se realiza en niveles realmente avanzados de la teoría de funciones de variable compleja.

Así tenemos que existe una versión del método de aproximaciones sucesivas, especialmente desarrollada para el cálculo de raíces complejas de funciones en general. El método en mención se conoce como método de Lin, y sus detalles pueden encontrarse en **HILDELBAND [13]**, página 454 y siguientes.

El ya renombrado método de Newton-Raphson posee también sus versiones, especialmente adaptadas para el cálculo de raíces complejas.

Una de éstas consiste en plantear el método empleando solamente aritmética compleja (y, lógicamente, empleando valores iniciales complejos). Los pormenores de este método, se pueden encontrar en **RALSTON [26]**, página 413 y siguientes.

Otro de estos procedimientos —basados en el método de Newton-Raphson— consiste en “esconder” las raíces complejas dentro de factores cuadráticos, y realizar el proceso iterativo de modo que se aproxime, en vez de una raíz compleja —que se puede considerar un factor lineal—, dos raíces complejas —que se encuentran encerradas dentro de un factor

cuadrático—. Este método presenta la ventaja de que no se requiere aritmética compleja, salvo en el momento en que se separan las raíces de cada factor cuadrático aproximado.

El método en mención posee una abundante bibliografía y, entre otros, se puede documentar en la ya citada obra de **RALSTON [26]**, página 414 y siguientes; así como en **HENRICI [12]**, página 124 y siguientes; y en **HAMMING [11]**, página 108 y siguientes.

Además, el método de Laguerre también puede usarse para aproximar raíces complejas, aunque todavía no se conoce en forma precisa las cualidades de su convergencia para esa situación.

También tenemos el método de la raíz cuadrada de Dandelin-Graeffe que consiste, básicamente, en construir una sucesión de ecuaciones, donde las raíces de una son los cuadrados de las raíces de la anterior, permitiendo este procedimiento, que las raíces se espacien en sus magnitudes, de modo que, llegado cierto momento, éstas puedan aproximarse en base a un cálculo, fundamentado en las fórmulas de Viète.

Este método se puede documentar en **SAGASTUME BERRA y FERNÁNDEZ [29]**, página 507 y siguientes; **HILDELBRAND [13]**, página 462 y siguientes; y en **RALSTON [26]**, página 397 y siguientes.

Y, finalmente, tenemos los métodos de Bernoulli y el Algoritmo QD (este último es una generalización del método de Bernoulli, y es conocido también como algoritmo de cociente-diferencia, o algoritmo de los rombos) que, al ser aplicados en forma conjunta, permiten resolver, en forma casi completa —en teoría—, el problema de aproximar las raíces complejas de un polinomio complejo.

Debemos resaltar que las pruebas de convergencia de todos estos métodos —en especial los que se han tratado de último—, así como su deducción, se ubican en las partes más avanzadas de la teoría de funciones de variable compleja —principalmente en el caso del Algoritmo QD o de cociente-diferencia—.

Sin embargo, dado que estos métodos resuelven, aunque sea en teoría, el problema, se presentan revestidos de gran importancia, por lo que no pueden pasar desapercibidos.

Así, puede encontrarse bastante información acerca de ellos sin adentrarse demasiado en los detalles de sus pruebas matemáticas, en las obras de: **HENRICI [12]**, capítulos 7 y 8; **STIEFEL [32]**, página 104 y siguientes; y **SCHEID [30]**, capítulo 25; así como en las obras de **HILDELBRAND [13]**, **RALSTON [26]**, y **HAMMING [11]**.

CONCLUSIONES

Finalizado nuestro trabajo de investigación, hemos llegado a las siguientes conclusiones:

1. La obtención de las raíces reales de ecuaciones algebraicas con ceros reales es un problema que, en general, no puede resolverse en forma exacta —en la práctica—, con los medios con que actualmente cuenta la matemática de modo que dicha solución sea de carácter práctico para cualquier usuario.
2. Las únicas fórmulas exactas para obtener las raíces de ecuaciones algebraicas, que están planteadas de modo que pueden ser aplicadas en la práctica por casi cualquier persona, y generar resultados útiles, son las fórmulas de las ecuaciones lineales y cuadráticas.
3. La aplicación práctica de las fórmulas para resolver ecuaciones cúbicas y cuárticas se ve anulada por la necesidad de tener que considerar, en última instancia, sistemas de ecuaciones no lineales, además que una larga secuencia de substituciones, que imposibilitan su empleo a personas que no posean un cierto dominio matemático, además de involucrar casos irreducibles.
4. Si prescindimos del requisito de la exactitud en la solución, los métodos iterativos se presentan como una alternativa viable, y que puede ser aplicada, en general, a cualquier tipo de ecuación, siempre y cuando se cuente con un medio para realizar la gran cantidad de operaciones aritméticas que involucran.
5. Los métodos de Laguerre-Thibault y del mayor de los módulos de los coeficientes negativos de la ecuación presentan una solución práctica

para el problema de obtener cotas superiores para las raíces de las ecuaciones.

