

TEORIA Y PRACTICA DE LA APROXIMACION DE PADE

TESIS QUE PRESENTA

JOSE ROBERTO MANDUJANO

PARA OBTENER EL GRADO DE MAESTRO EN MATEMATICAS

ABRIL DE 1991

ASESOR : DR. LUIS VERDE STAR

DIVISION DE CIENCIAS BASICAS E INGENIERIA

UNIVERSIDAD AUTONOMA METROPOLITANA - IZTAPALAPA

INDICE

PROLOGO

I. EL programa PADE

1.1. Introducción	1
1.2. Importancia del programa PADE	2
1.3. Tipo de monitor para usar PADE	5
1.4. Programas principales	5
1.5. Programas para pedir los datos	8
1.6. Programas que contienen los algoritmos	11
1.7. Programas de graficación	22
1.8. Una sesión con PADE	23

II. Fracciones continuas

2.1. Introducción	28
2.2. Notaciones y definiciones	28
2.3. Propiedades básicas	30
2.4. Transformaciones de equivalencia	34
2.5. Representación de funciones a través de de fracciones continuas	35

2.6.	Parte par e impar	37
2.7.	Algoritmo de Viscovatov	42
2.8.	Evaluación de convergentes	51

III. Aproximación de Padé.

3.1.	Introducción	59
3.2.	Problema de aproximación de Padé	59
3.3.	Algoritmo de Gauss	61
3.4.	Fórmulas para estimar el error	70
3.5.	Propiedades básicas	72
3.6.	Normalidad	75
3.7.	Convergencia	83
3.8.	Algoritmo cociente-diferencia	88
3.9.	Algoritmo epsilon	101
3.10.	Algoritmo de Viscovatov	114
3.11.	Fórmulas de recurrencia	116

IV. Interpolación racional.

4.1.	Introducción	119
4.2.	Definiciones	120
4.3.	Algoritmo de Gauss	122
4.4.	Fórmulas para estimar el error	133
4.5.	Propiedades básicas	137

4.6.	Normalidad	145
4.7.	Algoritmo cociente-diferencia	148
4.8.	Algoritmo epsilon	164
4.9.	Interpolación osculatoria	176
4.9.1.	Introducción	176
4.9.2.	Definición	176
4.9.3.	Formas equivalentes del problema de interpolación de Hermite	177
4.9.4.	Algoritmo de Gauss	185
Conclusiones		192
Bibliografía		195

PROLOGO

En 1892 en la Escuela Normal Superior de París, el matemático francés Henri Padé publicó un artículo sobre la aproximación de funciones a través de funciones racionales. El procedimiento para construir esas aproximantes se basa en hacerlas coincidir con las derivadas de la función en una colección de puntos. Existen tres casos importantes

(1) **Problema de aproximación de Padé.** Para una función analítica $f(x) = \sum c_i x^i$, el problema de aproximación de Padé de orden (m, n) consiste en encontrar polinomios p y q , tales que

$$\partial p \leq m, \quad \partial q \leq n \quad \text{y} \quad (fq - p)(x) = \alpha_{m+n+1} x^{m+n+1} + \dots$$

(2) **Problema de interpolación racional.** Para una función de variable compleja f , el problema de interpolación racional de orden (m, n) consiste en construir dos polinomios p y q , tales que

$$\partial p \leq m \quad , \quad \partial q \leq n \quad y$$

$$f(x_i) = (p/q)(x_i) \quad \text{para } i = 0, 1, \dots, m+n$$

(3) **Problema de interpolación racional osculatoria.** Para una función de variable compleja f , una colección de puntos $(x_i)_{i=0}^p$ de \mathbb{C} y una colección de puntos $(s_i)_{i=0}^p$ de \mathbb{N} , se buscan dos polinomios p y q tales

$$\partial p \leq m \quad , \quad \partial q \leq n \quad , \quad m+n+1 = \sum_0^p (s_i + 1) \quad y$$

$$f^{(l)}(x_i) = (p/q)^{(l)}(x_i) \quad \text{para } \begin{array}{l} i = 0, 1, \dots, p \\ l = 0, 1, \dots, s_i \end{array}$$

A partir de los problemas (1), (2) y (3) se obtienen sistemas de ecuaciones, en que las incógnitas son los coeficientes de los polinomios p y q . El algoritmo que consiste en resolver estos sistemas por eliminación gaussiana y pivoteo máximo, lo identificamos en este trabajo como *algoritmo de Gauss*. Este algoritmo tiene la desventaja de tener que resolver tantos sistemas de ecuaciones como aproximantes se quieran construir.

En la década de los cincuenta con el desarrollo de las computadoras, con la búsqueda de métodos de aproximación para

computadoras digitales de las funciones básicas, el desarrollo del lenguaje de alto nivel FORTRAN y el advenimiento de los vuelos espaciales se creó un ambiente adecuado para que se despertara el interés por la teoría de aproximación. Prueba de esto, es que de la década de los cincuenta en adelante nos llegan trabajos sobre este tema que ahora se consideran clásicos, por ejemplo

1955 C. Hastings, Jr., Approximations for digital computers. Princeton Univ. Press, Princeton, N. J.

1955 B. Carlson, M. Goldstein, Rational approximation of functions, Los Alamos Scientific Laboratory.

En esa misma década de los cincuenta la teoría de aproximación de Padé tuvo también un fuerte impulso. En los años de 1955 y 1956 los matemáticos P. Wynn y H. Rutishauser descubrieron los algoritmos epsilon y cociente-diferencia. Estos algoritmos construyen las aproximantes de Padé a través de un procedimiento eficiente.

Mi tesis tuvo como finalidad estudiar la teoría de Padé y la teoría que está detrás de los algoritmos de Rutishauser y Wynn, para después formar un conjunto de programas que nos

permitan trabajar con las aproximantes de Padé de una forma fácil y rápida. Con estos programas podemos conocer los coeficientes de los polinomios que forman a la aproximante, evaluarla, graficarla en un monitor VGA a color etc. El trabajo de programación que desarrollé en estos programas es original.

Hacer paquetes computacionales en las diferentes áreas de la matemática adquiere cada vez mayor importancia. El trabajo para elaborar un paquete de programas en matemáticas, de ninguna manera es fácil, ya que la gente que se dedica a esto debe saber programar al ciento por ciento, pues de ello dependen varias cosas como, la presentación del paquete, que el ambiente entre el usuario y los programas sea "agradable", la versatilidad de los programas(elegir los colores para las gráficas, dar la información a través de un procedimiento sencillo, manejar los programas a través de un "ratón", que los programas sepan mandar los resultados a una impresora en el formato que uno desee, que los programas funcionen en cualquier monitor, etc.). Aparte de esto debe conocer la teoría matemática sobre la cual piensa crear el paquete. Todo esto crea un problema que como ya lo indicamos no es fácil. En este sentido mis programas no llegan a cubrir totalmente estos requerimientos. Esto se debe, en primer lugar, a mis pocos conocimientos de programación, en segundo lugar, los

algoritmos que programé tienen limitaciones que son difíciles de detectar. Sin embargo hay muchos casos interesantes en que mis programas funcionan adecuadamente.

La tesis está organizada de la siguiente forma. En el capítulo uno hacemos una presentación del trabajo de programación que desarrolle, se dan los listados de los programas que contienen los algoritmos y se muestra como utilizar mis programas. Como los algoritmos de Rutishauser y Wynn se obtienen cuando las aproximantes de Padé se representan a través de fracciones continuas, en el capítulo dos hacemos una breve introducción a este tema. En los capítulos tres y cuatro presentamos la teoría de aproximación de Padé, damos las definiciones y propiedades básicas. Pero fundamentalmente estos capítulos van encaminados a desarrollar el concepto de normalidad (el cual tiene que ver con la repetición de aproximantes de Padé y con los casos en que los algoritmos de Wynn y Rutishauser no se pueden aplicar), los algoritmos cociente-diferencia y epsilon, la convergencia de sucesiones de aproximantes de Padé y la estimación del error entre la aproximante y la función.

CAPITULO I. El programa PADE.

1.1. Introducción

En este capítulo se mostrará el trabajo de programación que desarrollé en mi tesis. Como ya se indico en el prólogo, mi trabajo consistio en estudiar la teoría de aproximación racional del tipo Padé para funciones de \mathbb{C} en \mathbb{C} . Pero fundamentalmente comprender como funcionan algunos de los algoritmos que hay en esta teoría, para después formar un paquete de programas, escritos en lenguaje turbo pascal versión cuatro, que nos permitan construir, evaluar y graficar en el monitor de una computadora las aproximantes racionales del tipo Padé para funciones de \mathbb{R} en \mathbb{R} . A este conjunto de programas lo identificaremos por PADE.

Aquí se mostrará cómo utilizar los programas para obtener información numérica [pag. 13], cómo modificar los programas de graficación para que las gráficas puedan aparecer en un monitor CGA, por ejemplo [pág.22], cómo crear archivos de datos para que PADE lea la información y por lo tanto la

graficación sea más ágil [pág. 9,10,11]. Mostraremos también la forma en que PADE está estructurado [pág. 7], presentaremos los listados de los programas que contienen los algoritmos que son utilizados en PADE y en general las diferentes unidades que conforman a PADE.

1.2. Importancia del programa PADE.

Como sabemos el concepto de función en matemáticas y otras áreas es fundamental, a través de él se forman modelos matemáticos que nos sirven para estudiar fenómenos. Por ejemplo, en física podemos estar interesado en estudiar un fenómeno para el cual se tienen, a través de la experimentación, una colección de datos, digamos $(x_1, y_1), \dots, (x_2, y_2)$, el problema es encontrar una función que nos permita estimar o predecir el valor de la variable dependiente de la mejor manera posible. Es también frecuente en las aplicaciones, que nos encontremos con funciones definidas a través de ecuaciones diferenciales, por ejemplo, la ecuación de Bessel

$$z^2 \frac{d^2 w}{dz^2} + z \frac{dw}{dz} + (z^2 - \nu^2)w = 0 \quad (1.1)$$

A partir de la cual se obtienen las funciones de Bessel del primer tipo

$$J_{\nu}(x) = \left(\frac{x}{2}\right)^{\nu} \sum_{k=0}^{\infty} \frac{\left[-\frac{x^2}{4}\right]^k}{k! \Gamma(\nu+k+1)} \quad (1.2)$$

Podemos también estar interesado en estudiar funciones de \mathbb{C} en \mathbb{C} definidas por integrales, como la función de error

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (1.3)$$

Es común manejar funciones definidas implícitamente como

$$w^2xz - e^{-w^2z} = 0 \quad (1.4)$$

Podríamos seguir enumerando más ejemplos en los que tenemos la necesidad de estudiar funciones cuya regla de correspondencia no es fácil de manejar. A través de este tipo de funciones es como surge, como una opción, la teoría de

aproximación racional de Padé. La idea básica de la teoría de aproximación, para analizar este tipo de funciones, es la de utilizar funciones más simple para estudiarlas. En particular la aproximación racional del tipo Padé, utiliza funciones racional como aproximantes.

Ahora bien, cuando queremos llevar a la práctica los algoritmos que nos sirven para aproximar una función, por lo general es necesario hacer un número grande de operaciones como sumas, restas, multiplicaciones, divisiones, potencias, etc. Realizar estas operaciones a "mano" o con una calculadora es poco práctico, además si pensamos aplicar estos algoritmos varias veces nos conviene mejor programarlos. A través de los programas tenemos la oportunidad de trabajar con las aproximantes de una forma fácil y rápida. El valor de unos programas de este tipo es inegable, y lo es mucho más si se utilizan ampliamente. Para convencerse de que esto es así, basta mirar el esfuerzo que sea realizado para crear paquetes en Estadística, Investigación de Operaciones, Análisis Numérico, etc. El Trabajo de programación que hice para formar a PADE es original, paquetes que traten sobre este tema no son fáciles de conseguir. Además paquetes como *TOOLBOX*, *IMSL*, *NAG* o *MATHEMATICA* no tratan el tema de aproximación racional del tipo Padé.

1.3. Tipo de monitor para usar PADE.

Si se utiliza PADE para graficar aproximaciones racionales, es necesario contar con un monitor VGA con 16 colores. Si no se cuenta con un monitor de este tipo es posible hacer algunos cambios a las rutinas de graficación, para que se pueda utilizar PADE, por ejemplo, con un monitor CGA. Como estamos diciendo que es posible modificar los programas de PADE, obviamente los programas que ofrecemos son los *fuentes* y no los *ejecutables*. Utilizando el turbo pascal versión cuatro es como se pueden hacer estos cambios.

La condición de utilizar un monitor VGA se debe a que la alta resolución que tienen estos monitores, 640x480, proporcionan unas gráficas bastante claras. Los colores son necesarios, porque, como PADE puede graficar varias aproximantes es necesario poder distinguir una de otras.

1.4. Programas principales.

El paquete PADE está estructurado de la siguiente forma. Existen tres programas principales llamados *TESIS1.PAS*, *TESIS2.PAS* y *TESIS3.PAS*. Estos programas son los programas

maestros que sirven para graficar aproximaciones racionales, interpolantes racionales e interpolantes racionales de Hermite respectivamente. Si usted desea graficar una interpolante racional, por ejemplo, deberá invocar a TESIS2.PAS. Los programas TESIS1.PAS, TESIS2.PAS y TESIS3.PAS están a su vez formados por las unidades ALGO*.PAS (ver Fig. 1.1). Los programas ALGO*.PAS son los encargados de dar las instrucciones a otros programas más pequeños (ver la sección 1.6), para poder construir las aproximantes del tipo Padé a través de los diferentes algoritmos. A continuación damos una lista de las unidades ALGO*.PAS y lo que hacen.

ALGO1.PAS : Grafica una aproximación racional a través del algoritmo de Gauss [pag. 61].

ALGO21.PAS : Grafica una aproximación racional a través del algoritmo cociente-diferencia [pag. 88]

ALGO22.PAS : Lo mismo que ALGO21.PAS.

ALGO3.PAS : Grafica una aproximación racional a través del algoritmo epsilon [pag. 101].

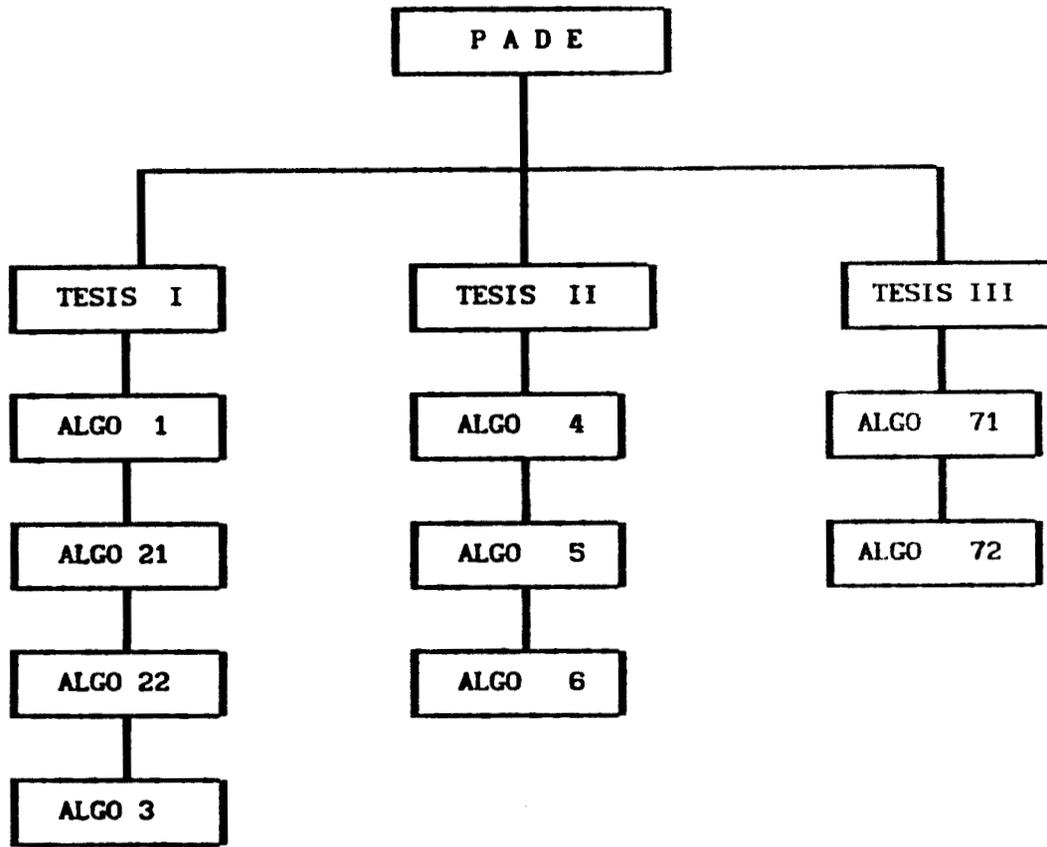


Fig. 1.1

ALGO4.PAS : Grafica una interpolante racional a través del algoritmo de Gauss [pag. 122].

ALGO5.PAS : Grafica una interpolante racional por medio del algoritmo cociente-diferencia [pag. 148].

ALG06.PAS : Grafica una interpolante racional por medio del algoritmo epsilon [pag. 164].

ALG071.PAS : Grafica una interpolante racional de Hermite a través del algoritmo de Gauss [pag. 185].

ALG072.PAS : Igual que ALG071.PAS.

Aparte de estos programas hay una colección de programas pequeños que contienen específicamente los algoritmos que llevan a cabo las operaciones. Digamos que son los programas "obreros" de PADE. Estos algoritmos se encuentran en la sección (1.6).

1.5. Programas para pedir los datos

VARIA.PAS : Unidad que contiene las variables y constantes

DAT1.PAS : Unidad encargada de pedir los datos para formar el recinto de graficación.

DAT2.PAS : Unidad encargada de pedir el orden de las aproximaciones racionales y el algoritmo que se va utilizar para construir el tipo de aproximación racional.

DAT3.PAS : Unidad encargada de pedir los nodos y valores de la función para interpolación racional. Aquí es necesario hacer un comentario más amplio.

El usuario puede dar esta información a través del teclado o de un archivo. Para crear un archivo basta con tener un editor, como el turbo pascal, y escribir un nodo y en el renglón siguiente el valor de la función en ese nodo. Por ejemplo, si se quiere graficar la interpolante racional de orden (1,1) para $f(x) = 1/x$, un posible archivo puede ser

```
-2.0    {#nodo 0#}  
-0.5    {#función en nodo 0#}  
-1.0    {#nodo 1#}  
-1.0    {#función en nodo 1#}  
-0.5    {#nodo 2 #}  
-2.0    {#función en nodo 2#}
```

Es importante que el usuario de exactamente el número de nodos necesarios para construir la interpolante racional. También tenga cuidado en no escribir nodos repetidos. En el caso de que se vayan a graficar varias interpolantes es posible cambiar el

archivo de lectura.

DAT4.PAS : Unidad encargada de pedir los coeficientes de la función analítica para aproximación racional. Al igual que en la unidad anterior se pueden leer los coeficientes de un archivo. Para crear el archivo simplemente se escribe un coeficiente por renglón.

DAT5.PAS : Unidad encargada de pedir los nodos, los valores de la función en los nodos, orden de las derivadas en los nodos y las derivadas para interpolación racional de Hermite. Esta información se debe dar a través de un archivo que es del siguiente tipo. En primer lugar se dan los nodos, uno por renglón. Después se escriben las derivadas del primer nodo desde la de orden cero a la de orden seis, si el usuario tiene sólo las derivadas hasta el orden tres, por ejemplo, deberá escribir que las restantes son cero. Se repite este procedimiento tantas veces como nodos se tengan. Por ejemplo, supongamos que para la función exponencial contamos con la siguiente información. Los nodos $x_0 = 0$, $x_1 = 1$ y $x_2 = 3$ y las derivadas de la función en los nodos hasta el orden dos. Si denotamos por s_0 , s_1 , s_2 el orden de las derivadas, entonces la relación entre los datos y el orden de la interpolante que podemos

construir debe ser [pag. 179]

$$m + n + 1 = (s_0+1)+(s_1+1)+(s_2+1)$$

Luego entonces podemos construir una interpolante racional de Hermite de orden (4,4). Para este ejemplo necesitamos un archivo del siguiente tipo

```
-1      Nodo 0
0      Nodo 1
3      Nodo 2
0.3678794  Derivada de orden 0 en Nodo 0
0.3678794  Derivada de orden 1 en Nodo 0
0.3678794  Derivada de orden 2 en Nodo 0
0  No se tiene la derivada de orden 3 en el nodo 0 o vale 0
0  No se tiene la derivada de orden 4 en el nodo 0 o vale 0
0  No se tiene la derivada de orden 5 en el nodo 0 o vale 0
0  No se tiene la derivada de orden 6 en el nodo 0 o vale 0
```

A continuación se hacen otros siete renglones para las derivadas del nodo uno, y lo mismo para el nodo dos.

1.6. Programas que contienen los algoritmos

PROC1.PAS : Unidad encargada de resolver un sistema de ecuaciones lineales por eliminación gaussiana y pivoteo máximo. Este procedimiento es el que utilizan los programas TESIS1.PAS, TESIS2.PAS y TESIS3.PAS para encontrar los coeficientes de los polinomios que forman cualquiera de las aproximaciones del tipo Padé.

PROC2.PAS : Unidad encargada de formar la tabla epsilon para interpolación racional. Las variables *m* y *n* le proporcionan al procedimiento el orden de la interpolante, *Parcial* proporciona la evaluación de los polinomios interpolantes, *Punto* proporciona el valor en que se evalúa la tabla epsilon, *Nodo* proporciona los nodos en que se interpola.

```

Unit Proc2;
Begin
  For j:=m-n to m+n do
    Begin
      e1[0,j]:=Parcial[j];
      e1[-1,j]:=0;
    End;
  For i:=1 to 2*n do
    Begin
      For j:=m-n to m+n-i do
        Begin
          If Abs((e1[i-1,j+1]-e1[i-1,j])*(Punto-Nodo[i+j]))
            > MinimoReal then
            e1[i,j]:=e1[i-2,j+1]+1/((Punto-Nodo[i+j])*(e1[i-1,j+1]-e1[i-1,j]));
          End;
        End;
      Evaluacion:=e1[2*n,m-n];
    End;
  End;
End.

```

Si el usuario está interesado en conocer esta tabla, sólo tiene que crear un archivo del tipo texto y guardar la información allí. Para crear el archivo siga los siguientes pasos

PROCEDIMIENTO PARA CREAR UN ARCHIVO

1. Defina una variable ARCH del tipo TEXTO dentro del conjunto de variables locales de este procedimiento.

2. Después de la primer instrucción BEGIN del procedimiento, escriba las siguientes instrucciones

```
Assign(Arch,'Trayectoria del archivo a crear');
```

```
Rewrite(Arch);
```

3. Utilice una instrucción del tipo

```
Writeln(Arch,variables a escribir en el archivo);
```

para mandar escribir al archivo la información requerida

4. Ponga una instrucción de la forma : Close(Arch), después de que son ejecutadas las instrucciones del paso 3, para

cerrar el archivo.

Este procedimiento para extraer información de PADE lo puede emplear en cualquier otra unidad. Una observación final, la tabla epsilon se construye o evalúa en un punto, por esta razón debe tenerse cuidado si los valores que toma PUNTO son muchos, ya que entonces el archivo puede ser muy grande y no crearse. La parte donde se corrige el número de puntos en que se evalúa una aproximante del tipo Padé se encuentra en las unidades *ALGO*.PAS*. Para mayor información sobre estas unidades ver los capítulos tres y cuatro en las secciones donde se localizan los algoritmos.

PROC3.PAS : Unidad encargada de formar una tabla de diferencias divididas con nodos repetidos. La variable *NumNodos* proporciona al procedimiento el número de nodos, *DF* las derivadas de la función, *Nodo* los puntos de interpolación, *u* los ordenes de las derivadas que se conocen en cada nodo. La función *Factorial* sirve para calcular factoriales. Este procedimiento se encuentra en la página 16.

PROC4.PAS : Unidad encargada de hacer una tabla de diferencias divididas sin nodos repetidos.

PROC5.PAS : Unidad encargada de formar la tabla epsilon para aproximación racional. Las variables *m* y *n* proporcionan el orden de la aproximante, *Parcial* las sumas parciales del desarrollo en serie de potencias de la función. Una copia de esta unidad se encuentra al final de esta página.

PROC6.PAS : Unidad encargada de formar la tabla cociente-diferencia para aproximación racional. La variable *Coef* proporciona los coeficientes del desarrollo en serie de potencias de la función y *m* y *n* el orden de la aproximante. Este procedimiento se encuentra en la página 17.

```

Unit Proc5;
Begin
  For j:=m-n to m+n do
    Begin
      e1[0,j]:=Parcial[j];
      e1[-1,j]:=0;
    End;
  For i:=1 to 2*n do
    Begin
      For j:=m-n to m+n-i do
        Begin
          If Abs(e1[i-1,j+1]-e1[i-1,j])>MinimoReal then
            e1[i,j]:=e1[i-2,j+1]+1/(e1[i-1,j+1]-e1[i-1,j]);
          If Abs(e1[i-1,j+1]-e1[i-1,j])<MinimoReal then
            e1[i,j]:=MaximoReal;
        End;
      End;
    End;
  Evaluacion:=e1[2*n,m-n];
End;
End.