6. El método de bisección de Bolzano, aunado con el de Sturm, presentan una solución exacta, pero extremadamente laboriosa, para el problema de separar las raíces de ecuaciones algebraicas.
7. Los métodos de bisección y de Newton-Raphson-Fourier presentan una solución relativamente sencilla para el problema de aproximar las raíces de las ecuaciones, asumiendo que éstas han sido previamente separadas y se obtuvieron subintervalos suficientemente pequeños.
8. El empleo de computadoras en la solución de ecuaciones de raíces reales puede representar un ahorro significativo de tiempo y esfuerzo en la solución de nuestro problema, siempre y cuando sea desarrollado previamente un programa adecuado.
9. El empleo de computadoras añade ciertas dificultades adicionales a la solución del problema —surgen dificultades de error por redondeo, entre otras—, que deben ser tomados en cuenta, si se desea desarrollar una aplicación de uso general.
10. La investigación de nuevos métodos, así como los avances tecnológicos en materia de equipo computacional, hacen que este sea un campo de investigación con un gran potencial futuro.

RECOMENDACIONES

Al culminar nuestro trabajo de investigación, nos sentimos en capacidad de realizar, para quienes les interese documentarse en el futuro acerca del problema tratado aquí, la siguiente lista de recomendaciones:

1. Consideramos que el estudio de las técnicas iterativas, en general, debe ser incentivado y ampliado a nivel de la licenciatura, ya que su aplicación es muy útil en la solución de problemas (tales como la resolución de ecuaciones diferenciales lineales y el cálculo de valores característicos, en los casos que las ecuaciones resultantes no poseen una factorización evidente) que, por otras vías, se presentan sumamente difíciles.
2. Recomendamos que el estudio de la teoría de ecuaciones clásica sea incorporando a los planes de estudio vigentes —principalmente a nivel secundario de bachillerato y universitario—, aunque sea en forma somera, ya que su aplicación conjunta con los más recientes métodos iterativos, dentro de un programa de computadoras, en general, aumenta la potencia de dichos métodos, haciéndolos más efectivos y seguros —por ejemplo, en el caso del método mixto de acotación—.
3. Dado que el problema aquí tratado no tiene una solución que, en general, sea exacta —en la práctica—, la experiencia y el buen juicio del investigador son esenciales para obtener resultados significativos en la investigación de estos temas.
4. Recomendamos a los que incursionen en estos temas de manera inicial estar preparados para enfrentarse al fracaso práctico de métodos que, en teoría, pueden parecer como la solución ideal del problema (ejemplo, la aplicación práctica del método de Sturm, al que tuvimos que añadir ciertos procedimientos no contemplados en la concepción teórica

del método, y que, sin ellos, obtendríamos resultados incorrectos), al someterlos a prueba —principalmente en los casos extremos—. En ese sentido, recomendamos que los investigadores y docentes futuros estén abiertos a emplear —y enseñar— cualquier resultado de la matemática —no importa que tan distantes estén los campos de estudio—, que presente una mejoría del planteamiento del problema.

11. También es recomendable que se aprovechen todas las facilidades que nos vayan brindando los avances tecnológicos, en la investigación de este tema, con el objeto de facilitar la labor investigativa.

ANEXOS

ANEXO #1: APÉNDICE DE TEORÍA DE ECUACIONES

1. TEORÍA DE TRANSFORMACIONES.

Sea $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$, una ecuación algebraica de grado n . Ésta puede ser transformada, mediante la traslación lineal $x = y + k$, en otra ecuación $F(y) = 0$, también de grado n , donde el coeficiente de y^{n-1} es igual a cero, eligiendo un k conveniente.

Para escoger el k adecuado, se desarrolla a $f(x)$ empleando la fórmula de Taylor (ver apéndice de cálculo, teorema número 1):

$$f(x+k) = \sum_{i=0}^n \frac{f^{(i)}(k)}{i!} x^i$$

e igualamos a cero el coeficiente de la potencia $(n-1)$ resultando:

$$k = -\frac{a_{n-1}}{na_n}$$

SAGASTUME BERRA y FERNÁNDEZ [29], página 429 y siguientes

2. RELACIONES ENTRE LAS RAÍCES DE UNA ECUACIÓN ALGEBRAICA Y SUS COEFICIENTES (FÓRMULAS DE VIETE).

Si x_1, x_2, \dots, x_n son todas las raíces de la ecuación:
 $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$, entonces:

$$\begin{aligned}
\sum_{i=1}^n x_i &= -\frac{a_{n-1}}{a_n} \\
\sum_{\substack{i,j=1 \\ i < j}}^n x_i \cdot x_j &= \frac{a_{n-2}}{a_n} \\
\sum_{\substack{i,j,k=1 \\ i < j < k}}^n x_i \cdot x_j \cdot x_k &= \frac{a_{n-3}}{a_n} \\
&\vdots \\
&\vdots \\
&\vdots \\
\sum_{\substack{i,j,k,\dots,z=1 \\ i < j < k < \dots < z}}^n x_i \cdot x_j \cdot x_k \cdot \dots \cdot x_z &= \frac{(-1)^n a_0}{a_n}
\end{aligned}$$

HALL y KNIGHT [10], página 544 y siguientes

3. TEOREMA.

Dada una ecuación algebraica cualquiera

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

(de coeficientes a_i aun complejos), con $a_n = 1$ (es decir, el polinomio f es mónico), es posible formar, con operaciones racionales, otra ecuación que tenga por raíces todas y sólo las de la ecuación dada, pero todas simples. Es la ecuación $g(x) = 0$, donde el polinomio g está dado por:

$$g(x) = \frac{f(x)}{\left[f, \frac{1}{n} f' \right]}$$

donde:

$$\left[f, \frac{1}{n} f' \right]$$

representa el máximo común divisor de los polinomios f y $\frac{1}{n} f'$.

SAGASTUME BERRA y FERNÁNDEZ [29], página 402 y siguientes

4. TEOREMA.

Para que un valor a sea raíz de multiplicidad r de un polinomio $f(x)$, es necesario y suficiente que se anulen, en a , el polinomio y sus derivadas hasta el orden $(r-1)$, inclusive, y no la de orden r , es decir:

$$f(a) = f'(a) = f''(a) = \dots = f^{r-1}(a) = 0; \quad f^r(a) \neq 0$$

SAGASTUME BERRA y FERNÁNDEZ [29], página 387 y siguientes

5. TEOREMA.

Para que a sea raíz, al menos r -ple de f , es necesario y suficiente que:

$$f(a) = f'(a) = f''(a) = \dots = f^{r-1}(a) = 0$$

SAGASTUME BERRA y FERNÁNDEZ [29], página 387 y siguientes

6. TEOREMA.

Las raíces múltiples del polinomio f están caracterizadas por ser raíces comunes a f y f' .

SAGASTUME BERRA y FERNÁNDEZ [29], página 387 y siguientes

7. TEOREMA.

Si a es raíz r -ple de f , es raíz $(r-1)$ -ple de f' , raíz $(r-2)$ -ple de f'' , ..., raíz $(r-k)$ -ple de f^k , para $k \leq r$.

SAGASTUME BERRA y FERNÁNDEZ [29], página 387 y siguientes

8. TEOREMA.

Todo polinomio de la forma: $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$ con $a_n \neq 0$, que posea todos los $a_i \geq 0$, para $i = 0, 1, 2, \dots, n$, (no todos los a_i serán simultáneamente nulos), tendrá todas sus raíces negativas y, por tanto, cero es una cota superior de éstas.

SAGASTUME BERRA y FERNÁNDEZ [29], página 448

9. TEOREMA (REGLA DE RUFFINI O DIVISIÓN SINTÉTICA).

Si $P = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$, es un polinomio de grado n y r es una raíz de P , entonces:

$$P = (x - r)[b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_2x + b_1] + b_0$$

con $P(r) = b_0$ y donde:

$$b_i = \begin{cases} a_i + rb_{i+1} & \text{si } i = 0, 1, \dots, n-1 \\ a_i & \text{si } i = n \end{cases}$$

Para fines de cálculo, podemos arreglar el trabajo de acuerdo al siguiente esquema:

a_n	a_{n-1}	a_{n-2}	a_{n-3}	\dots	a_1	a_0	r
	rb_n	rb_{n-1}	rb_{n-2}	\dots	rb_2	rb_1	
b_n	b_{n-1}	b_{n-2}	b_{n-3}	\dots	b_1	b_0	$= P(r)$

HAASER, LASALLE y SULLIVAN [8], página 508 y siguientes

10. TEOREMA (ESQUEMA DE HORNER GENERALIZADO).

Sea $P = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$, un polinomio de grado n . En su desarrollo por la fórmula de Taylor alrededor del punto $x = a$, el coeficiente $\frac{P_n^{(m)}(a)}{m!}$ de la potencia $(x - a)^m$, en dicha fórmula, es el resto que se obtiene de dividir $(m+1)$ veces a $P_n(x)$ por $x - a$ (con m variando a través de $0, 1, \dots, n-1$), reiterando la división sobre el cociente resultante, y tomando el coeficiente de $(x - a)^n$, como la constante que resulta al finalizar el algoritmo.