```

```

Unit Proc3;
Begin
  suma:=0;
  For i:=0 to NumNodos do
    suma:=suma+u[i];
  End;
  u[-1]:=0;d[-1]:=0;
  For i:=0 to NumNodos do
    Begin
      d[i]:=u[i-1]+d[i-1];
      For j:=0 to u[i]-1 do
        y[j+d[i]]:=Nodo[i];
      End;
    End;
  u[-1]:=0;d[-1]:=0;
  For i:=0 to NumNodos do
    Begin
      d[i]:=u[i-1]+d[i-1];
      For j:=0 to u[i]-1 do
        DifeDivididas[j+d[i],0]:=DF[i,0];
      End;
    End;
  u[-1]:=0;d[-1]:=0;
  For j:=0 to suma-1 do
    Begin
      For i:=0 to NumNodos do
        Begin
          d[i]:=u[i-1]+d[i-1];
          For p:=0 to u[i]-1 do
            Begin
              If (p+d[i])>=j then
                Begin
                  If y[p+d[i]]<>y[p+d[i]-j] then
                    DifeDivididas[p+d[i],j]:=(DifeDivididas[p+d[i],j-1]-
                    DifeDivididas[p+d[i]-1,j-1])/(y[p+d[i]]-y[p+d[i]-j]);
                  If y[p+d[i]]=y[p+d[i]-j] then
                    DifeDivididas[p+d[i],j]:=DF[i,j]/FACTORIAL(j);
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  End.

```

```

Unit Proc6;
Begin
  For j:= m-n+1 to m+n-1 Do
    Begin
      If Abs(Coef[j])>MinimoReal then
        q1[1,j]:= Coef[j+1]/Coef[j];
      If Abs(Coef[j])<MinimoReal then
        q1[1,j]:=MaximoReal;
    End;
  For j:= m-n+1 to m+n-2 Do
    e1[1,j]:= q1[1,j+1]-q1[1,j];
  End;
  For i:= 2 to n Do
    Begin
      For j:= m-n+1 to m+n-2*i+1 Do
        Begin
          If Abs(e1[i-1,j])>MinimoReal then
            q1[i,j]:=(q1[i-1,j+1]*e1[i-1,j+1])/e1[i-1,j];
          If Abs(e1[i-1,j])<MinimoReal then
            q1[i,j]:=MaximoReal;
        End;
      If (i<n) Then
        Begin
          For j:= m-n+1 to m+n-2*i Do
            e1[i,j]:=e1[i-1,j+1]+q1[i,j+1]-q1[i,j];
          End;
        End;
    End;
  For i:=1 to n Do
    Begin
      q[i]:=q1[i,m-n+1];
      e[i]:=e1[i,m-n+1];
    End;
  End;
End.

```

PROC7.PAS : Unidad encargada de formar la tabla *qd* para interpolación racional. Las variables *DifDiv1* y *DifDiv2* proporcionan diferencias divididas y *m* y *n* el orden de la

interpolante racional.

```
Unit Proc7;
Begin
  For j:= m-n+1 to m+n-1 Do
    Begin
      If Abs(DifDiv2[j+1])>MinimoReal then
        q1[1,j]:= DifDiv1[j+1]/DifDiv2[j+1];
      If Abs(DifDiv2[j+1])<MinimoReal then
        q1[1,j]:=MaximoReal;
    End;
  For j:= m-n+1 to m+n-2 Do
    Begin
      If Abs(1+q1[1,j]*(Nodo[0]-Nodo[j+1]))<MinimoReal then
        e1[1,j]:=(q1[1,j+1]-q1[1,j])/(1+q1[1,j]*(Nodo[0]-Nodo[j+1]));
    End;
  For i:= 2 to n Do
    Begin
      For j:= m-n+1 to m+n-2*i+1 Do
        Begin
          Num1:=e1[i-1,j+1]*q1[i-1,j+1]*(1+e1[i-1,j]*
            (Nodo[0]-Nodo[j+2*i-2]));
          Den1:=e1[i-1,j]*(1+q1[i-1,j+1]*(Nodo[0]-Nodo[j+2*i-3]))+
            e1[i-1,j+1]*(e1[i-1,j]-q1[i-1,j+1])*
            (Nodo[0]-Nodo[j+2*i-1]);
          If Abs(Den1)>MinimoReal then
            q1[i,j]:=Num1/Den1;
          End;
        If (i<n) Then
          Begin
            For j:= m-n+1 to m+n-2*i Do
              Begin
                Num2:=q1[i,j+1]-q1[i,j]+e1[i-1,j+1]*
                  (1+q1[i,j+1]*(Nodo[0]-Nodo[j+2]));
                Den2:=1+q1[i,j]*(Nodo[0]-Nodo[j+2*i-1]);
                If Abs(Den2)>MinimoReal then
                  e1[i,j]:=Num2/Den2;
                End;
              End;
            End;
          End;
        End;
      For i:=1 to n Do
        Begin
          q[i]:=q1[i,m-n+1];e[i]:=e1[i,m-n+1];
        End;
      End;
    End;
  End.
```

PROC8.PAS. : Unidad encargada de evaluar un polinomio por el algoritmo de Horner.

PROC9.PAS. : Unidad encargada de evaluar un polinomio, escrito como combinación de los polinomios de Newton. La variable *Grado* proporciona el grado del polinomio, *DifDiv* diferencias divididas, *Nodo* los puntos de interpolación y *Punto* el valor en que se evalúa el polinomio.

```
Unit Proc9;
Begin
  If (grado>1) then
    Begin
      Result:=DifDiv1[grado];
      For i:=grado-1 Downto 0 do
        Begin
          Result:=Result*(Punto-Nodo[i])+DifDiv1[i];
        End;
      End;
    End;
  If Grado=1 then
    Begin
      Result:=DifDiv1[1]*(Punto-Nodo[0])+DifDiv1[0];
    End;
  If grado=0 then
    Begin
      Result:=DifDiv1[0];
    End;
End;
End.
```

PROC10.PAS : Unidad encargada de formar la fracción continua que representa a la aproximación racional, en el

algoritmo cociente-diferencia. Las variables *e* y *q* proporcionan los elementos de la tabla *qd* para formar la fracción continua *Coef* los coeficientes del desarrollo en serie de potencias, *or1* y *or2* el orden de la aproximante, *Orden* el orden de la convergente y *Punto* el punto en que se evalúa la aproximante.

```

Unit   Proc10;
Begin
  For  i:=1 to Orden Do
    Begin
      Num[2*i]:=-q[i]*Punto;
      Num[2*i+1]:=-e[i]*Punto;
      Den[2*i]:=1.0;
      Den[2*i+1]:=1.0;
    End;
  End;
End;
End.

```

PROC11.PAS : Unidad encargada de formar la fracción continua que representa a la interpolante racional en el algoritmo cociente-diferencia. Las variables *e* y *q* proporcionan al procedimiento la parte de la tabla *qd* necesaria para formar la fracción continua, *Nodo* los puntos en que se interpola, *Orden* el orden de la convergente de la fracción continua que representa a la interpolante racional, *Punto* el valor en que se evalúa la interpolante, *DifDiv1* proporciona las diferencias divididas y

or1 y or2 el orden de la interpolante racional.

```
Unit Proc11;
Begin
  For i:=1 to Orden do
    Begin
      Num[2*i]:=-q[i]*(Punto-Nodo[or1-or2+2*i-1]);
      Num[2*i+1]:=-e[i]*(Punto-Nodo[or1-or2+2*i]);
      Den[2*i]:=1+q[i]*(Nodo[0]-Nodo[or1-or2+2*i-1]);
      Den[2*i+1]:=1+e[i]*(Nodo[0]-Nodo[or1-or2+2*i]);
    End;
  Den[1]:=1;
  Num[1]:=(Punto-Nodo[0])*DifDiv1[or1-or2+1];
  If or1>or2 then
    Begin
      For i:=1 to or1-or2 do
        Num[1]:=Num[1]*(Punto-Nodo[i]);
      End;
    End;
End;
End.
```

PROC12.PAS : Procedimiento que se encarga de evaluar la convergente de una fracción continua. La variable *B0* proporciona el denominador parcial cero, *Num* y *Den* los numeradores y denominadores parciales de la fracción continua, *Orden* el orden de la convergente.

```
Unit Proc12;
Begin
  Conv:=0;
  For i:=Orden Downto 1 Do
    Begin
      If Abs(Den[i]+Conv)>MinimoReal then
        Conv:=Num[i]/(Den[i]+Conv);
      End;
    Conv:=B0+Conv;
  End;
End;
End.
```

PROC13.PAS : Unidad encargada de calcular potencias de números reales.

PROC14.PAS : Unidad encargada de calcular los coeficientes de la serie recíproca de una serie dada.

1.7. Programas de graficación.

DIB.PAS : Unidad de graficación

```
*****
Procedimiento para llamar al modo grafico
*****
Begin
  Gd:=Detect;
  InitGraph(Gd,Gm,'c:tp4');
  If GraphResult<>grok then
    Halt(1);
  GetModeRange(Gd,LowMode,HighMode);
  SetGraphMode(HighMode);
End;
*****
Programa para trazar una linea entre dos puntos en la pantalla
*****
Begin
  Line(Round((480*(abscisa1-Xmin)/D1)+160),
    Round(480*(ordenada1-Ymax)/D2),
    Round((480*(abscisa2-Xmin)/D1)+160),
    Round(480*(ordenada2-Ymax)/D2));
End;
```

los siguientes comentarios son importantes.

Como ya se ha dicho, PADE solamente funciona con monitores VGA a

color. En el caso de que no se tenga un monitor con estas características, es necesario modificar los procedimientos de la unidad DIB.PAS. Por ejemplo, si se quieren graficar las aproximantes del tipo Padé en un monitor CGA con resolución de 640x200. Se buscan dos funciones lineales de la forma

$$L : [X_{\min}, X_{\max}] \longrightarrow [\alpha, \beta]$$

$$G : [Y_{\min}, Y_{\max}] \longrightarrow [\gamma, \delta] , \quad G'(x) < 0$$

donde $0 \leq \alpha < \beta \leq 640$ y $0 \leq \gamma < \delta \leq 200$. Con estas funciones el procedimiento para graficar en CGA es

```
Line(Round(L(abcisa1)),
      Round(G(ordenada1)),
      Round(L(abcisa2)),
      Round(G(Ordenada2)));
```

1.8. Una sesión con PADE.

Supongamos que se quiere tener una sesión con PADE en la parte de interpolación.

El primer paso es teclear TESIS2 y return, inmediatamente aparece una pantalla en la que se nos pide

introducir los datos para formar el recinto de graficación. Esta pantalla es de la siguiente forma

RECINTO DE GRAFICACION	
Xmin	:=
Xmax	:=
Ymin	:=
Ymax	:=
NumAprox	:=
EscalaX	:=
EscalaY	:=

Deberá llenarse esta pantalla, tomando en cuenta que $[Xmin, Xmax] \times [Ymin, Ymax]$ formará la región de graficación. Obviamente debe escoger $Xmin < Xmax$ y $Ymin < Ymax$. A la variable *NumAprox* se le asignará el natural entre 1 y 5 que le indique al programa cuántas aproximaciones se graficarán. Las variables *EscalaX* y *EscalaY* sirven para dar la separación entre dos marcas en los ejes de coordenadas. Después de proporcionar esta información aparece una indicación de si quiere corregir los datos, la respuesta es S(SI) o N(NO).

A continuación aparece una pantalla que

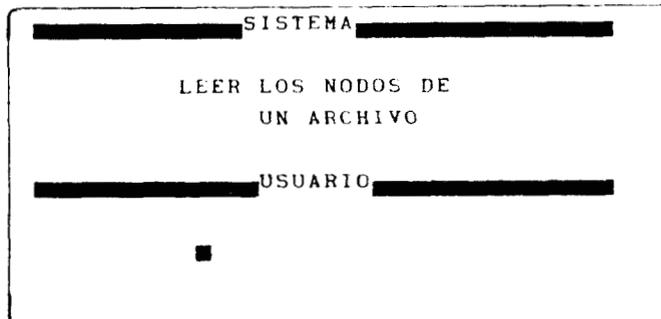
depende del valor de *NumAprox*. Supongamos que esta variable tiene el valor 2, entonces la pantalla es de la siguiente forma

```
ORDEN DE LAS APROXIMACIONES
m[1]          :=
n[1]          :=
Algoritmo[1] :=

m[2]          :=
n[2]          :=
Algoritmo[2] :=
```

En este caso deberá dar la información tomando en cuenta que *m* sirve para denotar el grado del polinomio numerador y *n* del polinomio denominador en la interpolante. A la variable *Algoritmo* se le pueden asignar los valores 1, 2 y 3 para el caso de aproximación e interpolación racional, y 1 para el caso de interpolación racional de Hermite. El número 1 corresponde a los algoritmos de Gauss, el 2 corresponde al algoritmo cociente diferencia y el 3 al algoritmo epsilon. Los algoritmos 2 y 3, sólo se deben utilizar si la aproximante racional es normal.

Después de esto aparece una pantalla del siguiente tipo



A lo que el usuario deberá responder S(SI) o N(NO). Si el usuario ya creo su archivo en el que se encuentra la información, la respuesta es S(si). En ese caso el programa le pregunta si quiere cambiar el archivo de lectura dependiendo de la interpolante. Si contesta que no, un sólo archivo se utilizará para formar las interpolantes. Si contesta que si, el programa le pedirá la(s) trayectoria(s) de los archivo(s) donde se encuentra la información. Como última pregunta, el programa pedirá que le indique si desea graficar la función, si la respuesta es S(SI), se tendrá que dar la trayectoria del archivo que contiene la información. A continuación se presenta un programa, como ejemplo, de como crear un archivo para graficar la función exponencial. Este se puede hacer con turbo pascal versión cuatro.

PROGRAMA PARA CREAR UN ARCHIVO

```
Program CREAM(Input,Output,Arch);
Uses CRT;
Var Arch : Text;
    i : Integer;
    x,y : Real;
    Trayectoria : String[20];
Begin
  ClrScr;
  Write('Trayectoria del archivo . . .');
  Readln(Trayectoria);
  Assign(Arch,'Trayectoria');
  Rewrite(Arch);
  For i:=0 to 200 do
    Begin
      x:=xmin+(xmax-xmin)*i/200;
      y:=exp(x);
      Writeln(Arch,x);
      Writeln(Arch,y);
    End;
  Close(Arch);
End.
```

Con esto terminamos este capítulo, en el siguiente abordaremos el tema de fracciones continuas que es necesario para desarrollar los algoritmos cociente-diferencia de Rutishauser y epsilon de Wynn.

CAPITULO II. FRACCIONES CONTINUAS

2.1. Introducción.

Los métodos con que se contaban, en los tiempos de Padé (1892), para calcular aproximaciones racionales, del tipo Padé, eran a través de fórmulas de determinantes y de soluciones de sistemas de ecuaciones. Alrededor de 1955 y 1956 H. Rutishauser [7] y P. Wynn [11] encontraron algoritmos que permiten construir las aproximaciones de Padé eficientemente. La idea básica para construir el algoritmo cociente-diferencia de Rutishauser y epsilon de Wynn, es la de representar las aproximaciones de Padé a través de fracciones continuas. Por esta razón, este capítulo está dedicado a mostrar los conceptos que utilizaremos en los capítulos tres y cuatro sobre el tema de fracciones continuas.

2.2. Notaciones y definiciones.

Definición 2.1. Sean $(a_i)_{i=1}^{\infty}$ y $(b_i)_{i=0}^{\infty}$ sucesiones en \mathbb{C} o funciones de \mathbb{C} en \mathbb{C} . Una *fracción continua* con numeradores

parciales $(a_i)_{i=1}^{\infty}$ y denominadores parciales $(b_i)_{i=0}^{\infty}$ es una fórmula del tipo

$$b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \dots}}} \quad (2.1) \quad \blacksquare$$

Una notación más compacta para la expresión (2.1) es

$$b_0 + \sum_{i=0}^{\infty} \frac{a_i}{b_i} \quad (2.2)$$

Definición 2.2. Para la fracción continua (2.1) se define la n -ésima convergente como

$$C_n = \begin{cases} b_0 & , \text{si } n = 0 \\ b_0 + \sum_{i=1}^n \frac{a_i}{b_i} & , \text{si } n > 0 \end{cases} \quad (2.3) \quad \blacksquare$$

Definición 2.3. Si la sucesión $(C_n)_{n=0}^{\infty}$ converge a C , diremos que (2.1) converge a C y por lo tanto que (2.1) vale C . \blacksquare

Si se efectúan las operaciones en (2.3), se obtendrá una función racional que depende de los $2n + 1$ numeradores y

denominadores parciales $b_0, a_1, b_1, \dots, a_n, b_n$.

$$C_n = \frac{P_n (b_0, a_1, b_1, \dots, a_n, b_n)}{Q_n (b_0, a_1, b_1, \dots, a_n, b_n)} \quad (2.4)$$

Definición 2.4. Los polinomios P_n y Q_n son llamados el n -ésimo numerador y denominador de la fracción continua (2.1) respectivamente. ■

2.3. Propiedades básicas.

Los polinomios P_n y Q_n se pueden construir recursivamente como lo ilustra el siguiente resultado.

Proposición 2.1. Si $P_{-1} = 1$, $Q_{-1} = 0$, $P_0 = b_0$ y $Q_0 = 1$, entonces para $n \geq 1$ se cumple

$$\begin{aligned} P_n &= b_n P_{n-1} + a_n P_{n-2} \\ Q_n &= b_n Q_{n-1} + a_n Q_{n-2} \end{aligned} \quad (2.5)$$

Demostración. La demostración se hace por inducción matemática en n . Para $n = 1$ se obtienen las fórmulas directamente. Ahora supongamos que (2.5) se satisface para n y

se demostrará para $n+1$. Como

$$\begin{aligned}
 C_{n+1} &= b_0 + \frac{a_1}{b_1} + \frac{a_2}{b_2} + \dots + \frac{a_n}{b_n + \frac{a_{n+1}}{b_{n+1}}} \\
 &= \frac{P_n \left(b_0, a_1, b_1, \dots, a_n, b_n + \frac{a_{n+1}}{b_{n+1}} \right)}{Q_n \left(b_0, a_1, b_1, \dots, a_n, b_n + \frac{a_{n+1}}{b_{n+1}} \right)}
 \end{aligned}$$

Entonces por la hipótesis de inducción podemos concluir que

$$\begin{aligned}
 &= \frac{\left(b_n + \frac{a_{n+1}}{b_{n+1}} \right) P_{n-1} + a_n P_{n-2}}{\left(b_n + \frac{a_{n+1}}{b_{n+1}} \right) Q_{n-1} + a_n Q_{n-2}},
 \end{aligned}$$

aplicando de nuevo la hipótesis de inducción tenemos

$$= \frac{P_n + \frac{a_{n+1}}{b_{n+1}} P_{n-1}}{Q_n + \frac{a_{n+1}}{b_{n+1}} Q_{n-1}} = \frac{b_{n+1} P_n + a_{n+1} P_{n-1}}{b_{n+1} Q_n + a_{n+1} Q_{n-1}}. \quad \blacksquare$$

Ahora se da una fórmula para la diferencia de dos convergentes consecutivas.

Proposición 2.2. Si $Q_n Q_{n-1} \neq 0$ entonces

$$C_n - C_{n-1} = (-1)^{n+1} \frac{a_1 a_2 \cdots a_n}{Q_n Q_{n-1}} \quad (2.6)$$

Demostración. Se probará primero por inducción que

$$P_n Q_{n-1} - Q_n P_{n-1} = (-1)^{n+1} a_1 a_2 \cdots a_n \quad (2.7)$$

Para $n=1$ la fórmula (2.7) es claramente verdadera. Suponga que es válida para n y se demostrará que lo es para $n+1$. Utilizando las fórmulas (2.5) para $n+1$ y la hipótesis de inducción en $P_{n+1} Q_n - Q_{n+1} P_n$, se obtiene la fórmula (2.7). Finalmente (2.6) se deduce factorizando $Q_n Q_{n-1}$ de (2.7). ■

Ahora se demostrarán dos proposiciones que van a jugar un papel importante en la sección [2.6].

Proposición 2.3. Para los numeradores y denominadores de la fracción continua (2.1) se satisface

$$P_{n+1} Q_{n-1} - P_{n-1} Q_{n+1} = (-1)^{n+1} a_1 a_2 \cdots a_n b_{n+1} \quad (2.8)$$

Demostración. Aplicando las fórmulas (2.5) a P_{n+1} y Q_{n+1} podemos escribir

$$\begin{aligned}
 P_{n+1}Q_{n-1} - P_{n-1}Q_{n+1} &= (b_{n+1}P_n + a_{n+1}P_{n-1})Q_{n-1} - \\
 -(b_{n+1}Q_n + a_{n+1}Q_{n-1})P_{n-1} &= b_{n+1}(P_nQ_{n-1} - P_{n-1}Q_n) = \\
 &= (-1)^{n+1} a_1 a_2 \dots a_n b_{n+1}
 \end{aligned}$$

La última igualdad se obtuvo aplicando (2.7). ■

Proposición 2.4. La siguiente igualdad es válida para los numeradores y denominadores de la fracción continua (2.1).

$$P_{n+2}Q_{n-2} - P_{n-2}Q_{n+2} = (-1)^n a_1 a_2 \dots a_{n-1} [b_n(b_{n+1}b_{n+2} + a_{n+2}) + b_{n+2}a_{n+1}] \quad (2.9)$$

Demostración. Aplicando las fórmulas (2.5) a P_{n+2} , Q_{n+2} , P_{n+1} , Q_{n+1} , P_n y Q_n , y simplificando se obtiene:

$$\begin{aligned}
 P_{n+2}Q_{n-2} - P_{n-2}Q_{n+2} &= (b_{n+2}P_{n+1} + a_{n+2}P_n)Q_{n-2} - \\
 -(b_{n+2}Q_{n+1} + a_{n+2}Q_n)P_{n-2} &=
 \end{aligned}$$

$$\begin{aligned}
&= [b_{n+2} (b_{n+1} P_n + a_{n+1} P_{n-1}) + a_{n+2} P_n] Q_{n-2} - \\
&- [b_{n+2} (b_{n+1} Q_n + a_{n+1} Q_{n-1}) + a_{n+2} Q_n] P_{n-2} = \\
&b_{n+2} b_{n+1} (P_n Q_{n-2} - Q_n P_{n-2}) + b_{n+2} a_{n+1} (P_{n-1} Q_{n-2} - Q_{n-1} P_{n-2}) + \\
&a_{n+2} (P_n Q_{n-2} - Q_n P_{n-2}) = \\
&+ b_{n+2} a_{n+1} [P_{n-1} Q_{n-2} - Q_{n-1} P_{n-2}] + a_{n+2} b_n [P_{n-1} Q_{n-2} - Q_{n-1} P_{n-2}] \\
&= (P_{n-1} Q_{n-2} - Q_{n-1} P_{n-2}) [b_{n+2} b_{n+1} b_n + b_{n+2} a_{n+1} + a_{n+2} b_n] = \\
&= (-1)^n a_1 a_2 \cdots a_{n-1} [b_n (b_{n+1} b_{n+2} + a_{n+2}) + b_{n+2} a_{n+1}]
\end{aligned}$$

La última igualdad se obtuvo aplicando (2.7). ■

2.4. Transformaciones de equivalencia.

Definición 2.5. Sea $(p_i)_{i=0}^{\infty}$ una sucesión con $p_i \neq 0$. La transformación que cambia la fracción continua (2.1) en

$$b_0 + \frac{p_1 a_1}{p_1 b_1} + \sum_{i=2}^{\infty} \frac{p_{i-1} p_i a_i}{p_i b_i} \quad (2.10)$$

es llamada una transformación de equivalencia. ■

Las transformaciones de la definición (2.5) reciben el

nombre de equivalencia porque las fracciones continuas (2.1) y (2.10) coinciden en sus n -ésimas convergentes. Es decir, podemos escribir una fracción continua de diferentes formas y no alterar su valor.

Definición 2.6. Dos fracciones continuas son equivalentes si sus n -ésimas convergentes son iguales, para $n \geq 0$. ■

2.5. Representación de funciones a través de fracciones continuas.

Un problema importante en la teoría de fracciones continuas es la representación de números complejos o funciones a través de fracciones continuas. Este problema es el que llevó a Rutishauser y Wynn a construir sus algoritmos.

Consideremos el siguiente problema. Sea $\sum_{i=0}^{\infty} c_i x^i$ una serie con sumas parciales $(S_i)_{i=0}^{\infty}$. Queremos construir una fracción continua, tal que su n -ésima convergente sea igual a S_n . Esta construcción es importante ya que nos permite escribir una función analítica como fracción continua. Es claro que los puntos de convergencia de la serie lo son también de la fracción continua, sin embargo, en general la región de convergencia de

la fracción continua es más grande que el círculo de convergencia de la serie. Aunque estamos planteando el problema para representar una función analítica, el siguiente resultado es válido para cualquier sucesión $(S_n)_{n=0}^{\infty}$.