Para fines de cálculo, podemos arreglar el trabajo de acuerdo al siguiente esquema, fundamentado en la regla de Ruffini:

a_n	a_{n-1}	a_{n-2}	a_{n-3}	\dots	a_1	a_0	$r = a$
	rb_n	rb_{n-1}	rb_{n-2}	\dots	rb_2	rb_1	
b_n	b_{n-1}	b_{n-2}	b_{n-3}	\dots	b_1	$b_0 = P_n(a)/0!$	
b_n	b_{n-1}	b_{n-2}	b_{n-3}	\dots	b_1	$r = b_0$	
	rc_n	rc_{n-1}	rc_{n-2}	\dots	rc_2		
c_n	c_{n-1}	c_{n-2}	c_{n-3}	\dots	c_1	$= P'_n(a)/1!$	
c_n	c_{n-1}	c_{n-2}	c_{n-3}	\dots	c_2	$r = c_1$	
	rd_n	rd_{n-1}	rd_{n-2}	\dots	rd_3		
d_n	d_{n-1}	d_{n-2}	d_{n-3}	\dots	d_2	$= P''_n(a)/2!$	
\cdot	\cdot	\cdot					
\cdot	\cdot	\cdot					
\cdot	\cdot	\cdot					

$$z_n = P_n^{(m)}(a)/m!$$

SAGASTUME BERRA y FERNÁNDEZ [29], página 373

11. TEOREMA.

Si a es una raíz de la ecuación $f(x)=0$ y ε un número positivo suficientemente pequeño, $f(a-\varepsilon)$ y $f'(a-\varepsilon)$ tienen signos opuestos; en cambio, $f(a+\varepsilon)$ y $f'(a+\varepsilon)$ tienen igual signo.

SAGASTUME BERRA y FERNÁNDEZ [29], página 467

12. TEOREMA (COROLARIO NÚMERO 1 DEL TEOREMA DE BOLZANO).

Si un polinomio real y continuo $f(x)$, no se anula en el intervalo (a,b) , el signo de $f(x)$ no cambia en dicho intervalo.

SAGASTUME BERRA y FERNÁNDEZ [29], página 465

13. TEOREMA (COROLARIO NÚMERO 2 DEL TEOREMA DE BOLZANO).

Sea dada la función $f(x)$, continua en el intervalo (a,b) . Si entre a y b $f(x)$ no tiene raíces o hay un número par de éstas, $f(a)$ y $f(b)$ tienen igual signo; en cambio, si entre a y b hay un número impar de raíces, $f(a)$ y $f(b)$ tienen signos opuestos; y recíprocamente, cada raíz múltiple debe contarse tantas veces como indique su orden de multiplicidad.

SAGASTUME BERRA y FERNÁNDEZ [29], página 466

14. TEORÍA DE POLINOMIOS ASOCIADOS.

Sea dado el polinomio $P(x) = 0$. Cualquier otro polinomio $P'(x) = 0$, que posea los mismos divisores (múltiplos o raíces) de $P(x)$, se denomina "polinomio asociado" de $P(x)$.

Se verifica que si $P(x)$ y $P'(x)$ son dos polinomios no nulos sobre un mismo cuerpo, serán asociados si y sólo si:

$$P(x) = aP'(x) \quad \wedge \quad P'(x) = bP(x)$$

donde $\frac{a}{b} = 1$.

Por otro lado, si a es un elemento no nulo del cuerpo, y $P(x)$ es un polinomio sobre dicho cuerpo, entonces los polinomios $P(x)$ y $aP(x)$ serán asociados entre sí.

Esto último sirve de base para fundamentar que todo polinomio $P(x)$, no nulo sobre un cuerpo, es asociado a otro cuyo coeficiente principal es la unidad.

Luego, es evidente que un polinomio posee un número infinito de polinomios asociados, y, para cualquier estudio de la divisibilidad de polinomios, relacionado con sus raíces, es lo mismo estudiar el polinomio original que cualquiera de sus polinomios asociados, de modo que se puede tomar, de ellos, el que más nos convenga.

SAGASTUME BERRA y FERNÁNDEZ [29], página 351 y siguientes

15. TEOREMA.

Si el número racional $\frac{c}{d}$, donde c y d son primos relativos entre sí —es decir, no poseen factores comunes—, es raíz de la ecuación de coeficientes enteros:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

con $a_n \neq 0$, entonces c divide a a_0 y d divide a a_n .

HAASER, LASALLE y SULLIVAN [8], página 510

ANEXO #2: APÉNDICE DE CÁLCULO

1. FÓRMULA DE TAYLOR EXACTA PARA POLINOMIOS.

Sea $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$, un polinomio cualquiera (sobre los coeficientes de un cuerpo) de grado n , y sea a un valor cualquiera de la variable x . Entonces: $f(x+a)$ puede escribirse como:

$$\begin{aligned} f(x+a) &= \sum_{i=0}^n \frac{f^i(a)}{i!} x^i \\ &= f(a) + \frac{f'(a)}{1!} x + \frac{f''(a)}{2!} x^2 + \dots + \frac{f^n(a)}{n!} x^n \end{aligned}$$

donde $f^i(x_0)$ denota la i -ésima derivada de la función $f(x)$ en el punto $x = x_0$.

SAGASTUME BERRA y FERNÁNDEZ [29], página 370 y siguientes

2. TEOREMA DE MOIVRE.

Sea $z \in \mathbb{C}$, donde $z = x + iy = r(\cos\theta + i\sin\theta)$, con $r = \sqrt{x^2 + y^2}$ y $\theta = \text{Tan}^{-1}\left[\frac{y}{x}\right]$, entonces las las n raíces n -ésimas de z están dadas por:

$$\sqrt[n]{z} = \sqrt[n]{r} \cdot \left[\cos\left[\frac{\theta + 2\pi k}{n}\right] + i \sin\left[\frac{\theta + 2\pi k}{n}\right] \right],$$

para $k = 0, 1, 2, \dots, n-1$.

SPIEGEL [31], páginas 4-5

3. AXIOMA DEL SUPREMO.

Sea F un cuerpo ordenado y sea S , un subconjunto de F , acotado superiormente. Un elemento $k \in F$ se llama supremo de S (o el extremo superior de S), sí y sólo si k es una cota superior de S y $k \leq m$, $\forall m \in F$ (con m cota superior de S). Si k es el extremo superior de S , se denotará por: $k = \text{Sup}(S)$.

KITCHEN, Jr. [17], página 189

4. TEOREMA.

Dado un subconjunto no vacío $S \subseteq R$, es posible extraer de S una sucesión $\{x_r\}_{r=1}^{\infty}$ tal que $\lim_{x \rightarrow \infty} x_r = \text{Sup}(S)$.

KITCHEN, Jr. [17], página 189

5. CONTINUIDAD DE UN POLINOMIO.

Sea $f(x)$ un polinomio de grado n y sea $\{x_r\}_{r=1}^{\infty}$ una sucesión de números reales que tiende al límite finito a . Entonces $f(x_r)$ tiende al límite $f(a)$.

SAGASTUME BERRA y FERNÁNDEZ [29], página 388

6. TEOREMA.

Si f y g son funciones continuas en el número a , entonces:

$f + g$ es continua en a

$f - g$ es continua en a

$(f)(g)$ es continua en a

$\frac{f}{g}$ es continua en a ; donde $g(a) \neq 0$

Luego, se puede decir, como corolario, que las funciones polinomiales son continuas en todos los elementos de su dominio.

LEITHOLD [20], página 142 y siguientes

7. SUCESIÓN DE STURM APLICADA A FUNCIONES POLINOMIALES.

Consideremos una ecuación de coeficientes reales:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0 \quad (*)$$

que sólo admite raíces simples (reales o complejas).

Construiremos, a partir de (*), una sucesión finita de polinomios en la siguiente forma:

Primer término de la sucesión: $f_1(x) = f(x)$.

Segundo término de la sucesión: $f_2(x) = f'(x)$.

$$\begin{array}{ccccc}
 \cdot & & \cdot & & \cdot \\
 & \cdot & & \cdot & \\
 & & \cdot & & \cdot \\
 & & & \cdot & \\
 & & & & \cdot
 \end{array}$$

i -ésimo término de la sucesión (denotado por $f_i(x)$): el resto, con el signo cambiado, que resulta de dividir $f_{i-2}(x)$ por $f_{i-1}(x)$, o sea que:

$$f_{i-2}(x) = q_{i-1}(x) \cdot f_{i-1}(x) - f_i(x) \quad (**)$$

para $i = 3, 4, \dots, k$, donde los $q_{i-1}(x)$ son lineales.

Debe observarse que en el primer caso de división, se tiene que el dividendo es de grado n , mientras que el divisor; es decir, su derivada, es de grado $(n-1)$; de donde, el cociente es de grado 1, es decir, lineal, una situación similar ocurre con los demás elementos de la sucesión.

Así, después de un número finito de pasos, este proceso termina, obteniéndose una sucesión compuesta a lo sumo de $k = n+1$ polinomios de grados decrecientes, llamada sucesión de Sturm. Dicha sucesión, cumple sobre un intervalo (a, b) de la recta real las siguientes condiciones:

1. Cada $f_i(x)$ es continua en el intervalo.
2. $f_k(x)$ es una constante distinta de cero.
3. Si $f_i(r) = 0$, entonces $f_{i-1}(r) \neq 0$ y $f_{i+1}(r) \neq 0$.

4. Si $f_i(r) = 0$, entonces $f_{i-1}(r)$ y $f_{i+1}(r)$ tienen signos opuestos.

5. Si $f_i(r) = 0$, entonces para h suficientemente pequeña $\frac{f_1(r-h)}{f_2(r-h)} < 0$
y $\frac{f_1(r+h)}{f_2(r+h)} > 0$.