Proposición 2.5. Sea $(S_n)_{n=0}^{\infty}$ una sucesión en la que no hay tres elementos consecutivos iguales. La fracción continua (2.1) con

$$b_0 = S_0, \quad a_1 = S_1 - S_0, \quad b_1 = 1$$

$$a_i = \frac{S_{i-1} - S_i}{S_{i-1} - S_{i-2}}, \quad b_i = \frac{S_i - S_{i-2}}{S_{i-1} - S_{i-2}}, \quad i \geq 2 \quad (2.11)$$

tiene a S_n como n-ésima convergente.

Demostración. Para empezar, obsérvese que si tres elementos consecutivos son iguales en la sucesión $(S_n)_{n=0}^{\infty}$ entonces $a_i = b_i = 0$, para algún $i \geq 1$. Esto trae como consecuencia que alguna convergente de la fracción continua (2.1) no exista. Pasando ahora a la demostración del resultado, se observa que si $C_n = P_n/Q_n$ y como se quiere que $C_n = S_n$, entonces la condición se satisface si escogemos $P_n = S_n$ y $Q_n = 1$, para $n \geq 0$. A partir de estas ecuaciones y de (2.5) se tiene :

$$b_0 = S_0, \quad b_1 b_0 + a_1 = S_1, \quad b_1 = 1$$

$$b_i S_{i-1} + a_i S_{i-1} = S_i, \quad i \geq 2$$

$$b_i + a_i = 1, \quad i \geq 2$$

Resolviendo el sistema anterior para las incógnitas a_i y b_i se obtiene (2.11). ■

La fracción continua de la proposición (2.5) se puede escribir también como:

$$S_0 + \frac{S_1 - S_0}{1} + \frac{S_1 - S_2}{S_2 - S_0} + \sum_{i=3}^{\infty} \frac{(S_{i-2} - S_{i-3})(S_{i-1} - S_i)}{S_i - S_{i-2}} \quad (2.12)$$

La cual se obtiene utilizando la transformación de equivalencia $(p_i)_{i=1}^{\infty}$, con $p_1 = 1$ y $p_i = S_{i-1} - S_{i-2}$, para $i \geq 2$.

2.6. Parte par e impar

Aprovechando las fórmulas (2.11) se definirá la parte par e impar de una fracción continua y además se dará una fórmula para ellas.

Definición 2.7. Para la fracción continua (2.1), con convergentes $(C_n)_{n=0}^{\infty}$, se definen su parte par como una fracción continua cuyas convergentes son $(C_{2n})_{n=0}^{\infty}$ y su parte impar como una fracción continua cuyas convergentes son $(C_{2n+1})_{n=0}^{\infty}$. ■

Proposición 2.6. La parte par de la fracción continua (2.1) es la fracción continua

$$\hat{b}_0 + \sum_{i=1}^{\infty} \frac{\hat{a}_i}{\hat{b}_i} \quad (2.13)$$

con

$$\begin{aligned} \hat{b}_0 &= b_0, \quad \hat{a}_1 = a_1 b_2, \quad \hat{b}_1 = b_1 b_2 + a_2 \\ \hat{a}_2 &= -a_2 a_3 b_4, \quad \hat{b}_2 = (b_2 b_3 + a_3) b_4 + b_2 a_4 \end{aligned} \quad (2.14)$$

$$\hat{a}_i = -a_{2i-2} a_{2i-1} b_{2i-4} b_{2i}, \quad i \geq 3$$

$$\hat{b}_i = b_{2i} (b_{2i-2} b_{2i-1} + a_{2i-1}) + b_{2i-2} a_{2i}, \quad i \geq 3$$

Demostración. Considere la fracción continua (2.1) con convergentes

$$\left\{ \frac{P_i}{Q_i} \right\}_{i=0}^{\infty}$$

Tomando en cuenta que las convergentes de la parte par son:

$$\left\{ \frac{P_{2i}}{Q_{2i}} \right\}_{i=0}^{\infty}$$

y utilizando las fórmulas (2.11), se concluye que los numeradores y denominadores de la parte par son:

$$\hat{b}_0 = \frac{P_0}{Q_0}, \quad \hat{a}_1 = \frac{P_2}{Q_2} - \frac{P_0}{Q_0}, \quad \hat{b}_1 = 1$$

$$\hat{a}_i = \frac{\frac{P_{2(i-1)}}{Q_{2(i-1)}} - \frac{P_{2i}}{Q_{2i}}}{\frac{P_{2(i-1)}}{Q_{2(i-1)}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}}}, \quad \hat{b}_i = \frac{\frac{P_{2i}}{Q_{2i}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}}}{\frac{P_{2(i-1)}}{Q_{2(i-1)}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}}}$$

La fracción continua así construida se puede simplificar utilizando la transformación de equivalencia $(p_i)_{i=1}^{\infty}$, donde $p_i = Q_{2i} / Q_{2(i-1)}$.

Las fórmulas para \hat{a}_i se pueden escribir entonces como:

$$\hat{b}_0 = \frac{P_0}{Q_0} \quad , \quad \hat{a}_1 = \left[\frac{P_2}{Q_2} - \frac{P_0}{Q_0} \right] \left[\frac{Q_2}{Q_0} \right] \quad , \quad \hat{b}_1 = \frac{Q_2}{Q_0}$$

$$\hat{a}_i = \frac{\left[\frac{P_{2(i-1)}}{Q_{2(i-1)}} - \frac{P_{2i}}{Q_{2i}} \right] \left[\frac{Q_{2i}}{Q_{2(i-1)}} \right] \left[\frac{Q_{2(i-1)}}{Q_{2(i-2)}} \right]}{\frac{P_{2(i-1)}}{Q_{2(i-1)}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}}}$$

$$= \frac{Q_{2i} Q_{2(i-1)} \left[\frac{P_{2i}}{Q_{2i}} - \frac{P_{2(i-1)}}{Q_{2(i-1)}} \right]}{Q_{2(i-2)} Q_{2(i-1)} \left[\frac{P_{2(i-2)}}{Q_{2(i-2)}} - \frac{P_{2(i-1)}}{Q_{2(i-1)}} \right]}$$

mientras que para \hat{b}_i se obtiene:

$$\hat{b}_i = \frac{\left[\frac{P_{2i}}{Q_{2i}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}} \right] \left[\frac{Q_{2i}}{Q_{2(i-1)}} \right]}{\frac{P_{2(i-1)}}{Q_{2(i-1)}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}}}$$

$$= \frac{Q_{2i} Q_{2(i-2)} \left[\frac{P_{2i}}{Q_{2i}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}} \right]}{Q_{2(i-2)} Q_{2(i-1)} \left[\frac{P_{2(i-1)}}{Q_{2(i-1)}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}} \right]}$$

Utilizando ahora las ecuaciones de las proposiciones (2.3) y (2.4), podemos escribir las siguientes fórmulas:

$$\frac{P_{2i}}{Q_{2i}} - \frac{P_{2(i-1)}}{Q_{2(i-1)}} = (-1)^{2i} \frac{a_1 a_2 \cdots a_{2i-1} b_{2i}}{Q_{2i} Q_{2(i-1)}}$$

$$\frac{P_{2(i-1)}}{Q_{2(i-1)}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}} = (-1)^{2i-2} \frac{a_1 a_2 \cdots a_{2i-3} b_{2i-2}}{Q_{2(i-2)} Q_{2(i-1)}}$$

$$\begin{aligned} \frac{P_{2i}}{Q_{2i}} - \frac{P_{2(i-2)}}{Q_{2(i-2)}} &= \\ &= (-1)^{2i-2} \frac{a_1 \cdots a_{2i-3} b_{2i} [b_{2i-2} (b_{2i-1} b_{2i} + a_{2i}) + b_{2i} a_{2i-1}]}{Q_{2i} Q_{2(i-2)}} \end{aligned}$$

Estas al ser utilizadas en las ecuaciones para \hat{a}_i y \hat{b}_i , permiten simplificar las ecuaciones y obtener

$$\hat{a}_i = - a_{2i-2} a_{2i-1} b_{2i} / b_{2i-2} \quad y$$

$$\hat{b}_i = \frac{b_{2i-2}(b_{2i-1}b_{2i} + a_{2i}) + b_{2i}a_{2i-1}}{b_{2i-2}} .$$

Utilizando de nuevo una transformación de equivalencia con $p_i = b_{2i-2}$, se obtienen las fórmulas (2.14). ■

Proposición 2.7. La parte impar de la fracción continua (2.1) es una fracción continua de la forma (2.13), donde

$$\hat{b}_0 = \frac{b_0 b_1 + a_1}{b_1}, \quad \hat{a}_1 = \frac{a_1 a_2 b_3}{b_1}$$

$$\hat{b}_1 = (b_1 b_2 + a_2) b_3 + b_1 a_3 \quad (2.15)$$

$$\hat{a}_i = -a_{2i} a_{2i-1} b_{2i-3} b_{2i+1}, \quad i \geq 2$$

$$\hat{b}_i = (b_{2i} b_{2i+1} + a_{2i+1}) b_{2i-1} + a_{2i} b_{2i+1}, \quad i \geq 2$$

Demostración. Sigue los mismos pasos de la proposición (2.6). ■

2.7. Algoritmo de Viscovatov.

Otro procedimiento importante que nos permite

representar una función, que es cociente de dos funciones analíticas, como una fracción continua es el algoritmo de Viscovatov (1810) [10].

Sea f una función del tipo

$$f(x) = \frac{d_{10} + d_{11}x + d_{12}x^2 + \dots}{d_{00} + d_{01}x + d_{02}x^2 + \dots} \quad (2.16)$$

Para representar a f a través de una fracción continua hacemos los siguientes pasos. Escribimos a f en la forma

$$\begin{aligned} f(x) &= \frac{1}{\frac{d_{00}}{d_{10}} + \frac{d_{00} + d_{01}x + d_{02}x^2 + \dots}{d_{10} + d_{11}x + d_{12}x^2 + \dots} - \frac{d_{00}}{d_{10}}} \\ &= \frac{d_{10}}{d_{00} + \frac{(d_{01}d_{10} - d_{00}d_{11})x + (d_{02}d_{10} - d_{00}d_{12})x^2 + \dots}{d_{10} + d_{11}x + d_{12}x^2 + \dots}} \end{aligned}$$

$$= \frac{d_{10}}{d_{00} + x \frac{d_{20} + d_{21}x + d_{22}x^2 + \dots}{d_{10} + d_{11}x + d_{12}x^2 + \dots}}$$

donde $d_{2i} = d_{10} d_{0,i+1} - d_{00} d_{1,i+1}$, $i \geq 0$.

El procedimiento que se hizo se puede repetir al cociente

$$\frac{d_{20} + d_{21}x + d_{22}x^2 + \dots}{d_{10} + d_{11}x + d_{12}x^2 + \dots}$$

para obtener el siguiente desarrollo

$$f(x) = \frac{d_{10}}{d_{00} + \frac{d_{20}x}{d_{10} + x \frac{d_{30} + d_{31}x + d_{32}x^2 + \dots}{d_{20} + d_{21}x + d_{22}x^2 + \dots}}}$$

donde $d_{3i} = d_{20} d_{1,i+1} - d_{10} d_{2,i+1}$, $i \geq 0$.

Se puede continuar este procedimiento indefinidamente para obtener la fracción continua

$$f_0(x) = \frac{d_{10}}{d_{00}} + \frac{d_{20}x}{d_{10}} + \frac{d_{30}x^2}{d_{20}} + \dots \quad (2.17)$$

donde $d_{k,i} = d_{k-1,0} d_{k-2,i+1} - d_{k-2,0} d_{k-1,i+1}$, $k \geq 2$ y $i \geq 0$

Un problema importante que aparece en el método de Viscovatov es el de analizar la relación entre (2.17) y la función f . En este sentido se presentan una definición y dos proposiciones.

Definición 2.8. La fracción continua (2.1) para la cual el desarrollo de Taylor de su n -ésima convergente $C_n(x)$, alrededor del origen, coincide con la serie $\sum c_i x^i$ hasta el grado n , es llamada correspondiente con la serie. ■

Obsérvese que una fracción continua correspondiente es, de alguna manera, una aproximante de la serie, utilizando como criterio de aproximación la coincidencia de los desarrollos en series de Taylor. Este criterio de aproximación es el que vamos a utilizar más adelante para definir las aproximaciones del tipo Padé.

Proposición 2.8. Sea f_0 dada por la ecuación (2.16)

y definamos

$$f_1 = d_{10} - d_{00} f_0$$

$$f_k = d_{k,0} x f_{k-2} - d_{k-1,0} f_{k-1}$$

para $k = 2, 3, \dots$. Entonces

$$f_k = x^k \left[\frac{d_{k+1,0} + d_{k+1,1} x + d_{k+1,2} x^2 + \dots}{d_{00} + d_{01} x + d_{02} x^2 + \dots} \right]$$

Demostración. La prueba se hace por inducción en k .

Para $k = 1$ la demostración es como sigue :

$$\begin{aligned} f_1 &= d_{10} - \frac{d_{10}d_{00} + d_{11}d_{00}x + d_{12}d_{00}x^2 + \dots}{d_{00} + d_{01}x + d_{02}x^2 + \dots} \\ &= x \left[\frac{d_{20} + d_{21}x + d_{22}x^2 + \dots}{d_{00} + d_{01}x + d_{02}x^2 + \dots} \right] \end{aligned}$$

donde se ha utilizado la definición de $d_{2,i}$ para simplificar.

Se supone ahora que el resultado es válido para los valores menores a k , y se demostrará para k . En este caso se

utiliza la definicion de los números $d_{k,i}$. Como

$$\begin{aligned}
 f_k &= d_{k,0} x f_{k-2} - d_{k-1,0} f_{k-1} = \\
 &= d_{k,0} x^{k-1} \left[\frac{d_{k-1,0} + d_{k-1,1} x + d_{k-1,2} x^2 + \dots}{d_{00} + d_{01} x + d_{02} x^2 + \dots} \right] - \\
 &- d_{k-1,0} x^{k-1} \left[\frac{d_{k,0} + d_{k,1} x + d_{k,2} x^2 + \dots}{d_{00} + d_{01} x + d_{02} x^2 + \dots} \right]
 \end{aligned}$$

efectuando la resta y simplificando se obtiene

$$\begin{aligned}
 &= x^k \left[\frac{(d_{k,0} d_{k-1,1} - d_{k-1,0} d_{k,1}) + (d_{k,0} d_{k-1,2} - d_{k-1,0} d_{k,2}) x + \dots}{d_{00} + d_{01} x + d_{02} x^2 + \dots} \right] \\
 &= x^k \left[\frac{d_{k+1,0} + d_{k+1,1} x + d_{k+1,2} x^2 + \dots}{d_{00} + d_{01} x + d_{02} x^2 + \dots} \right] \quad \blacksquare
 \end{aligned}$$

Proposición 2.9. La k -ésima convergente de (2.17), la cual denotaremos por P_k / Q_k , satisface la siguiente ecuación :

$$f_0 - \frac{P_k}{Q_k} = (-1)^k \frac{f_k}{Q_k} \quad (2.18)$$

donde f_k es como en la proposición (2.8).

Demostración. La prueba es por inducción en k . Para $k = 1$ la prueba es como sigue. Como $f_1 = d_{10} - d_{00} f_0$ entonces

$$f_0 - \frac{P_1}{Q_1} = f_0 - \frac{d_{10}}{d_{00}} = \frac{f_0 d_{00} - d_{10}}{d_{00}} = (-1) \frac{f_1}{Q_1}$$

Se supone ahora que el resultado es válido para los valores menores a k , y se demostrará que es válido para k .

$$\begin{aligned} f_0 - \frac{P_k}{Q_k} &= \frac{f_0 Q_k - P_k}{Q_k} = \\ &= \frac{f_0 (d_{k-1,0} Q_{k-1} + d_{k,0}^x Q_{k-2}) - (d_{k-1,0} P_{k-1} + d_{k,0}^x P_{k-2})}{Q_k} \\ &= \frac{d_{k-1,0} (f_0 Q_{k-1} - P_{k-1}) + d_{k,0}^x (f_0 Q_{k-2} - P_{k-2})}{Q_k} \end{aligned}$$

$$= \frac{d_{k-1,0} Q_{k-1} \left[f_0 - \frac{P_{k-1}}{Q_{k-1}} \right] + d_{k,0} x Q_{k-2} \left[f_0 - \frac{P_{k-2}}{Q_{k-2}} \right]}{Q_k}$$

utilizando la hipótesis de inducción, se puede escribir la ecuación anterior como :

$$= \frac{d_{k-1,0} Q_{k-1} (-1)^{k-1} \frac{f_{k-1}}{Q_{k-1}} + d_{k,0} x Q_{k-2} (-1)^{k-2} \frac{f_{k-2}}{Q_{k-2}}}{Q_k}$$

$$= (-1)^{k-2} \frac{f_k}{Q_k} = (-1)^k \frac{f_k}{Q_k} \quad \blacksquare$$

Si se aplica la proposición (2.8) en la ecuación (2.18), se llega a la conclusión que la función f_0 y la fracción continua (2.17) son casi correspondientes. Pues la ecuación (2.18) nos dice que la función f_0 y la k -ésima convergente de (2.17) coinciden hasta el grado $k-1$.

El siguiente procedimiento nos permite construir una fracción continua correspondiente con una serie. Sea f una serie

$$f(x) = c_0 + c_1 x + c_2 x^2 + \dots \quad (2.19)$$

Aplicando el método de Viscovatov al cociente

$$\frac{f(x) - c_0}{x} = c_1 + c_2 x + c_3 x^2 + \dots \quad (2.20)$$

se obtiene la fracción continua

$$\left| \frac{c_1}{1} \right| + \left| \frac{d_{20} x}{c_1} \right| + \left| \frac{d_{30} x}{d_{20}} \right| + \dots,$$

para la cual ya sabemos que el desarrollo en serie de Taylor de la k -ésima convergente coincide con la serie (2.20) hasta el grado $k - 1$. Es decir, si

$$\left| \frac{c_1}{1} \right| + \left| \frac{d_{20} x}{c_1} \right| + \left| \frac{d_{30} x}{d_{20}} \right| + \dots = \sum_{i=0}^{\infty} e_i x^i$$

entonces

$$e_0 = c_1, \quad e_1 = c_2, \quad \dots, \quad e_{k-1} = c_k$$

Ahora es claro que la fracción continua

$$c_0 + \left| \frac{c_1 x}{1} \right| + \left| \frac{d_{20} x}{c_1} \right| + \left| \frac{d_{30} x}{d_{20}} \right| + \dots = c_0 + \sum_{i=0}^{\infty} e_i x^{i+1}$$

coincide con la serie (2.19) hasta el grado k , teniendo así una fracción continua correspondiente para la serie (2.19).

2.8. Evaluación de convergentes.

En esta sección se estudiarán algunas técnicas para evaluar las convergentes de una fracción continua. Esto nos va hacer útil en los capítulos tres y cuatro, cuando estudiemos la representación de Rutishauser de las aproximantes de Padé a través de fracciones continuas.

El primer algoritmo para evaluar una convergente es el de *sustitución hacia adelante*. Si ponemos

$$P_{-1} = 1, \quad P_0 = b_0, \quad Q_{-1} = 0 \quad \text{y} \quad Q_0 = 1$$

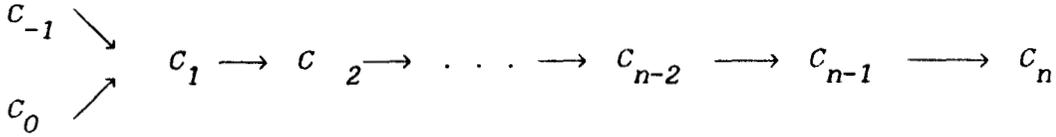
entonces se pueden calcular P_n y Q_n para $n \geq 1$, por medio de las ecuaciones (2.5)

$$P_n = b_n P_{n-1} + a_n P_{n-2}$$

$$Q_n = b_n Q_{n-1} + a_n Q_{n-2}$$

Así $C_n = P_n / Q_n$. Este algoritmo es útil porque nos permite

obtener las convergentes anteriores a la C_n , a partir de C_{-1} y C_0



El siguiente algoritmo se puede llamar de *sustitución hacia atrás*. Este consiste en lo siguiente, si se quiere evaluar la n -ésima convergente de (2.1), definimos la sucesión $(r_{i,n})_{i=1}^{n+1}$ con

$$\begin{aligned}
 r_{n+1,n} &= 0 \\
 r_{i,n} &= \frac{a_i}{b_i + r_{i+1,n}}, \quad i = n, \dots, 1 \quad (2.21)
 \end{aligned}$$

Entonces $C_n = b_0 + r_{1,n}$.

Una deficiencia de este método es que sirve para calcular la n -ésima convergente y solamente esa. Este procedimiento es el que se utiliza en el paquete PADE para evaluar convergentes.

Los resultados que siguen nos permiten expresar la n-ésima convergente a partir de un sistema tridiagonal de ecuaciones lineales.

Proposición 2.10. El n-ésimo numerador y denominador de la fracción continua (2.1) se pueden escribir como:

$$P_n = \begin{vmatrix} b_0 & -1 & 0 & \dots & \dots & 0 \\ a_1 & b_1 & -1 & \dots & \dots & \cdot \\ 0 & a_2 & b_2 & \dots & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \dots & -1 \\ 0 & \dots & \dots & 0 & a_n & b_n \end{vmatrix} \quad Q_n = \begin{vmatrix} b_1 & -1 & 0 & \dots & \dots & 0 \\ a_2 & b_2 & -1 & \dots & \dots & \cdot \\ 0 & a_3 & b_3 & \dots & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \dots & -1 \\ 0 & \dots & \dots & 0 & a_n & b_n \end{vmatrix}$$

Demostración. La demostración se hace observando que los determinantes satisfacen las relaciones de recurrencia (2.5)

$$P_{n+1} = b_{n+1}P_n + a_{n+1}P_{n-1} \quad y$$

$$Q_{n+1} = b_{n+1}Q_n + a_{n+1}Q_{n-1}$$

Comprobaremos únicamente la fórmula para P_{n+1} , pues para Q_{n+1} es similar. Sea

$$P_{n+1} = \begin{vmatrix} b_0 & -1 & 0 & \dots & 0 \\ a_1 & b_1 & -1 & \dots & \vdots \\ 0 & a_2 & b_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & -1 \\ 0 & \dots & 0 & a_{n+1} & b_{n+1} \end{vmatrix}$$

Si desarrollamos este determinante a lo largo del último renglón obtenemos

$$b_{n+1} \begin{vmatrix} b_0 & -1 & 0 & \dots & 0 \\ a_1 & b_1 & -1 & \dots & \vdots \\ 0 & a_2 & b_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & -1 \\ 0 & \dots & 0 & a_n & b_n \end{vmatrix} - a_{n+1} \begin{vmatrix} b_0 & -1 & 0 & \dots & 0 \\ a_1 & b_1 & -1 & \dots & \vdots \\ 0 & a_2 & b_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & -1 \\ 0 & \dots & 0 & a_n & -1 \end{vmatrix}$$

Si ahora calculamos el segundo determinante a lo largo del último renglón, obtenemos la fórmula que queríamos demostrar. ■

Proposición 2.11. La n -ésima convergente de la fracción continua (2.1), con $b_0 = 0$, es la incognita α_{1n} del siguiente sistema tridiagonal

$$\begin{bmatrix} b_1 & -1 & 0 & \dots & 0 \\ a_2 & b_2 & -1 & & \cdot \\ 0 & a_3 & b_3 & \cdot & 0 \\ \cdot & & \cdot & \cdot & \cdot \\ \cdot & & \cdot & & -1 \\ 0 & \cdot & \cdot & a_n & b_n \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \\ x_{3n} \\ \cdot \\ \cdot \\ x_{nn} \end{bmatrix} = \begin{bmatrix} a_1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad (2.22)$$

Demostración. Para evitar la escritura complicada de los determinantes, la demostración se puede explicar como sigue: Se aplica la regla de Cramer para determinar la incógnita x_{1n} y se desarrolla el determinante del numerador por cofactores a lo largo del primer renglón. Se utilizan también las fórmulas de la proposición (2.10). ■

Otros métodos para evaluar fracciones continuas se obtienen de triangularizar la matriz de coeficientes del sistema (2.22), y utilizar luego una sustitución hacia adelante o atrás para resolver el sistema de ecuaciones (2.22). Por ejemplo, triangularizando inferiormente se obtienen las fórmulas (2.21) del algoritmo de sustitución hacia atrás. Por otro lado triangularizando superiormente se obtiene el siguiente algoritmo

$$x_{1n} = \sum_{i=1}^n (-1)^{i-1} \frac{a_1 a_2 \dots a_i}{\rho_{1n}^2 \dots \rho_{i-1,n}^2 \rho_{i,n}} \quad (2.23)$$

donde

$$\rho_{1n} = b_1 \quad \text{y} \quad \rho_{in} = b_i + \frac{a_i}{\rho_{i-1,n}}, \quad i=2, \dots, n$$

El resultado es así, pues al triangularizar superiormente la matriz de coeficientes del sistema (2.22), ella toma la siguiente forma

$$\begin{bmatrix} \rho_1 & -1 & 0 & \dots & 0 \\ 0 & \rho_2 & -1 & & \cdot \\ 0 & 0 & \rho_3 & & 0 \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & -1 \\ 0 & \dots & 0 & 0 & \rho_n \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \\ x_{3n} \\ \cdot \\ \cdot \\ x_{nn} \end{bmatrix} = \begin{bmatrix} a_1 \\ -\frac{a_1 a_2}{\rho_1} \\ \frac{a_1 a_2 a_3}{\rho_1 \rho_2} \\ \cdot \\ \cdot \\ (-1)^{n-1} \frac{a_1 \dots a_n}{\rho_1 \dots \rho_{n-1}} \end{bmatrix}$$

El último algoritmo que se mencionará se obtiene cuando en el sistema (2.22) se hacen las siguientes sustituciones sucesivas: la incógnita x_{2n} de la primera ecuación se sustituye en la segunda, quedando ésta en función de x_{1n} y x_{3n} . La incógnita x_{3n} de la nueva ecuación dos se sustituye en la tercera,

quedando ésta en función de x_{1n} y x_{4n} . Si se continúa este procedimiento hasta llegar a la última ecuación se obtendrá una ecuación lineal ϕ que depende de x_{1n} únicamente. Ahora recordando que x_{1n} es la n -ésima convergente, resulta que la n -ésima convergente es la raíz de ϕ .