SAGASTUME BERRA y FERNÁNDEZ [29], página 472 y siguientes

SCHEID [30], página 325 y siguientes

8. TEORÍA DE CONTRACCIONES.

Sea f una función definida en algún intervalo cerrado $I \subseteq \mathbb{R}$ ($I = [a, b]$), con imagen en dicho intervalo, tal que $f(a) \cdot f(b) \leq 0$.

Se dice que f es una contracción de I en I si verifica la condición:

$$|f(x) - f(y)| \leq \alpha |x - y|, \quad \forall x, y \in I; \quad \alpha \in (0, 1) \quad (*)$$

La condición (*) es denominada “condición de Lipschitz” y la constante α , constante de Lipschitz, (sea o no $\alpha < 1$). Se puede verificar que si f es una contracción sobre I , entonces f es continua en dicho intervalo.

Geométricamente, la condición de Lipschitz significa que la pendiente de toda recta secante a la función f , en el intervalo I , está acotada por la constante α .

HAASER, LASALLE y SULLIVAN [9], página 695 y siguientes

ENRICI [12], página 75 y siguientes

TÍJONOV y KOSTOMÁROV [33], página 137 y siguientes

9. TEOREMA DEL VALOR MEDIO PARA DERIVADAS.

Sea f una función continua en un intervalo $[a, b]$ y diferenciable en (a, b) (con $a < b$), entonces existe un punto $c \in (a, b)$, tal que:

$$f(b) - f(a) = f'(c)(b - a)$$

HAASER, LASALLE y SULLIVAN [8], página 456 y siguientes

10. DESIGUALDAD TRIANGULAR GENERALIZADA.

Si $a_i \in R$, $i = 1, 2, \dots, k$, entonces:

$$|a_1 + a_2 + \dots + a_k| \leq |a_1| + |a_2| + \dots + |a_k|$$

BRONSHTEIN y SEMENDIAVEV [3], páginas 179-180

11. SERIE GEOMÉTRICA.

La serie geométrica infinita dada por:

$$a + ar + ar^2 + \dots + ar^k + \dots = \sum_{k=0}^{\infty} ar^k, \quad a, r \in R, \quad a \neq 0$$

converge a: $\frac{a}{1-r}$, si $|r| < 1$, y diverge si $|r| \geq 1$.

KAPLAN [16], página 508.

12. FÓRMULA DE TAYLOR CON RESIDUO, PARA FUNCIONES EN GENERAL.

Sea $y = f(x)$ una función, tal que f y sus primeras k derivadas son continuas en un intervalo $[a, b]$. Además f debe satisfacer que su derivada de orden $(k+1)$ es continua en el intervalo (a, b) . Entonces $\exists \xi \in (a, b)$, tal que:

$$f(b) = \sum_{i=0}^k \frac{f^{(i)}(a)}{i!} (b-a)^i + \frac{f^{(k+1)}(\xi)}{(k+1)!} (b-a)^{k+1}$$

donde $f^{(i)}(x_0)$ denota la i -ésima derivada de la función $f(x)$ en el punto $x = x_0$.

Por hipótesis, se asume que $a > b$, sin embargo, también la fórmula es válida para $b < a$, ya que en tal caso, se considera la inversión de los ejes, es decir, el intervalo (a, b) pasa a ser (b, a) y, $[a, b]$, $[b, a]$, respectivamente.

LEITHOLD [20], página 854 y siguientes

13. TEOREMA.

Toda sucesión monótona acotada es convergente.

KAPLAN [16], página 492 y siguientes

ANEXO #3: LISTA DE SÍMBOLOS

Símbolo	Significado
\in	Pertenencia
a_i	i -ésimo coeficiente en las ecuaciones algebraicas
$+$	Adición
\neq	Distinto de
$=$	Igualdad
R	Conjunto de los números reales
$-$	Sustracción
$a/b, \frac{a}{b}$	Cociente de a sobre b
i	Unidad imaginaria pura
C	Conjunto de los números complejos
$[x]^k$	Potencia k -ésima de x
\Leftrightarrow	Doble implicación
\Rightarrow	Implicación
$\sqrt[k]{x}$	Raíz k -ésima de x
$a \cdot b$	Producto de a por b
ω	Raíz cúbica compleja de la unidad
$\sum_{i=1}^k x_i$	Suma de las k x_i
$f^{(k)}$	Derivada k -ésima de f

Símbolo	Significado
[i]	Referencia bibliográfica número i
$\text{Sup}(S)$	Supremo del conjunto S
$\{x_k\}_{k=1}^{\infty}$	Sucesión x_k
\forall	Cuantificador universal
\wedge	Conectiva “y”
$>, <$	Mayor/Menor que
\geq, \leq	Mayor o igual/Menor o igual que
$k!$	Factorial de k
$\text{sen } \theta$	Seno del ángulo θ
$\text{cos } \theta$	Coseno del ángulo θ
$\tan^{-1}(a)$	Arcotangente del número a
θ	Ángulo θ , medido en radianes
$C[a, b]$	Conjunto de funciones continuas en el intervalo $[a, b]$
(a, b)	Intervalo abierto entre a y b
$[a, b]$	Intervalo cerrado entre a y b
$ a $	Valor absoluto de a
N	Conjunto de los números naturales
Δ	Discriminante de una ecuación
π	Pi (relación entre el perímetro de una circunferencia y su diámetro)