Para calcular la raíz de ϕ , basta con tener dos puntos de la recta. Estos puntos los obtenemos evaluando la recta en $x_{1n}^{(0)} = 0$ y $x_{1n}^{(1)} = 1$. La evaluación para $x_{1n}^{(0)}$ es como sigue

$$\begin{aligned} x_{2n}^{(0)} &= -a_1 \\ x_{3n}^{(0)} &= - (a_2 x_{1n}^{(0)} + b_2 x_{2n}^{(0)}) \\ &\vdots \\ x_{n-1,n}^{(0)} &= - (a_{n-2} x_{n-3,n}^{(0)} + b_{n-2} x_{n-2,n}^{(0)}) \\ x_{nn}^{(0)} &= - (a_{n-1} x_{n-2,n}^{(0)} + b_{n-1} x_{n-1,n}^{(0)}) \end{aligned}$$

Sustituyendo finalmente estos valores en la última ecuación se obtiene

$$\phi(x_{1n}^{(0)}) = b_n x_{nn}^{(0)} + a_n x_{n-1,n}^{(0)} \quad (2.24)$$

Este procedimiento se puede repetir para $x_{1n}^{(1)}$ para

para llegar al siguiente resultado

$$\phi(x_{1n}^{(1)}) = b_n x_{nn}^{(1)} + a_n x_{n-1,n}^{(1)} \quad (2.25)$$

Teniendo estos dos puntos de ϕ , se puede encontrar la ecuación de la recta como

$$\phi(x_{1n}) - \phi(x_{1n}^{(0)}) = \frac{\phi(x_{1n}^{(0)}) - \phi(x_{1n}^{(1)})}{x_{1n}^{(0)} - x_{1n}^{(1)}} \left[x_{1n} - x_{1n}^{(0)} \right],$$

la cual se simplifica a

$$x_{1n} = \frac{\phi(x_{1n}^{(0)})}{\phi(x_{1n}^{(0)}) - \phi(x_{1n}^{(1)})}$$

Tomando en cuenta los valores calculados por las ecuaciones (2.24) y (2.25) se obtiene en definitiva

$$x_{1n} = \frac{b_n x_{nn}^{(0)} + a_n x_{n-1,n}^{(0)}}{b_n (x_{nn}^{(0)} - x_{nn}^{(1)}) + a_n (x_{n-1,n}^{(0)} - x_{n-1,n}^{(1)})}$$

para llegar al siguiente resultado

$$\phi(x_{1n}^{(1)}) = b_n x_{nn}^{(1)} + a_n x_{n-1,n}^{(1)} \quad (2.25)$$

Teniendo estos dos puntos de ϕ , se puede encontrar la ecuación de la recta como

$$\phi(x_{1n}) - \phi(x_{1n}^{(0)}) = \frac{\phi(x_{1n}^{(0)}) - \phi(x_{1n}^{(1)})}{x_{1n}^{(0)} - x_{1n}^{(1)}} \left[x_{1n} - x_{1n}^{(0)} \right],$$

la cual se simplifica a

$$x_{1n} = \frac{\phi(x_{1n}^{(0)})}{\phi(x_{1n}^{(0)}) - \phi(x_{1n}^{(1)})}$$

Tomando en cuenta los valores calculados por las ecuaciones (2.24) y (2.25) se obtiene en definitiva

$$x_{1n} = \frac{b_n x_{nn}^{(0)} + a_n x_{n-1,n}^{(0)}}{b_n (x_{nn}^{(0)} - x_{nn}^{(1)}) + a_n (x_{n-1,n}^{(0)} - x_{n-1,n}^{(1)})}$$

CAPITULO III Aproximación de Padé.

3.1. Introducción

En este capítulo se considera el tema de aproximación de funciones a través de funciones racionales. Este tema se puede considerar como extensión del de aproximación polinomial de Taylor. Como se sabe, para una función f el polinomio de Taylor de grado n alrededor de $x = 0$, denotado por $T(x)$, se construye atendiendo al siguiente criterio de aproximación:

$$T^{(k)}(0) = f^{(k)}(0), \quad k = 0, \dots, n \quad (3.1)$$

Este criterio es el que utilizaremos para construir aproximaciones racionales locales a funciones analíticas.

3.2. Problema de aproximación de Padé.

Definición 3.1. Sea $f(x) = \sum c_i x^i$ una función analítica compleja. El problema de aproximación de Padé de orden

(m, n) consiste en encontrar dos polinomios p y q , tales que

$$\partial p \leq m, \quad \partial q \leq n \quad \text{y}$$

$$r_{m,n}^{(k)}(0) = f^{(k)}(0), \quad k = 0, \dots, m+n, \quad (3.2)$$

donde $r_{m,n}(x) = (p/q)(x)$ (∂p indica el grado de p). ■

Si f es una función analítica, denotaremos por $\omega(f)$ el grado de la potencia más pequeña en el desarrollo de f , cuyo coeficiente es diferente de cero. Llamamos a $\omega(f)$ el orden de f .

Proposición 3.1. El problema de aproximación (3.2) es equivalente a encontrar polinomios p y q , tales que

$$\partial p \leq m, \quad \partial q \leq n \quad \text{y} \quad \omega(fq - p) \geq m+n+1 \quad (3.3)$$

Demostración. Suponga que se cumple (3.2) y considere la función $e(x) = f(x) - r_{m,n}(x)$. Como $f^{(k)}(0) - r_{m,n}^{(k)}(0) = 0$ para $k = 0, 1, \dots, m+n$, entonces $x = 0$ es una raíz de multiplicidad $m+n+1$ para $fq - p$, y por lo tanto se satisface la última desigualdad en (3.3).

El recíproco se puede comprobar si $q(0) \neq 0$. Como

$$\frac{f(x)q(x) - p(x)}{q(x)} = \sum_{i=0}^{\infty} a_{m+n+1+i} x^{m+n+1+i} \quad (3.4)$$

entonces se satisface (3.2). ■

3.3. Algoritmo de Gauss.

También se puede observar, considerando el producto de series en $f q - p$, que el problema de aproximación (3.2) es equivalente a los dos sistemas de ecuaciones lineales

$$\begin{aligned} c_0 b_0 &= a_0 \\ c_1 b_0 + c_0 b_1 &= a_1 \\ \vdots & \\ c_m b_0 + c_{m-1} b_1 + \dots + c_{m-n} b_n &= a_m \end{aligned} \quad (3.5)$$

$$\begin{aligned} c_{m+1} b_0 + c_m b_1 + \dots + c_{m-n+1} b_n &= 0 \\ \vdots & \\ c_{m+n} b_0 + c_{m+n-1} b_1 + \dots + c_m b_n &= 0 \end{aligned} \quad (3.6)$$

donde $c_i = 0$ para $i < 0$.

Para resolver un problema de aproximación de Padé se encuentra primero una solución *no trivial* del sistema (3.6) (esto siempre se puede hacer), para luego, por sustitución directa en el sistema (3.5) calcular los coeficientes del polinomio q .

El procedimiento mediante el cual se obtienen los coeficientes de los polinomios p y q , del problema de aproximación (3.3) a través de la solución de los sistemas (3.5) y (3.6), fue programado en PADE. En las páginas 63, 64 y 65 presentamos una copia de este programa.

Las variables m y n denotan el orden de la aproximante, *Coef* es un vector que contiene los coeficientes del desarrollo en serie de la función, *Coef1* y *Coef2* son matrices que sirven para formar los coeficientes de los sistemas (3.5) y (3.6), *Gauss* es un procedimiento para resolver sistemas de ecuaciones, los vectores a y b contienen los coeficientes de los polinomios p y q respectivamente, *Eva* es un procedimiento para evaluar polinomios y *Dib* es una unidad encargada de dibujar en la pantalla de la computadora la aproximante.

```

Unit   Algo1;
Procedure  Aprx1;
Begin
  If (n[t]>1) then
    Begin
      For r:=1 to n[t] do
        Begin
          For s:=1 to n[t] do
            Coef1[r,s]:=Coef[m[t]+r-s];
          End;
        End;
      For r:=1 to n[t] do
        Coef2[r]:=-Coef[m[t]+r];
      End;
      b[0]:=1;
      Gauss(Coef2,Coef1,n[t],b);
      b[0]:=1.0;
      For r:=0 to m[t] do
        Begin
          Intermedio:=0;
          For s:=0 to r do
            Intermedio:=Coef[r-s]*b[s]+Intermedio;
          End;
          a[r]:=Intermedio;
        End;
      X1:=Xmin;
      Eva(m[t],a,X1,Evaluacion1);
      Eva(n[t],b,X1,Evaluacion2);
      If Abs(Evaluacion2)>MinimoReal then
        Y1:=Evaluacion1/Evaluacion2;
      For r:=1 to 200 do
        Begin
          X2:=Xmin+((Xmax-Xmin)*r)/200;
          Eva(m[t],a,X2,Evaluacion1);
          Eva(n[t],b,X2,Evaluacion2);
          If Abs(Evaluacion2)>MinimoReal then
            Y2:=Evaluacion1/Evaluacion2;
          Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
          X1:=X2;
          Y1:=Y2;
        End;
      End;
    End;
  End;

```

```

If (n[t]=1) then
Begin
  If Abs(Coef[m[t]])>MinimoReal then
  Begin
    b[0]:=1.0;b[1]:=-Coef[m[t]+1]*b[0]/Coef[m[t]];
    For r:=0 to m[t] do
      Begin
        If r=0 then
          a[0]:=Coef[0]*b[0];
        If r<>0 then
          a[r]:=Coef[r]+b[1]*Coef[r-1];
        End;
      End;
    End;
  If Abs(Coef[m[t]])<MinimoReal then
  Begin
    b[0]:=0;
    b[1]:=1;
    For r:=0 to m[t] do
      Begin
        If r=0 then
          a[0]:=0;
        If r>0 then
          a[r]:=Coef[r-1];
        End;
      End;
    End;
  X1:=Xmin;
  Eva(m[t],a,X1,Evaluacion1);
  Eva(n[t],b,X1,Evaluacion2);
  If Abs(Evaluacion2)>MinimoReal then
    Y1:=Evaluacion1/Evaluacion2;
  For r:=1 to 200 do
    Begin
      X2:=Xmin+((Xmax-Xmin)*r)/200;
      Eva(m[t],a,X2,Evaluacion1);
      Eva(n[t],b,X2,Evaluacion2);
      If Abs(Evaluacion2)>MinimoReal then
        Y2:=Evaluacion1/Evaluacion2;
      Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
      X1:=X2;
      Y1:=Y2;
    End;
  End;
End;

```

```

If (n[t]=0) then
  Begin
    X1:=Xmin;
    Eva(m[t],Coef,X1,Evaluacion1);
    Y1:=Evaluacion1;
    For r:=1 to 200 do
      Begin
        X2:=Xmin+(Xmax-Xmin)*r/200;
        Eva(m[t],Coef,X2,Evaluacion2);
        Y2:=Evaluacion2;
        Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
        X1:=X2;Y1:=Y2;
      End;
    End;
  End;
End.

```

Ejemplo 3.1. Ahora daremos un ejemplo en que se muestren las aproximaciones generadas por una aproximación de Padé. Para tal efecto consideremos la función de Bessel de orden 0 definida por

$$J_0(x) = \sum_{k=0}^{\infty} \frac{(-x^2/4)^k}{k! k!} \quad (3.7)$$

Para esta función tenemos los coeficientes que querramos para construir cualquier aproximante racional de Padé. En la tabla (3.1) se muestran las aproximaciones proporcionados por la aproximante racional de orden (5,5). ■

Una desventaja del algoritmo de Gauss es que el trabajo

tabla 3.1.

Aproximaciones a la función de Bessel de orden 0
a través de una aproximante de Padé de orden (5,5).

x_i	$r_{5,5}(x_i)$	Valor correcto*
0.0000000000E+00	1.0000000000E+00	1
1.0000000000E-01	9.9750156207E-01	0.99750 15620
2.0000000000E-01	9.9002497224E-01	0.99002 49722
3.0000000000E-01	9.7762624654E-01	0.97762 62465
4.0000000000E-01	9.6039822666E-01	0.96039 82266
5.0000000000E-01	9.3846980724E-01	0.93846 98072
6.0000000000E-01	9.1200486350E-01	0.91200 48634
7.0000000000E-01	8.8120088861E-01	0.88120 08886
8.0000000000E-01	8.4628735275E-01	0.84628 73527
9.0000000000E-01	8.0752379813E-01	0.80752 37981
1.0000000000E+00	7.6519768657E-01	0.76519 76865
1.1000000000E+00	7.1962201855E-01	0.71962 20185
1.2000000000E+00	6.7113274430E-01	0.67113 27442
1.3000000000E+00	6.2008598962E-01	0.62008 59895
1.4000000000E+00	5.6685512047E-01	0.56685 51203
1.5000000000E+00	5.1182767187E-01	0.51182 76717
1.6000000000E+00	4.5540216784E-01	0.45540 21676
1.7000000000E+00	3.9798485974E-01	0.39798 48594
1.8000000000E+00	3.3998641145E-01	0.33998 64110
1.9000000000E+00	2.8181855994E-01	0.28181 85593
2.0000000000E+00	2.2389077991E-01	0.22389 07791
2.1000000000E+00	1.6660698136E-01	0.16660 69803
2.2000000000E+00	1.1036226828E-01	0.11036 22669
2.3000000000E+00	5.5539786212E-02	0.05553 97884
2.4000000000E+00	2.5076855748E-03	0.00250 76832
2.5000000000E+00	-4.8383773565E-02	-0.04838 37764
2.6000000000E+00	-9.6804950728E-02	-0.09680 49543
2.7000000000E+00	-1.4244936545E-01	-0.14244 93700
2.8000000000E+00	-1.8503602766E-01	-0.18503 60333
2.9000000000E+00	-2.2431153877E-01	-0.22431 15457
3.0000000000E+00	-2.6005194631E-01	-0.26005 19549

* Tomados de [1]

computacional realizado para obtener una aproximante no sirve para obtener otra. Esto es un indicador de que el algoritmo de Gauss no es eficiente. Los algoritmos cociente-diferencia y epsilon, que estudiaremos más adelante, remedian ese problema.

Ejemplo 3.2. Utilizando aproximaciones racionales se pueden construir algoritmos que aproximen la solución $y(x)$ de una ecuación diferencial. Consideremos el problema de valor inicial

$$\frac{d y}{d x} = f(x, y) , \quad \text{con } y(a) = y_0 \quad (3.8)$$

Una forma de resolver (3.8) numéricamente, consiste en encontrar aproximaciones y_i de $y(x_i)$ en varios puntos $x_i \in [a, b]$. Para construir las aproximaciones y_i , hacemos lo siguiente. Si $r_{m,n}$ es una aproximación de Padé para $y(x)$ en x_i entonces una aproximación a $y(x_{i+1})$ la obtenemos de la ecuación

$$y_{i+1} = r_{m,n}(x_{i+1}) \quad (3.9)$$

Para calcular $r_{m,n}$ es necesario contar con el desarrollo de Taylor de $y(x)$ en x_i , es decir

$$y(x) = y(x_i) + (x-x_i)f(x_i, y(x_i)) + \frac{(x-x_i)^2}{2!} f'(x_i, y(x_i)) + \dots$$

Como $y(x_i)$ no se conoce, pero si una aproximación, entonces en su lugar se utiliza ésta, para así tener el desarrollo

$$y(x) = y_i + (x-x_i)f(x_i, y_i) + \frac{(x-x_i)^2}{2!} f'(x_i, y_i) + \dots \quad (3.10)$$

Un ejemplo permitirá ver como se utiliza (3.9). La ecuación diferencial que se resolverá numéricamente es

$$\frac{d y}{d x} = 1 + y^2, \quad y(0) = 1 \quad (3.11)$$

Es muy fácil ver que la solución a este problema es

$$y(x) = \text{Tan} \left(x + \frac{\pi}{4} \right) \quad (3.12)$$

Una particularidad de esta función es que tiene una asíntota vertical en $\pi/4 = 0.7853981$. Se encontrarán aproximaciones de (3.12) en el intervalo $[0, 0.75]$, con un paso de $h = 0.05$ y empleando una aproximación de Padé de orden (1,1). Lo primero que se debe hacer es encontrar la aproximación de Padé:

$$r_{1,1}(x) = \frac{a_1(x - x_i) + a_0}{b_1(x - x_i) + b_0} \quad (3.13)$$

donde $x_i = i * (0.05)$ y $i = 0, 1, \dots, 14$. Para encontrar los coeficientes de (3.13) resolvemos los sistemas (3.5) y (3.6), estos al ser sustituidos en (3.13) y después de simplificar obtenemos

$$y_{i+1} = y_i + h \left[\frac{2 f^2(x_i, y_i)}{2 f(x_i, y_i) - h f'(x_i, y_i)} \right] \quad (3.14)$$

La fórmula (3.14) es similar a la del método de Taylor de orden dos. En la tabla (3.2) se muestran los resultados obtenidos por ambos métodos.

TABLA 3.2.

x_i	Aproximación de Padé $m = 1, n = 1$	Solución Exacta	Serie de Taylor $m = 2$
0.05	1.1052631579E+00	1.1053555905E+00	1.1050000000E+00
0.10	1.2228412256E+00	1.2230488804E+00	1.2221868316E+00
0.15	1.3557335707E+00	1.3560878511E+00	1.3544934066E+00
0.20	1.5079526738E+00	1.5084976471E+00	1.5058248209E+00
0.25	1.6849974979E+00	1.6857964172E+00	1.6815009786E+00
0.30	1.8946189035E+00	1.8957651228E+00	1.8889628892E+00
0.35	2.1481115401E+00	2.1497476402E+00	2.1389447390E+00

0.40	2.4626095385E+00	2.4649627567E+00	2.4475109635E+00
0.45	2.8654314837E+00	2.8688840280E+00	2.8397986097E+00
0.50	3.4029820837E+00	3.4082234423E+00	3.3573744918E+00
0.55	4.1609668690E+00	4.1693640459E+00	4.0739766086E+00
0.60	5.3172019204E+00	5.3318552234E+00	5.1336682017E+00
0.65	7.3108707091E+00	7.3404365750E+00	6.8514405269E+00
0.70	1.1601853114E+01	1.1681373800E+01	1.0069735860E+01
0.75	2.7748629007E+01	2.8238252850E+01	1.7767556690E+01

3.4. Fórmula para estimar el error.

Como se observa de la *tabla 3.1*, las aproximaciones obtenidas a partir de $r_{5,5}$ son bastante buenas (coinciden con los correctos hasta en seis cifras decimales). Sin embargo es importante que podamos medir nuestra precisión, sin tener que recurrir a una tabla de valores de la función, pues esto no es lo natural. Para este propósito podemos utilizar la ecuación

$$f(x) - r_{m,n}(x) = \left[\sum_{i=m+n+1}^{\infty} \gamma_i x^i \right] / q(x). \quad (3.15)$$

Los coeficientes γ_i son fáciles de calcular porque

$$\gamma_i = c_i b_0 + \dots + c_{i-n} b_n \quad (3.16)$$

Utilicemos la suma parcial de orden uno de (3.15) para estimar el error para el ejemplo (3.1). Es decir

$$J_0(x) - r_{5,5}(x) \approx \gamma_{11} (-x*x/4)^{11} / q(x) \quad (3.17)$$

Haciendo los cálculos en (3.17) obtenemos los errores mostrados en la tabla (3.3). Como se observa los errores estimados son bastante pequeños, lo cual no corresponde a los errores reales. Sin embargo estas aproximaciones al error nos dan una pista de que también $r_{5,5}$ aproxima a la función.

Podríamos obtener mejores estimaciones del error si tomáramos en cuenta sumas parciales con mayor número de sumandos. También podríamos utilizar algunas técnicas de aceleración de convergencia de sucesiones para acelerar la convergencia de las sumas parciales de (3.15).

tabla 3.3.

Estimación de los errores en la aproximación

$$J_0(x) \approx r_{5,5}(x)$$

x_i	$e(x_i)$
0.0000000000E+00	0.0000000000E+00
1.0000000000E-01	-3.5306182895E-27
2.0000000000E-01	-7.2790708361E-24
3.0000000000E-01	-6.3383330529E-22

4.0000000000E-01	-1.5107633744E-20
5.0000000000E-01	-1.7705243480E-19
6.0000000000E-01	-1.3243139123E-18
7.0000000000E-01	-7.2660933410E-18
8.0000000000E-01	-3.1776623128E-17
9.0000000000E-01	-1.1686402713E-16
1.0000000000E+00	-3.7489409598E-16
1.1000000000E+00	-1.0767713427E-15
1.2000000000E+00	-2.8228837828E-15
1.3000000000E+00	-6.8544920169E-15
1.4000000000E+00	-1.5591873910E-14
1.5000000000E+00	-3.3526564014E-14
1.6000000000E+00	-6.8643632905E-14
1.7000000000E+00	-1.3462130661E-13
1.8000000000E+00	-2.5413729241E-13
1.9000000000E+00	-4.6371974189E-13
2.0000000000E+00	-8.2071243068E-13
2.1000000000E+00	-1.4130851494E-12
2.2000000000E+00	-2.3730153791E-12
2.3000000000E+00	-3.8954010004E-12
2.4000000000E+00	-6.2627413957E-12
2.5000000000E+00	-9.8781516524E-12
2.6000000000E+00	-1.5308657961E-11
2.7000000000E+00	-2.3341368552E-11
2.8000000000E+00	-3.5055631001E-11
2.9000000000E+00	-5.1914881361E-11
3.0000000000E+00	-7.5882572005E-11

3.5. Propiedades básicas.

Se mostrará que dos soluciones (p_1, q_1) y (p_2, q_2) al problema de aproximación de Padé de orden (m, n) son equivalentes como funciones racionales.

Proposición 3.2. Si (p_1, q_1) y (p_2, q_2) son soluciones de (3.3) entonces $p_1 q_2 = p_2 q_1$.

Demostración. Considere el polinomio $\phi = p_1 q_2 - p_2 q_1$, que puede ser escrito como $(fq_2 - p_2)q_1 - (fq_1 - p_1)q_2$. Como $\omega(fq_1 - p_1) \geq m + n + 1$ y $\omega(fq_2 - p_2) \geq m + n + 1$ entonces $\omega(\phi) \geq m + n + 1$. Pero al ser $\delta(p_1 q_2 - p_2 q_1) \leq m + n$ se infiere que estas se cumplen sólo si $\phi \equiv 0$. ■

Definición 3.2. Si $p(x)$ y $q(x)$ satisfacen el problema de aproximación de Padé de orden (m, n) , para f analítica compleja, se denotará por

$$r_{m,n}(x) = \frac{p_0(x)}{q_0(x)} \quad (3.18)$$

la fracción irreducible única de p/q normalizada ($q_0(0) = 1$). La función racional (3.18) es llamada la aproximación de Padé de orden (m, n) para f . ■

Antes de continuar conviene hacer algunas observaciones acerca de la definición (3.2). Si $p(x)$ y $q(x)$ tienen factores en común entonces p_0 y q_0 no son de grados m y n respectivamente, se denotarán por m^* y n^* los grados de p_0 y q_0 respectivamente.

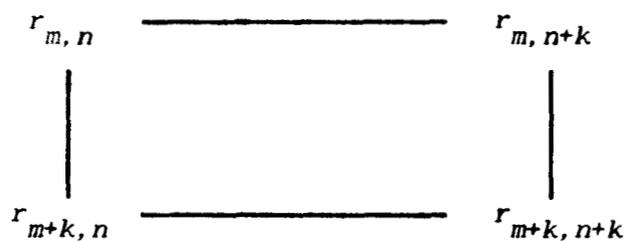
Otro matemático que contribuyó al desarrollo de las

aproximaciones racionales fue Charles Hermite (1822-1901). Hermite arregló en una tabla de doble entrada las aproximaciones y esto es lo que ahora se conoce como tabla de Padé

tabla de Padé.

$r_{0,0}$	$r_{0,1}$	$r_{0,2}$	·	·	·
$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	·	·	·
$r_{2,0}$	$r_{2,1}$	$r_{2,2}$	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

También estudió la llamada estructura de bloques en la tabla, o lo que ahora se conoce como *no-normalidad* de un elemento de la tabla. Este concepto significa que una aproximación racional $r_{m,n}$ sólo se puede repetir en bloques cuadrados, como en el siguiente diagrama.



3.6. Normalidad.

La propiedad de las aproximaciones racionales que se estudiará en esta sección es la normalidad. La importancia de la normalidad se debe primordialmente a que los algoritmos cociente-diferencia y epsilon para construir aproximantes de Padé, sólo se pueden usar si estas son normales.

Definición 3.3. Una aproximación racional es normal si ésta aparece sólo una vez en la tabla de Padé. ■

Existen funciones que muestran la existencia de la no-normalidad.

Ejemplo 3.3. La tabla de Padé para la función

$$f(x) = 1 + \text{Sen}(x) = 1 + x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

presenta repeticiones en los siguientes bloques :

$$\begin{array}{cc} r_{1,0} & r_{1,1} \\ r_{2,0} & r_{2,1} \end{array} \qquad \begin{array}{cc} r_{3,0} & r_{3,1} \\ r_{4,0} & r_{4,1} \end{array}$$

Proposición 3.3. Si la aproximación de Padé de orden (m, n) para f es $r_{m, n}(x) = \frac{p_0}{q_0}(x)$, entonces existe un entero ν con $0 \leq \nu \leq \min(m - m^*, n - n^*)$, tal que

$$p(x) = x^\nu p_0(x) \quad y \quad q(x) = x^\nu q_0(x)$$

satisfacen el problema de aproximación de Padé de orden (m, n) .

Demostración. Sea p_1 y q_1 una solución del problema de aproximación de Padé de orden (m, n) . Entonces

$$\partial p_1 \leq m, \quad \partial q_1 \leq n \quad y \quad \omega(fq_1 - p_1) \geq m + n + 1$$

Si p_0/q_0 es la fracción irreducible de p_1/q_1 entonces

$$p_1(x) = p_0(x) t(x) \quad y \quad q_1(x) = q_0(x) t(x) \quad (3.19)$$

El entero ν que se necesita es el orden de $t(x)$. Como el grado de $t(x)$ satisface $0 \leq \partial t \leq \min(m - m^*, n - n^*)$ entonces, obviamente para el orden de t también se cumple

$$0 \leq \omega(t) \leq \min(m - m^*, n - n^*)$$

Ahora se demostrará que $x^{\nu} p_0(x)$ y $x^{\nu} q_0(x)$ son solución del problema de aproximación de Padé de orden (m, n) . Claramente $\delta(x^{\nu} p_0(x)) \leq m$ y $\varepsilon(x^{\nu} q_0(x)) \leq n$. Ahora como

$$\begin{aligned} \omega(fq_1 - p_1) &= \omega(t(fq_0 - p_0)) \\ &= \omega(x^{\nu}(fq_0 - p_0)) = \omega(f(x^{\nu}q_0) - (x^{\nu}p_0)), \end{aligned}$$

entonces por ser (p_1, q_1) solución del problema de aproximación de Padé de orden (m, n) queda demostrada la proposición. ■

El siguiente resultado ayuda a encontrar repeticiones en la tabla de Padé.

Proposición 3.4. Si la aproximación de Padé de orden (m, n) para f es $r_{m, n} = p_0 / q_0$. Entonces

- a) $\omega(fq_0 - p_0) = m^* + n^* + t + 1$, con $t \geq 0$
- b) Para k y l que satisfagan $m^* \leq k \leq m^* + t$ y $n^* \leq l \leq n^* + t$, se cumple $r_{k, l} = r_{m, n}$.
- c) $m^* \leq m^* + t$ y $n^* \leq n^* + t$

Demostración. Para demostrar (a) se utiliza la

proposición (3.3), la cual indica que existe ν tal que

$$\omega (f q_0 - p_0) \geq m + n + 1 - \nu$$

donde $0 \leq \nu \leq \min (m - m^* , n - n^*)$. A partir de esto se deduce que $\omega (f q_0 - p_0) \geq m^* + n^* + 1$, pues $m \geq m^*$ y $-\nu \geq n^* - n$. De la antepenúltima desigualdad se deduce la existencia de un entero t no negativo que hace válida la igualdad. ■

Para demostrar (b) considere enteros k y l , tales que $m^* \leq k \leq m^* + t$ y $n^* \leq l \leq n^* + t$, y sean

$$\nu = \min (k - m^*) , \quad p(x) = x^\nu p_0(x) \quad \text{y} \quad q(x) = x^\nu q_0(x)$$

Entonces $\partial p \leq k$ y $\partial q \leq l$ y como consecuencia de (a)

$$\begin{aligned} \omega (fq - p) &= \omega (f x^\nu q_0 - x^\nu p_0) \\ &= \omega (x^\nu (f q_0 - p_0)) \\ &\geq m^* + n^* + t + \nu + 1 \end{aligned}$$

Por otro lado suponga que $\nu = k - m^*$, entonces

$$m^* + n^* + t + \nu + 1 \geq k + l + 1 , \quad (3.20)$$

pues $n^* + t \geq 1$. Si $\nu = 1 - n^*$ entonces también (3.20) es válida. Ahora la desigualdad $\omega(fq - p) \geq k + 1 + 1$ implica que p y q son solución del problema de aproximación de Padé de orden $(k, 1)$, pero también p y q son obviamente solución del problema de aproximación de Padé de orden (m, n) . ■

La demostración de (c) utiliza la proposición (3.4.a).

Si $p(x) = x^\nu p_0(x)$ y $q(x) = x^\nu q_0(x)$ entonces

$$m + n + 1 \leq \omega(fq - p) = m^* + n^* + t + \nu + 1$$

donde ν satisface $0 \leq \nu \leq \min(m - m^*, n - n^*)$. Como $\nu \leq m - m^*$ entonces $m + n + 1 \leq m + n^* + t + 1$ ■

El siguiente resultado da condiciones necesarias y suficientes para la normalidad

Proposición 3.5. La aproximación de Padé $r_{m,n} = p_0/q_0$ para f es normal si, y sólo si

- a) $m = m^*$ y $n = n^*$
- b) $\omega(fq_0 - p_0) = m + n + 1$

Demostración. Si $r_{m,n}$ es normal, (a) y (b) se demuestran por contraposición. Suponga que $m^* < m$ o $n^* < n$, utilizando la proposición (3.4.a) se tiene que :

$$\omega (fq_0 - p_0) = m^* + n^* + t + 1 , \quad t \geq 0 ,$$

entonces $\omega (fq_0 - p_0) \geq m^* + n^* + 1$. Esto implica que

$$r_{m^*, n^*} = \frac{p_0}{q_0} .$$

Pero debido a las suposiciones hechas en m^* y n^* se cumple $r_{m^*, n^*} \neq r_{m,n}$, contradiciendo la normalidad. Supóngase ahora que

$$\omega (fq_0 - p_0) \neq m + n + 1 ,$$

entonces por la proposición (3.4.a) se cumple

$$\omega (fq_0 - p_0) = m^* + n^* + t + 1$$

con $t > 0$, ya que si $t = 0$ entonces por la parte (a) de esta proposición $\omega (fq_0 - p_0) = m + n + 1$, lo cual no puede ser. De nuevo utilizando la parte (a) de esta proposición se tiene

$$\omega (fq_0 - p_0) = m + n + t + 1 , t > 0 ,$$

y utilizando la proposición (3.4.b) se deduce que $r_{k,l} = r_{m,n}$ para todos los enteros k y l que satisfagan $m^* \leq k \leq m^* + t$ y $n^* \leq l \leq n^* + t$.

Para probar que (a) y (b) garantizan la normalidad de $r_{m,n}$ se procede por reducción al absurdo. Suponga que $r_{m,n}$ no es normal, entonces $r_{m,n} = r_{k,l}$ para algunos enteros $k > m$ y $l > n$. Ahora utilizando la proposición (3.3), existe un entero ν con $0 \leq \nu \leq \min (k - m, l - n)$ tal que $p(x) = x^\nu p_0(x)$ y $q(x) = x^\nu q_0(x)$ satisfacen

$$\omega (x^\nu (fq_0 - p_0)) \geq k + l + 1 .$$

Por la hipótesis (b) se deduce que $m + n + \nu + 1 \geq k + l + 1$, por lo tanto $\nu \geq (k - m) + (l - n)$. En consecuencia $\nu \geq k - m$ o $\nu \geq l - n$, lo cual es una contradicción con las propiedades de ν . ■

Otras condiciones necesarias y suficientes para que una aproximación racional $r_{m,n}$ de una función analítica $\sum c_i x^i$ sea normal, se pueden dar en función de los siguientes determinantes:

$$D_{m, n+1} = \begin{bmatrix} c_n & c_{m-1} & \cdot & \cdot & \cdot & c_{m-n} \\ c_{n+1} & c_m & \cdot & \cdot & \cdot & c_{m-n+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{m+n} & c_{m+n-1} & \cdot & \cdot & \cdot & c_m \end{bmatrix}$$

con $D_{m,0} = 1$. La demostración del siguiente resultado se encuentra en [6].

Proposición 3.6. La aproximación de Padé $r_{m,n}$ para f es normal si, y sólo si los determinantes del siguiente diagrama son diferentes de cero

$$\begin{array}{ccc} D_{m,n} & \text{-----} & D_{m,n+1} \\ | & & | \\ D_{m+1,n} & \text{-----} & D_{m+1,n+1} \end{array} \quad \blacksquare$$

Si se aplica este resultado al ejemplo (3.3) y a la aproximación racional $r_{1,0}$, entonces

$$D_{1,0} = 1 \quad , \quad D_{2,0} = 1 \quad , \quad D_{1,1} = c_1 \quad \text{y} \quad D_{2,1} = c_2$$

y como $c_2 = 0$, se sigue la no-normalidad de $r_{1,0}$.

3.7. Convergencia

Una propiedad importante es la convergencia de sucesiones de aproximaciones racionales. Como las aproximaciones de Padé las utilizamos para aproximar funciones, es importante saber si la sucesión converge, en algún sentido, a la función. Como veremos los resultados que hay en este sentido no son fáciles de aplicar. La situación es similar al teorema de Weierstrass, aunque se sabe que existe un polinomio P tal que

$$| f(x) - P(x) | < \varepsilon \quad \text{para toda } x \text{ en } [a,b] ,$$

para f función continua en $[a,b]$. El resultado no nos indica como encontrar el polinomio P .

La utilización del paquete PADE, como una herramienta que nos ayuda a experimentar e intuir lo que sucede con la sucesión, puede ser valiosa.

Ejemplo 3.5. Utilizando los programas de Padé, nos podemos dar cuenta empíricamente que la sucesión de aproximantes

racionales $\Lambda = (r_{m,m})_{m=0}^{\infty}$ de la función $y = \ln(1+x)/x$ convergen a $f(x)$. En las tablas (3.4), (3.5) y (3.6) se muestran las estimaciones proporcionadas por algunos elementos de la sucesión.

Aunque la función es muy simple, es un ejemplo en el que se pone de manifiesto que las aproximantes de Padé convergen a f , inclusive cerca de una singularidad y en un intervalo bastante grande. ■

Los siguientes resultados muestran la convergencia, de sucesiones de aproximaciones racionales. El primer resultado es clásico.

Proposición 3.7. Si f es analítica en $B(0,r)$, el disco

Tabla 3.4.

Aproximaciones a $\ln(1+x)/x$ a través de $r_{2,2}$

x_i	$r_{2,2}(x_i)$	$f(x_i)$
-9.0000000000E-01	2.4355828220E+00	2.5584278811E+00
-8.0000000000E-01	1.9885057471E+00	2.0117973905E+00
-7.0000000000E-01	1.7144408252E+00	1.7199611490E+00
-6.0000000000E-01	1.5257731959E+00	1.5271512198E+00
-5.0000000000E-01	1.3859649123E+00	1.3862943611E+00
-4.0000000000E-01	1.2769953052E+00	1.2770640594E+00
-3.0000000000E-01	1.1889055472E+00	1.1889164798E+00
-2.0000000000E-01	1.1157167530E+00	1.1157177566E+00

-1.0000000000E-01	1.0536051340E+00	1.0536051566E+00
2.0000000000E-01	9.1160809372E-01	9.1160778397E-01
3.0000000000E-01	8.7454938717E-01	8.7454754823E-01
4.0000000000E-01	8.4118673647E-01	8.4118059155E-01
5.0000000000E-01	8.1094527363E-01	8.1093021622E-01
6.0000000000E-01	7.8336980306E-01	7.8333938208E-01
7.0000000000E-01	7.5809427948E-01	7.5804035866E-01
8.0000000000E-01	7.3482032218E-01	7.3473333113E-01
9.0000000000E-01	7.1330176496E-01	7.1317098464E-01
1.0000000000E+00	6.9333333333E-01	6.9314718056E-01
1.1000000000E+00	6.7474220400E-01	6.7448849521E-01
1.2000000000E+00	6.5738161559E-01	6.5704780030E-01
1.3000000000E+00	6.4112596456E-01	6.4069932533E-01
1.4000000000E+00	6.2586699306E-01	6.2533481240E-01
1.5000000000E+00	6.1151079136E-01	6.1086048792E-01
1.6000000000E+00	5.9797541576E-01	5.9719465314E-01
1.7000000000E+00	5.8518897704E-01	5.8426574883E-01
1.8000000000E+00	5.7308809292E-01	5.7201078732E-01
1.9000000000E+00	5.6161662464E-01	5.6037407210E-01

Tabla 3.5

Aproximación a $\ln(1+x)/x$ a través de $r_{3,3}$

x_i	$r_{3,3}(x_i)$	$f(x_i)$
-9.0000000000E-01	2.5242239926E+00	2.5584278811E+00
-8.0000000000E-01	2.0083493897E+00	2.0117973905E+00
-7.0000000000E-01	1.7194874631E+00	1.7199611490E+00
-6.0000000000E-01	1.5270814682E+00	1.5271512198E+00
-5.0000000000E-01	1.3862847222E+00	1.3862943611E+00
-4.0000000000E-01	1.2770629611E+00	1.2770640594E+00

-3.0000000000E-01	1.1889163944E+00	1.1889164798E+00
-2.0000000000E-01	1.1157177535E+00	1.1157177566E+00
-1.0000000000E-01	1.0536051566E+00	1.0536051566E+00
2.0000000000E-01	9.1160778460E-01	9.1160778397E-01
3.0000000000E-01	8.7454755591E-01	8.7454754823E-01
4.0000000000E-01	8.4118063361E-01	8.4118059155E-01
5.0000000000E-01	8.1093036530E-01	8.1093021622E-01
6.0000000000E-01	7.8333978515E-01	7.8333938208E-01
7.0000000000E-01	7.5804126554E-01	7.5804035866E-01
8.0000000000E-01	7.3473511882E-01	7.3473333113E-01
9.0000000000E-01	7.1317417567E-01	7.1317098464E-01
1.0000000000E+00	6.9315245477E-01	6.9314718056E-01
1.1000000000E+00	6.7449669527E-01	6.7448849521E-01
1.2000000000E+00	6.5705993176E-01	6.5704780030E-01
1.3000000000E+00	6.4071655225E-01	6.4069932533E-01
1.4000000000E+00	6.2535844951E-01	6.2533481240E-01
1.5000000000E+00	6.1089199028E-01	6.1086048792E-01
1.6000000000E+00	5.9723560407E-01	5.9719465314E-01
1.7000000000E+00	5.8431784683E-01	5.8426574883E-01
1.8000000000E+00	5.7207583253E-01	5.7201078732E-01
1.9000000000E+00	5.6045395265E-01	5.6037407210E-01

Tabla 3.6.

Aproximación a $\ln(1+x)/x$ a través de $r_{4,4}$

x_i	$r_{4,4}(x_i)$	$f(x_i)$
-9.0000000000E-01	2.5490234855E+00	2.5584278811E+00
-8.0000000000E-01	2.0112905452E+00	2.0117973905E+00
-7.0000000000E-01	1.7199206144E+00	1.7199611490E+00
-6.0000000000E-01	1.5271476888E+00	1.5271512198E+00

-5.0000000000E-01	1.3862940785E+00	1.3862943611E+00
-4.0000000000E-01	1.2770640418E+00	1.2770640594E+00
-3.0000000000E-01	1.1889164791E+00	1.1889164798E+00
-2.0000000000E-01	1.1157177566E+00	1.1157177566E+00
-1.0000000000E-01	1.0536051566E+00	1.0536051566E+00
2.0000000000E-01	9.1160778397E-01	9.1160778397E-01
3.0000000000E-01	8.7454754826E-01	8.7454754823E-01
4.0000000000E-01	8.4118059184E-01	8.4118059155E-01
5.0000000000E-01	8.1093021771E-01	8.1093021622E-01
6.0000000000E-01	7.8333938748E-01	7.8333938208E-01
7.0000000000E-01	7.5804037412E-01	7.5804035866E-01
8.0000000000E-01	7.3473336839E-01	7.3473333113E-01
9.0000000000E-01	7.1317106367E-01	7.1317098464E-01
1.0000000000E+00	6.9314733235E-01	6.9314718056E-01
1.1000000000E+00	6.7448876460E-01	6.7448849521E-01
1.2000000000E+00	6.5704824872E-01	6.5704780030E-01
1.3000000000E+00	6.4070003327E-01	6.4069932533E-01
1.4000000000E+00	6.2533588156E-01	6.2533481240E-01
1.5000000000E+00	6.1086204301E-01	6.1086048792E-01
1.6000000000E+00	5.9719684325E-01	5.9719465314E-01
1.7000000000E+00	5.8426874838E-01	5.8426574883E-01
1.8000000000E+00	5.7201479673E-01	5.7201078732E-01
1.9000000000E+00	5.6037931799E-01	5.6037407210E-01

con centro en el origen y radio r , entonces la sucesión

$$\Lambda = (r_{i,0})_{i=0}^{\infty}$$

converge uniformemente a f en cada subconjunto compacto de $B(0, r)$. ■

Como respaldo teórico del ejemplo (3.5), se da el siguiente resultado [5].

Proposición 3.10. Sea f una función meromorfa en K un subconjunto compacto de \mathbb{C} . Para $\epsilon > 0$ y $\delta > 0$, existe $k > 0$ tal que si $i > k$, se satisface

$$| r_{i,i}(x) - f(x) | < \epsilon, \quad x \in K_i,$$

donde K_i es un subconjunto de K , tal que la medida de Hausdorff de $K - K_i$ es menor a δ . ■

Una función meromorfa es una función analítica, excepto en un número finito de polos. Como la función del ejemplo (3.5) es meromorfa en el disco unitario, la sucesión $(r_{m,m})_{m=0}^{\infty}$ converge a f . Otros ejemplos en que podemos aplicar el resultado (3.5) son las funciones de Bessel y la función de error.

3.8. Algoritmo cociente-diferencia.

Como ya hemos marcado el algoritmo de Gauss no es eficiente para encontrar aproximaciones de Padé. Pues el trabajo computacional que hacemos para obtener una, no sirve para

encontrar otras. En esta sección presentamos el algoritmo cociente diferencia de Rutishauser [7], como una alternativa para construir, de una manera más eficiente, aproximantes de Padé. Empezamos con el siguiente resultado básico.

Proposición 3.8. Si $r_{m,n}(x) = p_1 / q_1$ y $r_{m+k,n+l} = p_2 / q_2$ con $k, l \geq 0$, entonces existe un polinomio $\mu(x)$ con $\delta(\mu) \leq \max(k-1, l-1)$, tal que

$$(p_1 q_2 - p_2 q_1)(x) = x^{m+n+1} \mu(x)$$

Demostración. El polinomio $v = p_1 q_2 - p_2 q_1$ tiene las siguientes propiedades:

$$\begin{aligned} \omega(v) &= \omega[(f q_2 - p_2) q_1 - (f q_1 - p_1) q_2] \\ &\geq \min(\omega(f q_2 - p_2), \omega(f q_1 - p_1)) \\ &\geq m + n + 1 \end{aligned}$$

y

$$\begin{aligned} \delta(v) &\leq \max(\delta p_1 + \delta q_2, \delta p_2 + \delta q_1) \\ &\leq m + n + \max(k, l) \\ &= m + n + 1 + \max(k-1, l-1) \end{aligned}$$

De estas desigualdades se deduce el resultado. ■

Ahora considere la siguiente sucesión de aproximaciones racionales localizadas sobre una escalera descendente en la tabla de Padé

$$T_k = \left\{ r_{k,0}, r_{k+1,0}, r_{k+1,1}, r_{k+2,1}, \dots \right\}$$

y una fracción continua de la forma

$$G_k(x) = \sum_{i=0}^k d_i x^i + \frac{d_{k+1} x^{k+1}}{1} + \sum_{i=2}^{\infty} \frac{d_{k+i} x}{1}$$

Proposición 3.9. Si cada tres elementos consecutivos en T_k son diferentes, entonces una fracción continua de la forma de G_k existe con $d_{k+i} \neq 0$, para $i \geq 1$, tal que su n -ésima convergente es igual al n -ésimo elemento de T_k para $n \geq 0$.

Demostración. Para empezar es fácil deducir que $d_i = c_i$ para $i = 0, 1, \dots, k+1$, del hecho de que se quiere

$$r_{k,0}(x) = \sum_0^k d_i x^i \quad \text{y} \quad r_{k+1,0}(x) = \sum_0^{k+1} d_i x^i$$

Ahora se sabe que una fracción continua para la cual su n -ésima convergente es igual P_n / Q_n , está dada por las fórmulas (2.11).

$$P_0 + \frac{P_1 - P_0}{1} + \sum_{i=1}^{\infty} \frac{\frac{P_i Q_{i+1} - P_{i+1} Q_i}{P_i Q_{i-1} - P_{i-1} Q_i}}{\frac{P_{i+1} Q_{i-1} - P_{i-1} Q_{i+1}}{P_i Q_{i-1} - P_{i-1} Q_i}} \quad (3.21)$$

Como se quiere que la i -ésima convergente sea igual al i -ésimo elemento de T_k , y como el i -ésimo elemento de T_k es

$$\frac{P_i}{Q_i} = \begin{cases} r_k + \frac{i+1}{2}, \frac{i+1}{2} - 1, & \text{si } i \text{ es impar} \\ r_k + \frac{i}{2}, \frac{i}{2} & \text{si } i \text{ es par} \end{cases} \quad (3.22)$$

entonces, utilizando la proposición (3.8) se deducen las siguientes fórmulas para los numeradores y denominadores de (3.21)

$$\begin{aligned} P_i Q_{i+1} - P_{i+1} Q_i &= \mu_1 x^{k+i} \\ P_i Q_{i-1} - P_{i-1} Q_i &= \mu_2 x^{k+i-1} \\ P_{i+1} Q_{i-1} - P_{i-1} Q_{i+1} &= \mu_3 x^{k+i-1} \end{aligned}$$

donde μ_1, μ_2, μ_3 son constantes. Estas fórmulas al ser

sustituidas en (3.21) y aplicar la transformación de equivalencia $p_1 = 1$ y $p_i = \mu_2/\mu_3$, demuestran el resultado. ■

Una deficiencia de la proposición (3.9) es que no sabemos como calcular los coeficientes d_{k+1} para $i \geq 2$. Antes de abordar este problema se da un resultado que es consecuencia de (3.9).

Proposición 3.10. Si la n -ésima convergente de $G_0(x)$ coincide con el n -ésimo elemento de T_0 , entonces $G_0(x)$ es una fracción continua correspondiente para la serie $\sum c_i x^i$.