Símbolo	Significado
$\lim_{x \rightarrow a} f(x)$	Límite de la función f cuando x tiende al valor a
\exists	Cuantificador existencial
$\operatorname{sgn}(x)$	Función signo de x
$\max(A)$	Máximo de los elementos del conjunto A
$\min(A)$	Mínimo de los elementos del conjunto A
\approx	Aproximadamente igual a

BIBLIOGRAFÍA

1. **ADAMIS**, Eddie. Iniciación al BASIC del IBM/PC. Traducido por José María Vela Bermúdez. Primera Edición. México, D.F., México: Libros McGraw-Hill de México, S.A. de C.V., 1987. 192 páginas.
2. **BOYER**, Carl B. Historia de la matemática. Traducido por Mariano Martínez Pérez. Primera edición. Madrid, España: Alianza Editorial, S.A., 1986. 808 páginas.
3. **BRONSHTEIN**, I., **SEMENDIAVEV**, K. Manual de matemáticas para ingenieros y estudiantes. Sin traductor. Segunda edición. Buenos Aires, Argentina: Ediciones Sapiens, 1976. 696 páginas.
4. **BURDEN**, Richard L., **FAIRES**, J. Douglas. Análisis numérico. Traducido por Simón Mochón C. Primera edición. México, D.F., México: Grupo editorial Iberoamérica, S.A. de C.V., 1985. 721 páginas.
5. **CONTE**, S. D., **DE BOOR**, Carl. Análisis numérico elemental. Traducido por Hernando Alonso Castillo. Segunda Edición. México, D.F., México: Libros McGraw-Hill de México, S.A. de C.V., 1985. 418 páginas.
6. **CHURCHILL**, Ruel V. Teoría de funciones de variable compleja. Traducido por Manfredo Monforte Soler. Segunda Edición. Madrid, España: McGraw-Hill Book Company, 1965. 291 páginas.
7. **GOTTFRIED**, Byron S. Teoría y problemas de programación BASIC. Traducido por Angel Segado Bernal. Tercera edición. Madrid, España: Libros McGraw-Hill de México, S.A. de C.V., 1987. 398 páginas.

8. **HAASER**, Norman, **LA SALLE**, Joseph, **SULLIVAN**, Joseph. Análisis matemático 1: Curso de introducción. Volumen 1. Traducido por Federico Velasco Coba. Primera edición. México, D.F., México: Editorial Trillas, S.A., 1984. 808 páginas.
9. **HAASER**, Norman, **LA SALLE**, Joseph, **SULLIVAN**, Joseph. Análisis matemático 2: Curso intermedio. Volumen 2. Traducido por Federico Velasco Coba. Primera edición. México, D.F., México: Editorial Trillas, S.A., 1985. 786 páginas.
10. **HALL**, H. S., **KNIGHT**, S. R. álgebra superior. Traducido por Rafael García Díaz. Primera edición. México, D.F., México: Unión tipográfica editorial hispanoamericana, 1964. 696 páginas.
11. **HAMMING**, Richard Wesley. Numerical methods for scientists and engineers. Segunda edición. Estados Unidos de América: Editorial McGraw-Hill Book Company, 1973. 721 páginas.
12. **HENRICE**, Peter. Elementos de análisis numérico. Traducido por Federico Velasco Coba. Primera Edición. México, D.F., México: Editorial Trillas, S.A., 1972. 363 páginas.
13. **HILDEBRAND**, F. B. Introduction to numerical analysis. Primera edición. Estados Unidos de América: McGraw-Hill Book Company, Inc., 1956. 511 páginas.
14. **JAMES**, Merlin L., **SMITH**, Gerald M., **WOLFORD**, James C. Métodos numéricos aplicados a la computación digital con FORTRAN. Traducido por José Antonio Nieto Ramirez. Primera edición. México, D.F., México: Representaciones y Servicios de Ingeniería, S.A., 1984. 575 páginas.