Demostración. Si P_n/Q_n es la n -ésima convergente de $G_0(x)$, entonces $\omega(fQ_n - P_n) \geq n + 1$ y $Q_n(0) = 1$. Esto se obtiene de las fórmulas (3.22). Luego entonces

$$\left[f - \frac{P_n}{Q_n} \right]^{(j)}(0) = 0, \quad j = 0, 1, \dots, n$$

donde se ha utilizado que $Q_n(0) = 1$, para deducir que Q_n es diferente de cero en una vecindad del origen. ■

Ahora continuamos con el problema de calcular los

numeradores parciales de G_k . Consideremos la fracción continua con k variable

$$F_k(x) = \sum_{i=0}^{\infty} c_k x^i + \frac{c_{k+1} x^{k+1}}{1} + \frac{-q_1^{(k+1)} x}{1} + \frac{-e_1^{(k+1)} x}{1} +$$

$$+ \frac{-q_2^{(k+1)} x}{1} + \frac{-e_2^{(k+1)} x}{1} + \dots$$

Si se calculan los coeficientes q 's y e 's utilizando la proposición (3.9), entonces las n -ésimas convergentes de F_{k-1} y F_k son iguales a los n -ésimos elementos de T_{k-1} y T_k respectivamente. Utilizando este hecho es fácil observar que la parte par de F_k es igual a la parte impar de F_{k-1} , pues tienen en común las convergentes $r_{k,0}, r_{k+1,1}, r_{k+2,2}, \dots$. Por lo tanto los denominadores y numeradores parciales de la parte par de F_k son iguales a los denominadores y numeradores parciales de la parte impar de F_{k-1} respectivamente. Utilizando las fórmulas de las proposiciones (2.6) y (2.7), se puede calcular la parte par de F_k para obtener

$$\text{Par}(F_k) = \sum_0^k c_i x^i + \frac{c_{k+1} x^{k+1}}{1 - q_1^{(k+1)} x} + \frac{-q_1^{(k+1)} e_1^{(k+1)} x^2}{1 - (q_2^{(k+1)} + e_1^{(k+1)}) x} +$$

$$+ \frac{-q_2^{(k+1)} e_2^{(k+1)} x^2}{1 - (q_3^{(k+1)} + e_2^{(k+1)})x} + \dots \quad (3.23)$$

Mientras que la parte impar de F_{k-1} es

$$\begin{aligned} \text{Impar}(F_{k-1}) = & \sum_0^k c_i x^i + \frac{c_k q_1^{(k)} x^{k+1}}{1 - (q_1^{(k)} + e_1^{(k)})x} + \frac{-e_1^{(k)} q_2^{(k)} x^2}{1 - (q_2^{(k)} + e_2^{(k)})x} \\ & \frac{-e_2^{(k)} q_3^{(k)} x^2}{1 - (q_3^{(k)} + e_3^{(k)})x} + \dots \quad (3.24) \end{aligned}$$

Haciendo la igualdad de los numeradores y denominadores parciales de (3.23) y (3.24), se deducen las siguientes fórmulas

$$\begin{aligned} q_l^{(k)} &= \frac{c_{k+1}}{c_k} \\ e_l^{(k)} &= e_{l-1}^{(k+1)} + q_l^{(k+1)} - q_l^{(k)} \\ q_{l+1}^{(k)} &= \frac{q_l^{(k+1)} * e_l^{(k+1)}}{e_l^{(k)}} \end{aligned} \quad (3.25)$$

donde se define $e_0^{(k)} = 0$, para que la fórmula para calcular las e 's siga siendo válida cuando $l = 1$.

Los números e 's y q 's son arreglados en una tabla, donde un super-índice sirve para indicar la diagonal en la que se localiza el número y un sub-índice la columna. Esta tabla es llamada la tabla cociente-diferencia o tabla qd (pág. 96).

Las fórmulas (3.25) se pueden recordar fácilmente si se observan que los términos que intervienen en ellas se localizan en los vértices de un rombo, y el elemento que se encuentra en la esquina derecha es el que se calcula en función de los otros tres.

$$\begin{array}{cccccc}
 & & q_1^{(k)} & & & e_{l-1}^{(k)} \\
 e_{l-1}^{(k+1)} & & & e_1^{(k)} & & q_{l-1}^{(k+1)} & & & q_1^{(k)} \\
 & & q_1^{(k+1)} & & & & e_{l-1}^{(k+1)} & & \\
 & & & & & & & &
 \end{array}$$

Tabla cociente diferencia.

$e_0^{(1)}$	$q_1^{(1)}$	$e_1^{(1)}$	$q_2^{(1)}$	$e_2^{(1)}$	$q_3^{(1)}$	$e_3^{(1)}$.	.
$e_0^{(2)}$	$q_1^{(2)}$	$e_1^{(2)}$	$q_2^{(2)}$	$e_2^{(2)}$	$q_3^{(2)}$	$e_3^{(2)}$.	.
$e_0^{(3)}$	$q_1^{(3)}$	$e_1^{(3)}$	$q_2^{(3)}$	$e_2^{(3)}$	$q_3^{(3)}$	$e_3^{(3)}$.	.
.
.
.

La siguiente etapa consiste en dar un procedimiento,

basado en el algoritmo qd , para calcular aproximaciones de Padé.

Supóngase que $m \geq n$ en $r_{m,n}$. Es claro que tal aproximación racional es un elemento de T_{m-n} , y es el $2n$ -ésimo elemento. Por lo tanto $r_{m,n}$ se puede calcular a partir de la $2n$ -ésima convergente de F_{m-n}

$$\sum_0^{m-n} c_i x^i + \frac{c_{m-n+1} x^{m-n+1}}{1} + \frac{-q_1^{(m-n+1)} x}{1} + \frac{-e_1^{(m-n+1)} x}{1} + \dots + \frac{-q_n^{(m-n+1)} x}{1} \quad (3.26)$$

Para determinar a (3.26), es necesario decir cómo se calculan los coeficientes $q_1^{(m-n+1)}, \dots, q_n^{(m-n+1)}, e_1^{(m-n+1)}, \dots, e_{n-1}^{(m-n+1)}$. La siguiente región de la tabla cociente-diferencia es la que necesitamos formar para determinar estos coeficientes.

$$\begin{array}{ccc} q_1^{(m-n+1)} & \text{---} & q_n^{(m-n+1)} \\ | & & / \\ q_1^{(m+n-1)} & & \end{array} \quad (3.27)$$

Para generar la región (3.27), basta con conocer los elementos q 's de la columna uno, el resto se genera por columnas de izquierda a derecha utilizando las ecuaciones (3.25). Como

$$q_1^{(m-n+i)} = \frac{c_{m-n+1+i}}{c_{m-n+i}}, \quad i = 1, 2, \dots, 2n-1$$

entonces la mínima información necesaria para calcular (3.27) son los coeficientes $c_{m-n+1}, c_{m-n+2}, \dots, c_{m+n}$ de f .

Para completar el algoritmo para calcular aproximaciones racionales, es necesario decir que hacer si $m \leq n$. La siguiente propiedad de las aproximaciones soluciona el problema.

Proposición 3.11. Covarianza Recíproca. Si $f(0) \neq 0$ entonces, para toda $m, n \geq 0$ se satisface

$$\left[\frac{m}{n} \right]_f * \left[\frac{n}{m} \right]_{1/f}^{(x)} = 1$$

donde los paréntesis se utilizan para denotar la aproximación de Padé.

Demostración. Esta se deduce de la siguiente igualdad al multiplicarla por $1/f$.

$$f(x)q(x) - p(x) = \alpha_{m+n+1} x^{m+n+1} + \alpha_{m+n+2} x^{m+n+2} + \dots \blacksquare$$

En PADE se encuentra programado el algoritmo cociente-diferencia exactamente como lo hemos desarrollado en esta sección. A continuación presentamos el listado de este. Las variables m y n denotan el orden de la aproximante, $Coef$ los coeficientes del desarrollo en serie de la función, e y q son vectores que contienen los elementos de la diagonal $(m-n+1)$ de la tabla de Padé, Num y Den son vectores que sirven para guardar los numeradores y denominadores de (3.26). Los procedimientos QD , $ForFracCont$, Eva y $ConEva$ son para formar la tabla cociente diferencia, formar los numeradores y denominadores de (3.26), evaluar un polinomio y evaluar una fracción continua a través del algoritmo de sustitución hacia atrás respectivamente.

```

Unit   Algo22;
Procedure   Aprx22;
Begin
  If     ((n[t]>1)and(m[t]>=n[t]))   Then
    Begin
      QD(Coef,e,q,m[t],n[t]);
      X1:=Xmin;
      Den[1]:=1.0;
      Num[1]:=Coef[m[t]-n[t]+1]*Pot(m[t]-n[t]+1,X1);
      ForFracCont(Num,Den,e,q,Coef,n[t],X1,m[t],n[t]);
      Eva(m[t]-n[t],Coef,X1,BO);
      ConEva(BO,Num,Den,Conv,2*n[t]);
      Y1:=Conv;
      For r:=1 to 200 Do

```

```

Begin
  X2:=Xmin+((Xmax-Xmin)*r)/200;
  Den[1]:=1.0;
  Num[1]:=Coef[m[t]-n[t]+1]*Pot(m[t]-n[t]+1,X2);
  ForFracCont(Num,Den,e,q,Coef,n[t],X2,m[t],n[t]);
  Eva(m[t]-n[t],Coef,X2,B0);
  ConEva(B0,Num,Den,Conv,2*n[t]);
  Y2:=Conv;
  Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
  X1:=X2;
  Y1:=Y2;
End;
End;

If ((n[t]=0)and(m[t]>=n[t])) Then
Begin
  X1:=Xmin;
  Eva(m[t],Coef,X1,B0);
  Y1:=B0;
  For r:=1 to 200 Do
  Begin
    X2:=Xmin+((Xmax-Xmin)*r)/200;
    Eva(m[t],Coef,X2,B0);
    Y2:=B0;
    Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
    X1:=X2;
    Y1:=Y2;
  End;
End;

If ((n[t]=1)and(m[t]>=n[t])) then
Begin
  q[m[t]]:=Coef[m[t]+1]/Coef[m[t]];
  X1:=Xmin;
  Eva(m[t]-1,Coef,X1,B0);
  If Abs(1-q[m[t]]*X1)>MinimoReal then
    Y1:=B0+(Coef[m[t]]*Pot(m[t],X1)/(1-q[m[t]]*X1));
  For r:=1 to 200 do
  Begin
    X2:=Xmin+((Xmax-Xmin)*r)/200;
    Eva(m[t]-1,Coef,X2,B0);
    If Abs(1-q[m[t]]*X2)>MinimoReal then
      Y2:=B0+(Coef[m[t]]*Pot(m[t],X2)/(1-q[m[t]]*X2));
    Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
  End;
End;

```

```

        X1:=X2;
        Y1:=Y2;
    End;
End;
End;
End.

```

```

Unit    Algo21;
Procedure  Aprx21;
Begin
    If    ((m[t]>1)and(m[t]<n[t]))    Then
        Begin
            SerRec(Coef,m[t]+n[t],CoefRec);
            QD(CoefRec,e,q,n[t],m[t]);
            X1:=Xmin;
            Den[1]:=1.0;
            Num[1]:=CoefRec[n[t]-m[t]+1]*Pot(n[t]-m[t]+1,X1);
            ForFracCont(Num,Den,e,q,CoefRec,n[t],X1,n[t],m[t]);
            Eva(n[t]-m[t],CoefRec,X1,B0);
            ConEva(B0,Num,Den,Conv,2*m[t]);
            Y1:=1/Conv;
            For r:=1 to 200 Do
                Begin
                    X2:=Xmin+((Xmax-Xmin)*r)/200;
                    Den[1]:=1.0;
                    Num[1]:=CoefRec[n[t]-m[t]+1]*Pot(n[t]-m[t]+1,X2);
                    ForFracCont(Num,Den,e,q,CoefRec,m[t],X2,n[t],m[t]);
                    Eva(n[t]-m[t],CoefRec,X2,B0);
                    ConEva(B0,Num,Den,Conv,2*m[t]);
                    Y2:=1/Conv;
                    Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
                    X1:=X2;
                    Y1:=Y2;
                End;
            End;
        End;
    If    ((m[t]=0)and(m[t]<n[t]))    Then
        Begin
            SerRec(Coef,m[t]+n[t],CoefRec);X1:=Xmin;
            Eva(n[t]-m[t],CoefRec,X1,B0);
            If    Abs(B0)>MinimoReal    then
                Y1:=1/B0;
            For r:=1 to 200 Do
                Begin
                    X2:=Xmin+((Xmax-Xmin)*r)/200;
                    Eva(n[t]-m[t],CoefRec,X2,B0);
                End;
            End;
        End;
    End;
End.

```

```

      If Abs(B0)>MinimoReal then
        Y2:=1/B0;
        Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
        X1:=X2;Y1:=Y2;
      End;
    End;

  If ((m[t]=1)and(m[t]<n[t])) then
    Begin
      SerRec(Coef,1+n[t],CoefRec);
      q[n[t]]:=CoefRec[n[t]+1]/CoefRec[n[t]];
      X1:=Xmin;
      Eva(n[t]-1,CoefRec,X1,B0);
      Y1:=1/(B0+(CoefRec[n[t]]*Pot(n[t],X1)/(1-q[n[t]]*X1)));
      For r:=1 to 200 do
        Begin
          X2:=Xmin+((Xmax-Xmin)*r)/200;
          Eva(n[t]-1,CoefRec,X2,B0);
          Y2:=1/(B0+CoefRec[n[t]]*Pot(n[t],X2)/(1-q[n[t]]*X2));
          Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
          X1:=X2;
          Y1:=Y2;
        End;
      End;
    End;
  End;
End.

```

3.9. Algoritmo epsilon.

El algoritmo que a continuación se estudiará es el epsilon de Wynn [11]. Empezamos con la siguiente propiedad.

Proposición 3.12. Para las convergentes de la fracción continua (2.1)

$$C_k = P_k / Q_k = b_0 + \sum_{i=1}^k \frac{a_i}{b_i}$$

se satisface, siempre y cuando tres convergentes sucesivos son distintos, que

$$\frac{1}{c_{k+1} - c_k} + \frac{1}{c_k - c_{k-1}} = \frac{b_{k+1}b_{k+2}Q_k^2}{P_{k+2}Q_k - P_kQ_{k+2}} \quad (3.28)$$

Demostración. Basta aplicar las fórmulas (2.6), (2.5) y (2.8) del capítulo dos para obtener el resultado. La deducción es como sigue.

$$\begin{aligned} \frac{1}{c_{k+1} - c_k} + \frac{1}{c_k - c_{k-1}} &= \frac{Q_{k+1}Q_k}{(-1)^{k+2} a_1 \dots a_{k+1}} + \frac{Q_k Q_{k-1}}{(-1)^{k+1} a_1 \dots a_k} \\ &= \frac{Q_k}{(-1)^k a_1 \dots a_k} \left[\frac{Q_{k+1} - a_{k+1}Q_{k-1}}{a_{k+1}} \right] \\ &= \frac{b_{k+1}Q_k^2}{(-1)^k a_1 \dots a_{k+1}} = \frac{b_{k+1}b_{k+2}Q_k^2}{P_{k+2}Q_k - P_kQ_{k+2}} \quad \blacksquare \end{aligned}$$

El siguiente resultado expresa una relación de recurrencia entre los elementos contiguos de la tabla de Padé.

$$\begin{array}{rcl}
 r_{m-1,n} & = & N \\
 r_{m,n-1} = W & & r_{m,n} = \mathcal{E} & & r_{m,n+1} = \mathcal{E} \\
 & & r_{m+1,n} = \mathcal{Y}
 \end{array}$$

Proposición 3.13. (Regla cruzada). Si $r_{m,-1} = \infty$ y $r_{-1,n} = 0$, entonces la siguiente relación entre N , \mathcal{Y} , \mathcal{E} , W y \mathcal{E} es válida

$$(N - \mathcal{E})^{-1} + (\mathcal{Y} - \mathcal{E})^{-1} = (\mathcal{E} - \mathcal{C})^{-1} + (W - \mathcal{E})^{-1} \quad (3.29)$$

Demostración. Se aplica la proposición (3.12) a las fracciones continuas F_{m-n} y F_{m-n+1} , las cuales se definieron en la sección 3.8, con $k = 2n$ y $k = 2n+1$ respectivamente, para deducir

$$\frac{1}{r_{m+1,n} - r_{m,n}} + \frac{1}{r_{m,n} - r_{m,n-1}} = \frac{Q_{2n}^2}{P_{2n+2}Q_{2n} - P_{2n}Q_{2n+2}}$$

$$\frac{1}{r_{m,n+1} - r_{m,n}} + \frac{1}{r_{m,n} - r_{m-1,n}} = \frac{Q_{2n}^2}{P_{2n+2}Q_{2n} - P_{2n}Q_{2n+2}} \quad \blacksquare$$

Ahora si se introduce la siguiente notación para las aproximaciones de Padé $r_{m,n} = \varepsilon_{2n}^{(m-n)}$, entonces se obtiene la

siguiente configuración para la tabla de Padé.

Tabla epsilon

$\epsilon_0^{(0)}$				
$\epsilon_0^{(1)}$	$\epsilon_2^{(0)}$			
$\epsilon_0^{(2)}$	$\epsilon_2^{(1)}$	$\epsilon_4^{(0)}$		
$\epsilon_0^{(3)}$	$\epsilon_2^{(2)}$	$\epsilon_4^{(1)}$	$\epsilon_6^{(0)}$...
⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	

donde $\epsilon_0^{(m)} = r_{m,0}$ es el polinomio de Taylor de grado m de f alrededor del origen.

La tabla ϵ puede ser completada en las columnas impares de la siguiente forma. Se definen

$$\epsilon_{2n+1}^{(m-n-1)} = \epsilon_{2n-1}^{(m-n)} + \frac{1}{\epsilon_{2n}^{(m-n)} - \epsilon_{2n}^{(m-n-1)}} \quad (3.30)$$

$$\epsilon_{-1}^{(m)} = \epsilon_{2n}^{(-n-1)} = 0$$

estas fórmulas son válidas para $m, n = 0, 1, 2, \dots$

Utilizando la proposición (3.13), se puede demostrar una relación similar a (3.30) para generar los ϵ_n de las columnas pares.

Proposición 3.14. Para $m, n = 0, 1, 2, \dots$ es válido

$$\epsilon_{2n}^{(m-n)} = \epsilon_{2n-2}^{(m-n+1)} + \frac{1}{\epsilon_{2n-1}^{(m-n+1)} - \epsilon_{2n-1}^{(m-n)}} \quad (3.31)$$

Demostración. Por la identidad (3.29) se deduce que

$$\begin{aligned} & (\epsilon_{2n+2}^{(m-n-1)} - \epsilon_{2n}^{(m-n)})^{-1} + (\epsilon_{2n}^{(m-n)} - \epsilon_{2n}^{(m-n-1)})^{-1} = \\ & = (\epsilon_{2n}^{(m-n+1)} - \epsilon_{2n}^{(m-n)})^{-1} + (\epsilon_{2n}^{(m-n)} - \epsilon_{2n-2}^{(m-n+1)}) \end{aligned} \quad (3.32)$$

A partir de (3.30) se obtienen las ecuaciones

$$(\epsilon_{2n}^{(m-n)} - \epsilon_{2n}^{(m-n-1)})^{-1} = \epsilon_{2n+1}^{(m-n-1)} - \epsilon_{2n-1}^{(m-n)} \quad (3.33)$$

$$(\epsilon_{2n}^{(m-n+1)} - \epsilon_{2n}^{(m-n)})^{-1} = \epsilon_{2n+1}^{(m-n)} - \epsilon_{2n-1}^{(m-n+1)}$$

Al sustituir las ecuaciones (3.33) en (3.32) se obtiene

$$\begin{aligned}
& (\epsilon_{2n+2}^{(m-n-1)} - \epsilon_{2n}^{(m-n)})^{-1} + \epsilon_{2n+1}^{(m-n-1)} - \epsilon_{2n-1}^{(m-n)} = \\
& = \epsilon_{2n+1}^{(m-n)} - \epsilon_{2n-1}^{(m-n+1)} + (\epsilon_{2n}^{(m-n)} - \epsilon_{2n-2}^{(m-n+1)})^{-1}
\end{aligned} \tag{3.34}$$

Ahora utilizando (3.34) es fácil demostrar por inducción la fórmula (3.31), después de un despeje

$$\epsilon_{2n-1}^{(m-n+1)} = \epsilon_{2n-1}^{(m-n)} + \frac{1}{\epsilon_{2n}^{(m-n)} - \epsilon_{2n-2}^{(m-n+1)}} \tag{3.35}$$

Se supone que (3.35) se cumple para n y se demostrará para $n+1$, es decir

$$\epsilon_{2n+1}^{(m-n)} = \epsilon_{2n+1}^{(m-n-1)} + \frac{1}{\epsilon_{2n+2}^{(m-n-1)} - \epsilon_{2n}^{(m-n)}} \tag{3.36}$$

Al aplicar (3.34) al segundo miembro de (3.36) y utilizar la hipótesis de inducción se obtiene el resultado. ■

El algoritmo ϵ también fue programado en PADE, a continuación presentamos un listado de la unidad ALGO3.PAS, que es la unidad que lo contiene. Las variables m y n son para denotar el orden de la aproximante, *Coef* los coeficientes del

desarrollo en serie de la función, *Parcial* es un vector que contiene los polinomios de Taylor evaluados en un punto, los procedimientos *Eva* y *Epsilon* se encargan de evaluar un polinomio y la región de la tabla ϵ para construir una aproximante.

```

Unit   Algo3;
Procedure  Aprx3;
Begin
  If    ((m[t]>=n[t])and(n[t]>0))    then
    Begin
      X1:=Xmin;
      For  s:=(m[t]-n[t])  to  (m[t]+n[t])  do
        Begin
          Eva(s,Coef,X1,Result);Parcial[s]:=Result;
        End;
      If  Abs(X2)>0.1  then
        Begin
          Epsilon(m[t],n[t],Evaluacion,Parcial);
          Y1:=Evaluacion;
        End;
      If  Abs(X2)<0.1  then
        Y1:=Coef[0];
      For  r:=1  to  200  do
        Begin
          X2:=Xmin+((Xmax-Xmin)*r)/200;
          For  s:=(m[t]-n[t])  to  (m[t]+n[t])  do
            Begin
              Eva(s,Coef,X2,Result);
              Parcial[s]:=Result;
            End;
          If  Abs(X2)>0.1  then
            Begin
              Epsilon(m[t],n[t],Evaluacion,Parcial);
              Y2:=Evaluacion;
            End;
          If  Abs(X2)<0.1  then
            Y2:=Coef[0];
          Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
          X1:=X2;Y1:=Y2;
        End;
      End;
    End;
End;

```

```

If ((m[t]<n[t])and(m[t]>0)) then
Begin
SerRec(Coef,m[t]+n[t],CoefRec);
X1:=Xmin;
For s:=n[t]-m[t] to m[t]+n[t] do
Begin
Eva(s,CoefRec,X1,Result);
Parcial[s]:=Result;
End;
If Abs(X1)>0.1 then
Begin
Epsilon(n[t],m[t],Evaluacion,Parcial);
If Abs(Evaluacion)>MinimoReal then
Y1:=1/Evaluacion;
If Abs(Evaluacion)<MinimoReal then
Y1:=MaximoReal;
End;
If Abs(X1)<0.1 then
Y1:=Coef[0];
For r:=1 to 200 do
Begin
X2:=Xmin+((Xmax-Xmin)*r)/200;
For s:=n[t]-m[t] to m[t]+n[t] do
Begin
Eva(s,CoefRec,X2,Result);
Parcial[s]:=Result;
End;
If Abs(X2)>0.1 then
Begin
Epsilon(n[t],m[t],Evaluacion,Parcial);
If Abs(Evaluacion)>MinimoReal then
Y2:=1/Evaluacion;
If Abs(Evaluacion)<MinimoReal then
Y2:=MaximoReal;
End;
If Abs(X1)<0.1 then
Y2:=Coef[0];
Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
X1:=X2;
Y1:=Y2;
End;
End;
End;

```

```

If n[t]=0 then
  Begin
    X1:=Xmin;
    Eva(m[t],Coef,X1,Result);
    Y1:=Result;
    For r:=1 to 200 do
      Begin
        X2:=Xmin+((Xmax-Xmin)*r)/200;
        Eva(m[t],Coef,X2,Result);
        Y2:=Result;
        Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
        Barras(t,r);
        X1:=X2;
        Y1:=Y2;
      End;
    End;
  End;
If m[t]=0 then
  Begin
    SerRec(Coef,m[t]+n[t],CoefRec);
    X1:=Xmin;
    Eva(n[t],CoefRec,X1,Result);
    If Abs(Result)>MinimoReal then
      Y1:=1/Result;
    If Abs(Result)<MinimoReal then
      Y1:=MaximoReal;
    For r:=1 to 200 do
      Begin
        X2:=Xmin+((Xmax-Xmin)*r)/200;
        Eva(n[t],CoefRec,X2,Result);
        If Abs(Result)>MinimoReal then
          Y2:=1/Result;
        If Abs(Result)<MinimoReal then
          Y2:=MaximoReal;
        Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
        X1:=X2;
        Y1:=Y2;
      End;
    End;
  End;
End;
End.

```

Se puede comprender un poco más del algoritmo e hablando

un poco sobre su origen.

La transformación

$$e_m(S_n) = \frac{\begin{array}{ccc} S_n & & S_{n+m} \\ \Delta S_n & \cdot & \Delta S_{n+m} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \Delta S_{n+m-1} & \cdot & \Delta S_{n+2m-1} \end{array}}{\begin{array}{ccc} 1 & \cdot & 1 \\ \Delta S_n & \cdot & \Delta S_{n+m} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \Delta S_{n+m-1} & \cdot & \Delta S_{n+2m-1} \end{array}} \quad (3.37)$$

estudiada por Schmidt y Shanks [8] es la fuente del algoritmo ϵ , aquí $(S_n)_{n=0}^{\infty}$ es una sucesión y Δ algun operador. Sin dar más detalles, esta transformación mostró ser útil para acelerar la convergencia de la sucesión $(S_n)_{n=0}^{\infty}$. Wynn en 1956 demostró la conexión entre la transformación de Shanks y las aproximantes de Padé [11]. A continuación se presenta la demostración de este hecho

Proposición 3.15. Sea $f(x) = \sum c_i x^i$ una serie con

$c_0 \neq 0$. Si la aproximación de Padé de orden (m, n) para f es $r_{m, n}$
 $= p_0/q_0$ y $D = \det (D_{m, n}) \neq 0$ entonces

$$P_0(x) = \frac{1}{D} \begin{vmatrix} S_m(x) & x S_{m-1}(x) & \dots & x^n S_{m-n}(x) \\ c_{m+1} & & & \\ \cdot & & & \\ \cdot & & & \\ c_{m+n} & & & \end{vmatrix} \quad (3.38)$$

mientras que para q_0 se tiene

$$q_0(x) = \frac{1}{D} \begin{vmatrix} 1 & x & \dots & x^n \\ c_{m+1} & & & \\ \cdot & & & \\ \cdot & & & \\ c_{m+n} & & & \end{vmatrix} \quad (3.39)$$

donde $S_k = \sum_0^k c_i x^i$ y $D_{m, n}$ es una matriz como la de la proposición
 (3.6).

Demostración. Como $D \neq 0$, el sistema (3.6) tiene una solución única para la elección $b_0 = 1$. Por lo tanto el siguiente sistema tiene una solución no trivial.

$$\begin{aligned}
 (1-q_0(x))b_0 + x b_1 + \dots + x^n b_n &= 0 \\
 c_{m+1} b_0 + c_m b_1 + \dots + c_{m+1-n} b_n &= 0 \\
 \vdots & \\
 \vdots & \\
 \vdots & \\
 c_{m+n} b_0 + c_{m+n-1} b_1 + \dots + c_m b_n &= 0
 \end{aligned}$$

Esto implica que el determinante de la matriz de coeficientes del sistema es igual a cero

$$\begin{vmatrix}
 (1-q_0(x)) & x & \dots & x^n \\
 c_{m+1} & c_m & \dots & c_{m+1-n} \\
 \vdots & \vdots & & \vdots \\
 \vdots & \vdots & & \vdots \\
 \vdots & \vdots & & \vdots \\
 c_{m+n} & c_{m+n-1} & \dots & c_m
 \end{vmatrix} = 0$$

De aquí se deduce la fórmula para q_0 .

Para demostrar la fórmula para $p_0(x)$, considere el producto $f(x)q_0(x)$, y tómesese en cuenta que $p_0(x)$ coincide con este producto desde el grado cero hasta el grado $m+n$ y p_0 es de grado m . Entonces

$$f(x)q_0(x) = \frac{1}{D} \begin{vmatrix} f(x) & x f(x) & \dots & x^n f(x) \\ c_{m+1} & & & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ c_{m+n} & & & \end{vmatrix} \quad \blacksquare$$

Después de esta proposición mostraremos la conexión entre la transformación de Shanks y las aproximaciones de Padé. Para $f(x) = \sum c_i x^i$ consideremos la sucesión $(S_k)_{k=0}^{\infty}$ de sumas parciales. Si se aplica la transformación de Shanks a la sucesión de sumas parciales evaluadas en 1, $S_k = c_0 + \dots + c_k$, y el operador Δ es

$$\Delta S_k = S_{k-1} - S_k = c_{k+1}.$$

Entonces utilizando la proposición (3.15), se puede demostrar que $e_m(S_n)$ coincide con $r_{m+n,m}(1)$.

3.10. Algoritmo de Viscovatov.

Utilizando el algoritmo de Viscovatov (Cap.2 pag. 42) es posible generar los elementos de la escalera

$$T_k = \left\{ r_{k,0}, r_{k+1,0}, r_{k+1,1}, r_{k+2,1}, \dots \right\}$$

Si se aplica el algoritmo a la serie

$$\left[f(x) - \sum_0^k c_i x^i \right] / x^{k+1} = c_{k+1} + c_{k+2} x + c_{k+3} x^2 + \dots \quad (3.40)$$

se obtiene una fracción continua

$$\frac{c_{k+1}}{1} + \frac{d_{20} x}{d_{10}} + \frac{d_{30} x}{d_{20}} + \dots \quad (3.41)$$

para la cual se sabe que el desarrollo en serie de Taylor de la l -ésima convergente coincide con (3.40) hasta el grado $l-1$. Es decir si (3.41) es igual a $\sum e_i x^i$ entonces $e_0 = c_{k+1}, \dots, e_{l-1} = c_{k+l}$. De aquí se deduce que el desarrollo de Taylor de la l -ésima convergente de

$$\sum_{i=0}^{\infty} c_i x^i + \frac{c_{k+1} x^{k+1}}{1} + \frac{d_{20} x}{d_{10}} + \frac{d_{30} x}{d_{20}} + \dots \quad (3.42)$$

coincide con la serie $\sum c_i x^i$ desde el grado 0 hasta el grado $k+1$. Propiedad que tiene precisamente $r_{k+[1/2],[1/2]}$, el l -ésimo elemento de T_k , salvo por la condición $q_0(0) = 1$.

Para hacer que (3.42) satisfaga la condición $q_0(0) = 1$, le aplicamos la transformación de equivalencia $p_1=1$ y $p_i = 1/d_{i-1,0}$ para $i \geq 2$. Aplicando esta transformación, (3.42) se transforma en

$$\sum_{i=0}^{\infty} c_i x^i + \frac{c_{k+1} x^{k+1}}{1} + \frac{p_2 p_1 d_{20} x}{1} + \frac{p_3 p_2 d_{30} x}{1} + \dots$$

Esto implica que los nuevos numeradores se deben calcular utilizando las fórmulas (2.17) con el siguiente cambio

$$\begin{aligned} p_i p_{i-1} d_{i0} &= \left[d_{i-1,0} d_{i-2,1} - d_{i-2,0} d_{i-1,1} \right] / d_{i-1,0} d_{i-2,0} \\ &= \frac{d_{i-2,1}}{d_{i-2,0}} - \frac{d_{i-1,1}}{d_{i-1,0}} \end{aligned}$$

De aquí resulta que la tabla de Viscovatov de d 's se debe calcular utilizando las fórmulas

$$d_{1,i} = c_{k+i+1} \quad , \quad i = 0, 1, 2, \dots$$

$$d_{0,0} = 1 \quad , \quad d_{0i} = 0 \quad \text{para } i = 1, 2, 3, \dots$$

y para $j \geq 2$

$$d_{j,i} = \frac{d_{j-2,i+1}}{d_{j-2,0}} - \frac{d_{j-1,i+1}}{d_{j-1,0}}$$

3.11. Fórmulas de recurrencia.

El último procedimiento a considerar para construir aproximaciones racionales, consta de unas fórmulas de recurrencia entre las aproximaciones. Estas fórmulas nos permiten construir las aproximaciones que se localizan sobre una escalera ascendente en la tabla de Padé. La siguiente notaciones se utilizarán para las aproximaciones racionales

$$r_{m,n}(x) = \frac{\sum_0^m a_{m,n}^{(i)} x^i}{\sum_0^n b_{m,n}^{(i)} x^i}$$

Proposición 3.16. Si las aproximaciones de Padé

$$r_{m,n} = p_3 / q_3 \quad , \quad r_{m,n-1} = p_2 / q_2 \quad \text{y} \quad r_{m+1,n-1} = p_1 / q_1$$

son normales, entonces

$$\begin{aligned} p_3 &= a_{m,n-1}^{(m)} p_1 - a_{m+1,n-1}^{(m+1)} \times p_2 \\ q_3 &= a_{m,n-1}^{(m)} q_1 - a_{m+1,n-1}^{(m+1)} \times q_2 \end{aligned} \quad (3.43)$$

Demostración. Basta comprobar que p_3 / q_3 es solución del problema de aproximación de Padé de orden (m,n) para f . Es decir

$$\partial p_3 \leq m \quad , \quad \partial q_3 \leq n \quad \text{y} \quad \omega [f q_3 - p_3] \geq m+n+1$$

Las dos primeras desigualdades son claras y la última se deduce de

$$\text{que } \omega [f q_2 - p_2] \geq m+n-1 \quad \text{y} \quad \omega [f q_1 - p_1] \geq m+n. \quad \blacksquare$$

El siguiente resultado es complementario del anterior y se demuestra de la misma forma.

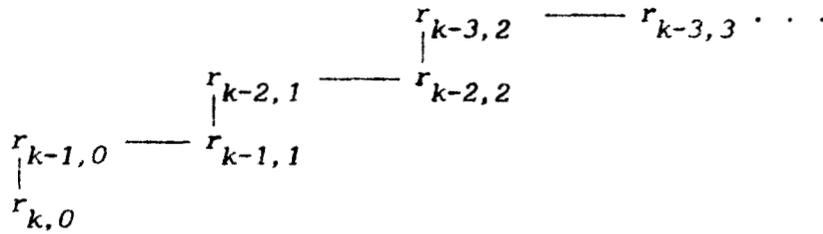
Proposición 2.23. Si $r_{m-1,n} = p_3 / q_3$, $r_{m,n} = p_2 / q_2$

y $r_{m,n-1} = p_1/q_1$ son normales entonces

$$p_3 = a_{m,n-1}^{(m)} p_2 - a_{m,n}^{(m)} p_1 \quad ,$$

$$q_3 = a_{m,n-1}^{(m)} q_2 - a_{m,n}^{(m)} q_1 \quad . \quad \blacksquare$$

Con éstos dos resultados se puede empezar con dos polinomios de Taylor de la función f y entonces generar las aproximaciones racionales localizadas sobre la siguiente escalera



CAPITULO IV. Interpolación racional osculatoria.

4.1. Introducción.

En este capítulo se estudia el tema de interpolación racional de Hermite u osculatoria, a saber, se busca una función racional $p(x)/q(x)$ que coincida con las derivadas de una función compleja en una colección de puntos, llamados nodos. Este capítulo lo empezamos estudiando la interpolación sencilla, es decir, cuando conocemos únicamente las derivadas de orden cero, nos referiremos a este tipo de interpolación como interpolación racional. Obviamente la interpolación racional de Hermite contempla, como casos particulares, la aproximación racional del capítulo tres y la interpolación racional simple. El trabajo a realizar es comprender como funcionan algunos de los algoritmos que hay en interpolación racional y de Hermite, para después formar un conjunto de programas que nos permitan graficar las interpolantes racionales en el monitor de una computadora.

Algunas de las aplicaciones interesantes de la interpolación racional osculatoria aparecen en la aproximación de

funciones. Como se mostrará, las interpolantes racionales son fáciles de construir y pueden servirnos para estimar el valor de una función con gran exactitud.

4.2. Definiciones

En esta sección se considera el problema de interpolación racional en su forma sencilla, es decir, cuando contamos únicamente con los valores de la función en los nodos.

Definición 4.1. El problema de interpolación racional de orden (m, n) para f consiste en encontrar polinomios

$$p(x) = a_m x^m + \dots + a_0 \quad \text{y} \quad q(x) = b_n x^n + \dots + b_0$$

tales que $p(x)/q(x)$ es irreducible y

$$f(x_i) = \frac{p(x_i)}{q(x_i)} \quad i = 0, 1, \dots, m+n \quad (4.1)$$

donde $(x_i)_{i=0}^{\infty}$ es una sucesión de puntos distintos en el dominio de f . ■

En lugar de considerar el problema (4.1) se puede plantear el siguiente sistema de ecuaciones lineales

$$f(x_i)q(x_i) - p(x_i) = 0 \quad i=0, 1, \dots, m+n \quad (4.2)$$

Como (4.2) tiene $m+n+2$ incógnitas con $m+n+1$ ecuaciones, tiene una solución no trivial por lo menos.

Para que los sistemas (4.1) y (4.2) sean equivalentes, es necesario y suficiente que los $(x_i)_{i=0}^{m+n}$ no sean raíces de q .

El siguiente resultado muestra que dos soluciones no triviales de (4.2) son equivalentes como funciones racionales.

Proposición 4.1. Si las dos parejas (p_1, q_1) y (p_2, q_2) son soluciones del problema de interpolación de orden (m, n) , entonces $p_1q_2 = p_2q_1$.

Demostración. El polinomio $\phi(x) = (p_1q_2 - p_2q_1)(x)$ tiene las $m+n+1$ raíces x_i porque

$$\phi(x_i) = [(fq_2 - p_2)q_1 - (fq_1 - p_1)q_2](x_i) = 0 \quad i=0, 1, \dots, m+n$$

Pero como $\partial\phi \leq m+n$, entonces $\phi \equiv 0$. ■

Utilizando la proposición (4.1) se puede dar la siguiente definición.

Definición 4.2. La única fracción irreducible p_0/q_0 que es solución del problema de interpolación (4.2), con $q_0(x_0) = 1$, recibe el nombre de interpolante racional de orden (m, n) para t . Esta función racional se representará por $r_{m, n}$. ■

Antes de terminar esta sección un comentario final. Las interpolantes racionales que se construyen a partir de una sucesión fija de nodos $(x_i)_{i=0}^{\infty}$, se colocan en una tabla, la cual recibe el nombre de *Tabla de Padé*.

Tabla de Padé.

$r_{0,0}$	$r_{0,1}$	$r_{0,2}$	$r_{0,3}$	$r_{0,4}$	·	·	·
$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$	$r_{1,4}$	·	·	·
$r_{2,0}$	$r_{2,1}$	$r_{2,2}$	$r_{2,3}$	$r_{2,4}$	·	·	·

4.3. Algoritmo de Gauss.

Si queremos encontrar la interpolante racional $r_{m, n}$

para una función f , hasta el momento contamos con el siguiente procedimiento. Aplicamos a (4.2) algún algoritmo de solución de sistemas de ecuaciones lineales, junto con alguna técnica de pivoteo.

En el paquete PADE se encuentran estructurados unos programas encargados de graficar las interpolantes racionales, de la siguiente forma. En la unidad ALGO4.PAS se forma el sistema de ecuaciones (4.2), después se llama a la unidad PROC1.PAS, que contiene el algoritmo de eliminación gaussiana con sustitución hacia atrás y pivoteo máximo, para calcular los coeficientes de la interpolante racional. A continuación se evalúa la interpolante racional en algún intervalo, se mandan los puntos (x_i, y_i) a la pantalla y se unen a través de segmentos de recta para obtener la gráfica. Este procedimiento para graficar una interpolante racional lo identificamos como el algoritmo uno o algoritmo de gauss, dentro del paquete PADE.

A continuación se muestra la parte principal de la unidad ALGO4.PAS. Las variables *Coef1* y *Coef2* guardan los coeficientes del sistema (4.2), m y n son el orden de la interpolante racional, F y *Nodo* son vectores que guardan el valor de la función y nodos, *Pot* es una función para calcular

potencias, a y b son las variables que guardan los coeficientes de los polinomios que forman la interpolante racional, *Gauss* y *Eva* son procedimientos para resolver un sistema de ecuaciones y evaluar un polinomio respectivamente. El procedimiento *Grafica* se encarga de trazar la gráfica de la interpolante en la pantalla.

```

Begin
  If n[t]>0 then
    Begin
      *****
      Se construyen los coeficientes del sistema (3.2) para n>0
      *****
      For r:=1 to m[t]+n[t]+1 do
        Begin
          For s:=1 to n[t] do
            Coef1[r,s]:=F[r-1]*(Pot(s,Nodo[r-1])-Pot(s,Nodo[0]));
          End;
        For r:=1 to m[t]+n[t]+1 do
          Begin
            For s:=n[t]+1 to m[t]+n[t]+1 do
              Begin
                If s>(n[t]+1) then
                  Coef1[r,s]:=-Pot(s-n[t]-1,Nodo[r-1]);
                If s=(n[t]+1) then
                  Coef1[r,s]:=-1;
              End;
            End;
          End;
        *****
        Términos constantes del sistema (3.2)
        *****
        For r:=0 to m[t]+n[t] do
          Coef2[r+1]:=-F[r];
        End;
  End;

```

```

If n[t]=0 then
  Begin
  .....
      Coeficientes del sistema (3.2) para n=0
  .....
    For r:=1 to m[t]+1 do
      Begin
        For s:=1 to m[t]+1 do
          Begin
            If s>1 then
              Coef1[r,s]:=Pot(s-1,Nodo[r-1]);
            If s=1 then
              Coef1[r,s]:=1;
          End;
        End;
      End;
    .....
      Términos constantes del sistema (3.2) para n=0
    .....
    For r:=0 to m[t] do
      Coef2[r+1]:=F[r];
    End;
    .....
      Se resuelve el sistema de ecuaciones (3.2)
    .....
    Gauss(Coef2,Coef1,m[t]+n[t]+1,Incognita);
    If n[t]>0 then
      Begin
      .....
        Se forman los coeficientes del polinomio denominador
        de la interpolante
      .....
        For r:=1 to n[t] do
          b[r]:=Incognita[r];
        b[0]:=1;
        For s:=1 to n[t] do
          b[0]:=-b[s]*Pot(s,Nodo[0])+b[0];
        End;
      End;
    End;

```

```

If n[t]=0 then
  b[0]:=1;
.....
  Se forman los coeficientes del polinomio numerador
  de la interpolante racional
.....
For r:=0 to m[t] do
  a[r]:=Incongnita[m[t]-r+1];
.....
Evaluación de la interpolante racional y graficaciOn en pantalla
.....
X1:=Xmin;
Eva(m[t],a,X1,Evaluacion1);
Eva(n[t],b,X1,Evaluacion2);
If Abs(Evaluacion2)>MinimoReal then
  Y1:=Evaluacion1/Evaluacion2;
For r:=1 to 200 do
  Begin
    X2:=Xmin+(Xmax-Xmin)*r/200;
    Eva(m[t],a,X2,Evaluacion1);
    Eva(n[t],b,X2,Evaluacion2);
    If Abs(Evaluacion2)>MinimoReal then
      Y2:=Evaluacion1/Evaluacion2;
    Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
    Barras(t,r);
    X1:=X2;
    Y1:=Y2;
  End;
End;
End.
End.

```

Como se observa a través de la unidad ALGO4.PAS, para para construir diferentes interpolantes, es necesario resolver diferentes sistemas de ecuaciones (4.2). También es necesario hacer la siguiente observación, una limitación de la unidad ALGO4.PAS aparece porque la condición $q_0(x_0) = 1$, nosotros la

traducimos en que

$$b_0 = 1 - \sum_{i=1}^m b_i x_0^i ,$$

sin embargo puede suceder que el sistema (4.2) generado por esta condición sea inconsistente. A pesar de esto no debe pensarse que no existe la interpolante racional. Cuando el sistema es inconsistente, PADE lo identifica y automáticamente se termina la sesión. También es necesario mostrar como crece el tamaño del sistema (4,2), al aumentar el orden de la interpolante

$m+n$	TAMAÑO DEL SISTEMA
4	5x5
6	7x7
8	9x9
10	11x11
14	15x15

El programa PADE sólo puede construir interpolantes racionales de a lo más orden (19,19). Se ha visto a través de ejemplos, que no es necesario considerar interpolantes racionales $r_{m,n}$, tales que $m+n+1$ sea grande para tener buenas

aproximaciones. El ejemplo (4.1) ilustra muy bien esta situación. Cuando a mi me plantearon el problema de utilizar la teoría de Padé para obtener una interpolante que proporcionara tan buenas estimaciones de la función de error, como la función (4.4), se pensaba que era necesario construir una función racional de orden (96,96). Sin embargo no fue así, a lo más necesitamos de una interpolante racional de orden (0,6).

Ejemplo 4.1. Una primer aplicación de la interpolación aparece, en forma natural, cuando contamos únicamente con una muestra de valores de alguna función f y queremos una interpolante para estimar los valores de f . Esta situación es común en ciencias como física, química, economía, etc. En matemáticas aparece, por ejemplo, cuando se aproxima la solución de una ecuación diferencial en un conjunto finito de puntos, el problema es que si uno quiere tener una estimación en algún otro punto, se hace necesario introducir el tema de interpolación. Obviamente el modelo debe ser fácil de construir, evaluar, derivar, integrar y proporcionar estimaciones de f bastante buenas. Para este tipo de problemas las interpolantes racionales son una opción. ¿Por qué utilizar estas y no otras, como interpolación polinomial, trigonométrica, polinomial mini-máx o inclusive mini-máx racional ? No es fácil de explicar, realmente uno se guía por

varios cosas : que tan buenas son la estimaciones y el trabajo que necesitamos para construir la aproximante. Lo que si se puede decir, para el caso de las interpolantes racionales, es que si f posee asíntotas verticales u horizontales las interpolantes han demostrado, a través de ejemplos, que dan buenas estimaciones del valor de f . También, ellas tienden a uniformizar el error de aproximación dentro del intervalo donde se localizan los nodos, a diferencia de los polinomios de Lagrange que tienden a oscilar fuertemente. Comparandolas, ahora con las aproximaciones mini-máx, resulta que aunque las mini-máx dan más uniformidad del error, obtenerlas es extremadamente difícil. Por estos comentarios que hemos hecho no debe pensarse que cualquier otro tipo de aproximación, que no sea del tipo Padé no debe usarse, no, no es así, simplemente se han mostrado las ventajas y desventajas que ofrecen cada método.

La función de error real

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (4.3)$$

tiene una asíntota horizontal en $1/2$. Nos preguntamos si las

interpolantes racionales podrían sernos útiles, para evaluar a Φ .

En [4] se propone la función

$$\Phi^*(x) = 1 - \left[1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 \right]^{-16} \quad (4.4)$$

en donde

$$\begin{aligned} a_1 &= 0.0705230784, & a_2 &= 0.0422820123, & a_3 &= 0.0092705272 \\ a_4 &= 0.0001520143, & a_5 &= 0.0002765672, & a_6 &= 0.0000430638, \end{aligned}$$

como una aproximante a (4.3), donde el máximo error de aproximación en el intervalo $[0, \infty)$ es de 0.0000003. Como aplicación de las interpolantes racionales, construiremos una que nos ayude a evaluar a Φ , "casí" tan bien como (4.4).

Antes de empezar una observación. En lugar de considerar la función Φ , interpolaremos la función

$$\psi(x) = (1 - \Phi(x))^{1/16} \quad (4.5)$$

Y utilizaremos la siguiente idea intuitiva, si $r_{0,6}$ es una interpolante racional que aproxima a ψ , entonces

$$\rho(x) = 1 - (r_{0,6}(x))^{16} \quad (4.6)$$

es una interpolante de ϕ .

Para construir la interpolante racional $r_{0,6}$, necesitamos siete nodos y los valores de ψ en los nodos. Supongamos que son los de la tabla 4.1. Utilizando el programa PADE es fácil de obtener los coeficientes de los polinomios que forman a $r_{0,6}$ y algunas aproximaciones de ϕ en el intervalo $[0,2]$, generadas por (4.6). Los resultados aparecen en las tablas 4.2. y 4.3.

TABLA 4.1.

Valores de la función ψ	
x_i	$\psi(x_i)$
0.0	1.0000000
0.3	0.9754056
0.5	0.9551010
0.8	0.9187886
1.0	0.8908312
1.3	0.8437586
1.6	0.7913459

Tabla 4.2.

Coefficientes de los polinomios

de la interpolante racional $r_{0,6}$

$b[1]= 7.0526304851E-02$
 $b[2]= 4.2255109946E-02$
 $b[3]= 9.3511798024E-03$
 $b[4]= 4.2853502513E-05$
 $b[5]= 3.4398988871E-04$
 $b[6]= 2.7684292737E-05$
 $b[0]= 1.0000000000E+00$
 $a[0]= 1.0000000000E+00$

Tabla 4.3.