15. **JOYANES AGUILAR, Luis.** Programación BASIC para microcomputadoras. Primera edición. México, D.F., México: Libros McGraw-Hill de México, S.A. de C.V., 1985. 441 páginas.
16. **KAPLAN, Wilfred.** Cálculo avanzado. Traducido por Miguel Lara Aparicio. Primera edición. México, D.F., México: Compañía Editorial Continental, S.A., de C.V., 1985. 912 páginas.
17. **KITCHEN, Jr., Joseph W.** Cálculo. Traducido por Lorenzo Abellanas Rapun. Primera Edición. México, D.F., México: Libros McGraw-Hill de México, S.A. de C.V., 1987. 863 páginas.
18. **KREYSZIG, Erwin.** Matemáticas avanzadas para ingeniería. Traducido por José Hernán Pérez Castellanos. Tercera edición. México, D.F., México: Editorial Limusa, S.A., 1981. 1060 páginas.
19. **LANG, Serge.** Álgebra lineal. Traducido por Miguel Lara Aparicio. Segunda edición. México, D.F., México: Fondo Educativo Interamericano, S.A. de C.V., 1985. 400 páginas.
20. **LEITHOLD, Louis.** El cálculo con geometría analítica. Traducido por Juan Carlos Vega Faguana. Quinta edición. Bogotá, Colombia: Editorial Harla, S.A. de C.V., 1987. 1615 páginas.
21. **LIPSCHUTZ, Seymour, POE, Arthur.** Teoría y problemas de programación con FORTRAN. Traducido por Máximo Vilorio Santamaría. Primera edición. México, D.F., México: Libros McGraw-Hill de México, S.A. de C.V., 1986. 314 páginas.

22. **LUTHE**, Rodolfo, **OLIVERA**, Antonio, **SCHUTZ**, Fernando. Métodos numéricos. Primera edición. México, D.F., México: Editorial Limusa, S.A. de C.V., 1986. 443 páginas.
23. **MCCRAKEN**, Daniel D., **DORN**, William S. Métodos numéricos y programación FORTRAN. Traducido por José A. Nieto Ramírez. Primera edición. México, D.F., México: Editorial Limusa, S.A. de C.V., 1986. 476 páginas.
24. **MILNE**, William E., **DAVIS**, David R. Introductory college mathematics. Tercera Edición. Estados Unidos De Norteamérica: Ginn and Company, 1962. 579 páginas.
25. **PINZÓN ESCAMILLA**, Álvaro. Cálculo II (integral). Volumen 2. Primera edición. México, D.F., México: Editorial Harla, S.A. de C.V., 1978. 314 páginas.
26. **RALSTON**, Anthony. Introducción al análisis numérico. Traducido por Carlos E. Cervantes De Gortari. Primera Edición. México, D.F., México: Editorial Limusa-Wiley, S.A., 1970. 629 páginas.
27. **REES**, Paul K., **SPARKS**, Fred W. Álgebra. Traducido por José Emilio Amores. Cuarta Edición. Barcelona, España: Editorial Reverté mexicana, S.A., 1968. 447 páginas.
28. **REINARTH**, Fritz, **SOEDER**, Heinrich. Atlas de matemáticas 1: Fundamentos, álgebra y geometría. Volumen 1. Traducido por Juan Luis Vázquez Suárez y Mario Rodríguez Artalejo. Primera Edición. Madrid, España: Alianza Editorial, S.A., 1984. 265 páginas.

29. **SAGASTUME BERRA**, Alberto E., **FERNÁNDEZ**, Germán. Álgebra y cálculo numérico. Primera edición. Buenos Aires, Argentina: Editorial Kapelusz, S.A., 1960. 726 páginas.
30. **SCHEID**, Francis. Teoría y problemas de análisis numérico. Traducido por Hernando Alonso Castillo. Primera edición. Naulcapán, México: Libros McGraw-Hill de México, S.A. de C.V., 1985. 422 páginas.
31. **SPIEGEL**, Murray R. Teoría y problemas de variable compleja. Traducido por Carlos J. Rodríguez Buitrago. Primera edición. México, D.F., México: Libros McGraw-Hill de México, S.A. de C.V., 1985. 318 páginas.
32. **STIEFEL**, Eduard. Introducción a la matemática numérica. Traducido por Miguel Jerez Juan. Tercera edición. Barcelona, España: Editorial Labor, S.A., 1966. 308 páginas.
33. **TÍJONOV**, A. N., **KOSTOMÁROV**, D. P. Algo acerca de la matemática aplicada. Traducido por M. S. Petrova. Primera edición. Moscú, Unión de Repúblicas Socialistas Soviéticas: Editorial Mir, 1983. 295 páginas.
34. **TREJO**, Cesar A. Funciones de variable compleja. Primera edición. México, D.F., México: Editorial Harla, S.A de C.V., 1974. 452 páginas.
35. **USPENSKI**, V. A. Máquina de Post. Traducido por Stanislav N. Belousov. Primera edición. Moscú, Unión de Repúblicas Socialistas Soviéticas: Editorial Mir, 1983. 96 páginas.
36. **VOEVODIN**, V. V. Álgebra lineal. Traducido por K. P. Medrov. Primera edición. Moscú, Unión de Repúblicas Socialistas Soviéticas: Editorial Mir, 1986. 424 páginas.