Estimaciones de la función de error

y_i	$\rho(y_i)$	Valor Exacto*
0.0000000000E+00	0.0000000000E+00	0.00000 00000
1.0000000000E-01	1.1246447042E-01	0.11246 29160
2.0000000000E-01	2.2270348105E-01	0.22270 25892
3.0000000000E-01	3.2862693426E-01	0.32862 67595
4.0000000000E-01	4.2839224374E-01	0.42839 23550
5.0000000000E-01	5.2049979585E-01	0.52049 98778
6.0000000000E-01	6.0385613174E-01	0.60385 60908
7.0000000000E-01	6.7780130486E-01	0.67780 11938
8.0000000000E-01	7.4210105368E-01	0.74210 09647
9.0000000000E-01	7.9690822279E-01	0.79690 82124
1.0000000000E+00	8.4270073039E-01	0.84270 07929
1.1000000000E+00	8.8020498729E-01	0.88020 50696
1.2000000000E+00	9.1031393944E-01	0.91031 39782
1.3000000000E+00	9.3400798361E-01	0.93400 79449
1.4000000000E+00	9.5228521626E-01	0.95228 51198
1.5000000000E+00	9.6610522604E-01	0.96610 51465
1.6000000000E+00	9.7634833573E-01	0.97634 83833
1.7000000000E+00	9.8379016981E-01	0.98379 04586
1.8000000000E+00	9.8908988613E-01	0.98909 05016
1.9000000000E+00	9.9278945230E-01	0.99279 04292
2.0000000000E+00	9.9532095025E-01	0.99532 22650

* Tomados de [1]

ii Realmente resulta sorprendente que la interpolante racional haga esas estimaciones !! ■

4.4. Fórmulas para estimar el error

Para saber si las aproximaciones generadas por una interpolante racional son buenas, es necesario tener una "buena" fórmula para medir el error. Lo ideal sería contar con un procedimiento, fácil de aplicar, que nos dijera exactamente el valor del error absoluto o relativo de las estimaciones. Sin embargo, una fórmula con estas características no es fácil de encontrar. En esta sección se presentan algunas ideas para estimar el error en interpolación racional

$$e(x) = f(x) - r_{m,n}(x) . \quad (4.7)$$

Un primer procedimiento que se nos ocurre para estimar el valor máximo de (4.7), para f de \mathbb{R} en \mathbb{R} , consiste en calcular los extremos de $e(x)$ por medio de cálculo diferencial.

Un segundo procedimiento para calcular $e(x)$, se obtiene de repetir la demostración para la fórmula de error en interpolación de Lagrange, a las interpolantes racionales.

Proposición 4.2. Si $I \subseteq \mathbb{R}$ es un intervalo que contiene todos los puntos de interpolación x_0, x_1, \dots, x_{m+n} . Si además f y $r_{m,n}$ son de clase C^{m+n+1} en I , entonces

$$(f - r_{m,n})(x) = \frac{1}{(m+n+1)!} (f - r_{m,n})^{(m+n+1)}(y_x) \prod_{i=0}^{m+n} (x - x_i) \quad \blacksquare \quad (4.8)$$

La utilización de esta fórmula es restringida, pues la estimación de la derivada que involucra al número y_x no es fácil, a menos de que el intervalo I sea pequeño y de que las derivadas sean fácil de evaluar.

Un tercer procedimiento se basa en escribir el problema de interpolación (4.2) en términos de los polinomios de Newton.

Definición 4.3. Sean $(x_i)_{i=0}^{\infty}$ una sucesión en \mathbb{C} . Se define el polinomio de Newton de grado n como

$$B_n(x) = \begin{cases} 1 & \text{si } n = 0 \\ (x-x_0)(x-x_1) \dots (x-x_{n-1}) & \text{si } n > 0 \end{cases} \quad \blacksquare$$

Proposición 4.3. Sean f una función y $(x_i)_{i=0}^{\infty}$ una

colección de puntos diferentes en algún intervalo I . Se cumple

$$\left(f - \frac{p}{q} \right) (x) = \frac{1}{q(x)} \sum_{i=m+n+1}^{\infty} d_i B_i(x) \quad (4.9)$$

con $d_i = c_{0,i} b_0 + c_{1,i} b_1 + \dots + c_{n,i} b_n$ y

$$c_{j,i} = f[x_j, \dots, x_i]. \quad \blacksquare$$

Con la serie convergente (4.9) sólo se puede estimar el error a través de las sumas parciales. Además, para calcular los coeficientes d_i es necesario contar con los valores de f en otros puntos distintos a los nodos. La demostración de (4.9) se da en la sección (4.9.3).

Podemos aplicar (4.9) para estimar el error por medio de la suma parcial de orden uno, para $e(x) = \psi(x) - r_{0,6}(x)$ del ejemplo (4.1). Para esto añadimos el nodo $x_7 = 0.9$ y $y_7 = 0.7969082124$. Con esta información se da el siguiente estimador

$$e(x) \approx \frac{d_7 B_7(x)}{q(x)} \quad (4.10)$$

En la tabla 4.4. se muestran algunas de las estimaciones

Tabla 4.4.

x_j	$e(x)$
0.0000000000E+00	0.0000000000E+00
1.0000000000E-01	-1.7470147508E-31
2.0000000000E-01	-8.4704561772E-32
3.0000000000E-01	0.0000000000E+00
4.0000000000E-01	1.9424307372E-32
5.0000000000E-01	0.0000000000E+00
6.0000000000E-01	-1.8456878865E-32
7.0000000000E-01	-1.6398068595E-32
8.0000000000E-01	0.0000000000E+00
9.0000000000E-01	1.0621221720E-32
1.0000000000E+00	0.0000000000E+00
1.1000000000E+00	-2.6914698521E-32
1.2000000000E+00	-4.0369094145E-32
1.3000000000E+00	0.0000000000E+00
1.4000000000E+00	1.0672156833E-31
1.5000000000E+00	1.9784986700E-31
1.6000000000E+00	0.0000000000E+00
1.7000000000E+00	-1.0791997079E-30
1.8000000000E+00	-4.1076492737E-30
1.9000000000E+00	-1.0810959890E-29
2.0000000000E+00	-2.3775312206E-29

Aunque los errores dados en la tabla 4.4. son muy pequeños, deben ser tomados con reserva. Por el valor que tienen los errores, uno esperaría una mejor aproximación a la función ψ , sin embargo esto no es así. Ante esta situación es necesario hacer una observación, los errores que se obtienen por hacer los cálculos en una aritmética finita o de computadora no han sido tomados en cuenta. Todos los cálculos que hasta el momento hemos hecho, se realizaron con precisión simple en una computadora. Aunque saber como se propagan los errores en un algoritmo es

importante, no abordaremos este tema.

4.5. Propiedades básicas.

Una dificultad que surge al usar el algoritmo de Gauss de la sección (4.3), es que, como es difícil trabajar con la fórmula del error (4.7), no se sabe generalmente el orden de la interpolante racional necesario para lograr la precisión deseada hasta que los cálculos han sido completados. Al examinar el trabajo realizado para calcular una interpolante racional, no se ve como este pueda ayudarnos a reducir el trabajo necesario para calcular otra. En la sección (4.6) se darán unos algoritmos en la que los cálculos realizados para obtener una interpolante racional son aprovechados para construir otras. Sin embargo estos procedimientos no son aplicables si la interpolante racional se repite. En esta sección se estudian algunos resultados, que nos ayudarán a saber cuando hay repeticiones en la tabla de Padé.

Aunque se dice que $r_{m,n}$ interpola a f , puede suceder que $r_{m,n}$ no satisfaga (4.1) en algunos de los nodos, a pesar de satisfacer (4.2). Los puntos en que (4.1) no se cumple se llaman *puntos no alcanzables por $r_{m,n}$* . El siguiente ejemplo ilustra esta situación.

Ejemplo 4.2. Para la función f definida por la tabla

x_i	0	1	2
f_i	0	3	3

se cumple que el punto $(0,0)$ es no alcanzable por la interpolante racional $r_{1,1}$. El sistema (4.2) tiene como solución $p(x) = 3x$ y $q(x) = x$, por lo tanto $r_{m,n}(x) = 3$. Este ejemplo muestra que los puntos no alcanzables están íntimamente relacionados con las raíces del denominador de la solución de (3.2). De hecho, un punto es no alcanzable si, y sólo si es raíz de $q(x)$, para (p,q) solución de (4.2). Con el ejemplo también se observa el fenómeno de la repetición de interpolantes, pues $r_{1,1}$ es igual a $r_{0,0}$. ■

La notación que se utilizará para denotar los grados de los polinomios p_0 y q_0 , donde p_0/q_0 es la interpolante racional de orden (m,n) irreducible, es m^* y n^* respectivamente.

Proposición 4.4. Si la interpolante racional de orden (m,n) para f es $r_{m,n} = p_0/q_0$, entonces existe un entero s con $0 \leq s \leq \min(m-m^*, n-n^*)$ y s puntos y_1, y_2, \dots, y_s del conjunto de nodos $\{x_0, x_1, \dots, x_{m+n}\}$, tales que

$$p(x) = p_0(x) \prod_{i=1}^s (x - y_i) \quad \text{y} \quad q(x) = q_0(x) \prod_{i=1}^s (x - y_i)$$

satisfacen (4.2).

Demostración. Sea (p_1, q_1) una solución de (4.2) de orden (m, n) . Entonces $(fq_1^s(x_i) - p_1(x_i)) = 0$, para $i=0, 1, \dots, m+n$. Se observa que si $q_1(x_i) = 0$ entonces $p_1(x_i) = 0$. Sea

$$\{y_1, y_2, \dots, y_s\} \tag{4.11}$$

el conjunto de los ceros de q_1 que además son nodos y sea el polinomio $t(x) = \prod_{i=1}^s (x - y_i)$. Como la parte reducible de p_1/q_1 consta de un polinomio $v(x)$, cuyas raíces no son nodos, multiplicado por el polinomio $t(x)$. Entonces

$$p_1(x) = v(x) t(x) p_0(x) \quad \text{y} \quad q_1(x) = v(x) t(x) q_0(x) \tag{4.12}$$

De (4.12) es fácil demostrar que el entero s satisface las desigualdades $s \leq m - m^*$ y $s \leq n - n^*$, las cuales implican que

$$s \leq \min(m - m^*, n - n^*) \tag{4.13}$$

Ahora es fácil demostrar que los polinomios que buscamos son

$$p(x) = t(x) p_0(x) \quad \text{y} \quad q(x) = t(x) q_0(x)$$

Para demostrarlo se observa que si x_i pertenece al conjunto (4.11) entonces $(fq - p)(x_i) = 0$, porque $t(x_i) = 0$. Ahora si x_i no pertenece al conjunto (4.11), demostramos que $(fq - p)(x_i) = 0$ multiplicando y dividiendo por $v(x_i)$

$$(fq - p)(x_i) = \frac{(fqv - pv)(x_i)}{v(x_i)} = \frac{(fq_1 - p_1)(x_i)}{v(x_i)} = \frac{0}{v(x_i)} = 0$$

El único detalle que se debe aclarar es que $v(x_i) \neq 0$, pero esto es cierto porque $q_1(x_i) = v(x_i) t(x_i) q_0(x_i)$ y x_i no es un cero de q_1 . ■

Antes de continuar hacemos una observación que necesitaremos mas adelante. Multiplicando y dividiendo por $v(x_i) t(x_i)$ la ecuación $(fq_0 - p_0)(x_i)$, se demuestra que

$$(fq_0 - p_0)(x_i) = 0 \quad (4.14)$$

si x_i no pertenece al conjunto (4.11). Esto significa que (4.14)

se satisface en $m + n + 1 - s$ nodos. Utilizando la desigualdad (4.13) y $n \geq n^*$, se deduce que

$$m + n + 1 - s \geq m^* + n^* + 1$$

Luego entonces existe un entero $t \geq 0$ tal que

$$m + n + 1 - s = m^* + n^* + t + 1 \quad (4.15)$$

El número t de (4.15) juega un papel importante para la identificación de interpolantes racionales repetidas.

Proposición 4.5. Sea $r_{m,n} = p_0/q_0$ y supóngase que

- (a) $(fq_0 - p_0)(x_i) = 0 \quad i = 0, 1, \dots, m^* + n^* + t$
 (b) $(fq_0 - p_0)(x_i) \neq 0 \quad i = m^* + n^* + t + 1, \dots, m + n$

Entonces (c) Para k y l que satisfagan $m^* \leq k \leq m^* + t$ y $n^* \leq l \leq n^* + t$ se cumple que $r_{k,l} = r_{m,n}$.
 (d) $m \leq m^* + t$ y $n \leq n^* + t$

Demostración. Sean k y l tales que $m^* \leq k \leq m^* + t$ y $n^* \leq l \leq n^* + t$. Si $s = \min(k - m^*, l - n^*)$ y

$$t(x) = \prod_{i=m^* + n^* + t + 1}^{m^* + n^* + t + s} (x - x_i)$$

entonces $p(x) = p_0(x) t(x)$ y $q(x) = q_0(x) t(x)$ satisfacen

$$(fq - p)(x_i) = 0 \quad i = 0, 1, \dots, m^* + n^* + t + s,$$

donde se ha utilizado la hipótesis (a). Es fácil comprobar ahora que $k + l \leq m^* + n^* + t + s$, por lo tanto

$$(fq - p)(x_i) = 0 \quad i = 0, 1, \dots, k+l.$$

Así (p, q) es solución del problema de interpolación de orden (m, n) y (k, l) . Con esto terminamos la demostración de (c).

Para demostrar (d), utilizamos la desigualdad (4.15) para deducir que $m + n \leq m^* + n^* + t + s$, combinando esta desigualdad con la (4.13) se deduce el resultado. ■

Un comentario inevitable respecto a las dos proposiciones anteriores se debe hacer. Como se observa, es necesario contar con la fracción irreducible p_0/q_0 para poder

investigar las repeticiones en la tabla de Padé, esto en la práctica no es fácil de obtener. En este sentido sería deseable contar con un procedimiento más fácil de aplicar.

También es importante reflexionar sobre las hipótesis de la proposición (4.4). Obsérvese que la repetición en bloques en la tabla de Padé aparece, sólo si los nodos están ordenados de forma que los puntos x_i tales que $(fq_0 - p_0)(x_i) = 0$ aparecen al principio de la lista. Los siguientes ejemplos ilustran estos aspectos.

Ejemplo 4.3. Para la función

x_i	0	1	2	3	4	5
f_i	1	1	5/3	5/2	17/5	13/5

si se calcula $r_{3,2}$ se obtiene

$$r_{3,2}(x) = \frac{1 + x^2}{1 + x} = \frac{p_0(x)}{q_0(x)}$$

Para esta interpolante racional se cumple $(fq_0 - p_0)(x_i) = 0$ para $i = 0, 1, 2, 3, 4$, mientras que $(fq_0 - p_0)(x_5) \neq 0$. Para este ejemplo la variable t de la proposición (4.4) es 1. Así que

tenemos el siguiente bloque de interpolantes racionales iguales

$$\begin{array}{cc} r_{2,1} & r_{2,2} \\ r_{3,1} & r_{3,2} \end{array}$$

■ □

El siguiente ejemplo muestra que la repetición en bloques no se da si los nodos no son ordenados como lo indica la hipótesis de (4.4).

Ejemplo 4.4. Para la función

x_i	0	1	2	3
f_i	2	3/2	4/5	1/2

si se calcula $r_{2,1}$ se obtiene $r_{2,1}(x) = 2 - x/2$, que coincide con $r_{1,0}$. Sin embargo, $r_{2,0} \neq r_{2,1}$ y $r_{1,1} \neq r_{2,1}$.

Se puede comprobar en este ejemplo que

$$\begin{array}{ll} f(x_0)q(x_0) - p(x_0) \neq 0 & f(x_1)q(x_1) - p(x_1) = 0 \\ f(x_2)q(x_2) - p(x_2) \neq 0 & f(x_3)q(x_3) - p(x_3) = 0 \end{array}$$

Ahora bien, si los nodos se ordenen en la forma (z_0, z_1, z_2, z_3)

con $z_0 = x_1$, $z_1 = x_3$, $z_2 = x_0$ y $z_3 = x_2$, entonces por la proposición (4.4) se afirma que ahora sí existe un bloque de interpolantes iguales. ■

4.6. Normalidad

El concepto de normalidad es igual que en aproximación racional a saber: $r_{m,n}$ es normal si aparece sólo una vez en la tabla de Padé. Los siguientes resultados dan condiciones necesarias y suficientes para que una interpolante sea normal.

Proposición 4.6. Si $r_{m,n} = p_0/q_0$ es normal y

$$(fq_0 - p_0)(x_i) = 0 \quad \text{para } i = 0, \dots, m^* + n^*$$

entonces

a) $m = m^*$ y $n = n^*$

b) $(fq_0 - p_0)(x_i) \neq 0$ para $i = m+n+1, m+n+2$

Demostración. Como p_0/q_0 es la interpolante racional de orden (m,n) y también solución del problema de interpolación racional de orden (m^*, n^*) , entonces $r_{m^*, n^*} = r_{m,n}$, que por la normalidad implica que $m = m^*$ y $n = n^*$.

Ahora supóngase que $(fq_0 - p_0)(x_{m+n+1}) = 0$ entonces (p_0, q_0) es solución del problema de interpolación racional de orden (m, n) , $(m+1, n)$ y $(m, n+1)$. Esto implica que $r_{m, n} = r_{m+1, n} = r_{m, n+1}$, lo cual contradice la normalidad.

Si $(fq_0 - p_0)(x_{m+n+2}) = 0$ entonces los polinomios

$$p(x) = (x - x_{m+n+1}) p_0(x) \quad \text{y} \quad q(x) = (x - x_{m+n+1}) q_0(x)$$

satisfacen $(fq - p)(x_i) = 0$ para $i = 0, \dots, m+n+2$. Por lo tanto $r_{m, n} = r_{m+1, n+1}$, lo que contradice la normalidad. ■

Las condiciones (a) y (b) de la proposición (4.5) no son suficientes para garantizar la normalidad de $r_{m, n}$, el siguiente ejemplo lo muestra.

Ejemplo 4.5. Para la función

x_i	0	1	2	3	4	5	6	...
f_i	0	1	3	4	4	5	6	...

la interpolante $r_{1,0}(x) = x$ satisface las condiciones (a) y (b) de la proposición (4.5). Sin embargo esta interpolante no es

normal, pues $r_{1,0} = r_{3,2}$. Que esta igualdad es cierta se puede comprobar observando que $p(x) = x(x-2)(x-3)$ y $q(x) = (x-1)(x-3)$ satisfacen $(fq - p)(x_i) = 0$, para $i = 0, 1, 2, 3, 4, 5$. Inclusive la interpolante de orden $(1,0)$ es igual a la interpolante $r_{k,i}$ para $k \geq 3$ y $l \geq 2$. ■

El siguiente resultado da condiciones suficientes para la normalidad de $r_{m,n}$.

Proposición 4.7. Si $r_{m,n} = p_0/q_0$, con $m = m^*$, $n = n^*$ y $(fq_0 - p_0)(x_i) = 0$ para a lo más en $m+n+1$ puntos de la sucesión $(x_i)_0^\infty$ entonces $r_{m,n}$ es normal.

Demostración. Supóngase que $r_{m,n}$ no es normal. En otras palabras, $r_{m,n} = r_{k,l}$ para $k \geq m$ o $l \geq n$, donde por lo menos una de las desigualdades es estricta. Utilizando la proposición (3.2) existe un entero s con $0 \leq s \leq \min(k-m, l-n)$ y s nodos $\{y_1, y_2, \dots, y_s\}$, tales que los polinomios

$$p(x) = p_0(x) \prod_{i=1}^s (x - y_i) \quad \text{y} \quad q(x) = q_0(x) \prod_{i=1}^s (x - y_i)$$

satisfacen $(fq - p)(x_i) = 0$ para $i = 0, \dots, k+1$. De esto se

deduce que $(fq_0 - p_0)(x_i) = 0$ para $k + l + 1 - s$ puntos. Como $-s \geq m - k$ y $l > n$ entonces $k + l + 1 - s > m + n + 1$, lo que contradice la hipótesis de que $(fq_0 - p_0)(x_i) = 0$ para a lo más $m + n + 1$ puntos. ■

En este tema de normalidad, también es necesario marcar la falta de un criterio sencillo para comprobar que una interpolante racional es normal. Si observamos la proposición (4.6), ésta funciona si podemos comprobar que los polinomios que forman la interpolante son irreducibles. Esto no es fácil en general. Como probar la normalidad de una interpolante no es fácil, esto nos limita para saber si los algoritmos que veremos en la siguiente sección se pueden aplicar para contruir una interpolante racional.

4.7. Algoritmo cociente-diferencia

Para calcular interpolantes racionales se ha supuesto que se cuenta con algun algoritmo de solución de sistemas de ecuaciones lineales, como eliminación gaussiana. En efecto hemos formulado el problema de manera que equivale a resolver el sistema de ecuaciones lineales

$$f(x_i) q(x_i) - p(x_i) = 0 \quad , \quad i = 0, 1, \dots, m+n$$

$$q(x_0) = 1$$

En esta sección se estudiarán los algoritmos cociente-diferencia de Rutishauser [7] y epsilon de Winn [11]. Estos dos algoritmos tienen la ventaja de aprovechar los cálculos que se han hecho para obtener una interpolante, para encontrar otras. El único inconveniente que tienen, es que solamente se pueden aplicar si la interpolante racional es normal.

El siguiente resultado es básico para desarrollar el algoritmo cociente-diferencia.

Proposición 4.8. Si $r_{m,n} = p_1/q_1$ y $r_{m+k,n+l} = p_2/q_2$ con $k, l \geq 0$, entonces existe un polinomio $v(x)$ tal que

$$\partial v \leq \max(k-1, l-1) \quad \text{y} \quad (p_1 q_2 - p_2 q_1)(x) = v(x) B_{m+n+1}(x)$$

donde $B_{m+n+1}(x) = \prod_{i=0}^{m+n} (x-x_i)$.

Demostración. Se sabe que

$$(f q_1 - p_1)(x_i) = 0 \quad , \quad i = 0, \dots, m+n$$

$$(fq_2 - p_2)(x_i) = 0 \quad , \quad i = 0, \dots, m+n+k+1$$

Por lo tanto

$$(p_1q_2 - p_2q_1)(x_i) = [(fq_2 - p_2)q_1](x_i) - [(fq_1 - p_1)q_2](x_i) = 0$$

para $i = 0, 1, \dots, m+n$. Esto implica la existencia de un polinomio $v(x)$, tal que $(p_1q_2 - p_2q_1)(x) = v(x) B_{m+n+1}(x)$. Ahora como $\partial(p_1q_2 - p_2q_1) \leq \max(m+n+k, m+n+1)$ y $\partial B_{m+n+1} = m+n+1$ entonces debe tenerse que $\partial v \leq \max(k-1, l-1)$. ■

El algoritmo QD, se obtiene pidiendo que el n-ésimo elemento de la escalera descendente (4.16) sea igual a la n-ésima convergente de la fracción continua (4.17). Esta idea es la nos permite relacionar fracciones continuas con interpolantes racionales

Escalera descendente T_k

$$T_k : \begin{array}{ccccccc} r_{k,0} & & & & & & \\ & r_{k+1,0} & & & & & \\ & & r_{k+1,1} & & & & \\ & & & r_{k+2,1} & & & \\ & & & & r_{k+2,2} & \dots & \end{array} \quad (4.16)$$

Fracción continua g_k

$$\begin{aligned}
 g_k(x) = & d_0 + d_1(x-x_0) + \dots + d_k(x-x_0)(x-x_1) \dots (x-x_{k-1}) + \\
 & + \frac{d_{k+1}(x-x_0) \dots (x-x_k)}{b_1} + \frac{d_{k+2}(x-x_0) \dots (x-x_{k+1})}{b_1 b_2} + \dots
 \end{aligned}
 \tag{4.17}$$

Proposición 4.9. Si cada tres elementos consecutivos de T_k son diferentes, entonces existe una fracción continua del tipo (4.17), tal que $a_i(x-x_{k+i}) \neq 0$, $b_i \neq 0$ y su n -ésima convergente coincide con el n -ésimo elemento de T_k .

Demostración. Se utiliza la proposición (1.4), el hecho de que el n -ésimo elemento de T_k es $r_{k + \left[\frac{n+1}{2} \right], \left[\frac{n}{2} \right]}$

y la n -ésima convergente de g_k es $C_n = \frac{P_n}{Q_n}$. Como

$$C_n = \frac{P_n}{Q_n} = r_{k + \left[\frac{n+1}{2} \right], \left[\frac{n}{2} \right]}$$

para $n = 0, 1, \dots$, la fracción continua g_k tiene que ser de la forma

$$P_0 + \frac{P_1 - P_0}{I} + \sum_{I=1}^{\infty} \left[\frac{\frac{P_i Q_{i+1} - P_{i+1} Q_i}{P_i Q_{i-1} - P_{i-1} Q_i}}{\frac{P_{i+1} Q_{i-1} - P_{i-1} Q_{i+1}}{P_i Q_{i-1} - P_{i-1} Q_i}} \right] \quad (4.18)$$

Antes de seguir adelante con la demostración hacemos una observación. La condición de que tres elementos consecutivos en T_k sean diferentes es necesaria, ya que de otra forma la fracción continua (4.18) podría tener un fracción del tipo $0/0$. El programa PADE no está capacitado para saber si una interpolante racional es normal, y por lo tanto si el usuario no tiene la precaución de ver si las interpolantes que desea graficar se repiten, aparecerán gráficas extrañas. Continuamos con la demostración.

Ahora se calcularán los numeradores y denominadores parciales de (3.5). Como $C_0 = P_0 = r_{k,0}$ entonces P_0 es el polinomio de Lagrange que interpola a la función $f(x)$ en $(x_i)_{i=0}^{k-1}$. A continuación, se observa que $C_1 = P_1 = r_{k+1,0}$, lo cual indica que P_1 es el polinomio de Lagrange que interpola en $(x_i)_{i=0}^k$. Para calcular el resto de los coeficientes de la fracción continua observamos primero que

$$\frac{P_{i-1}}{Q_{i-1}} = r_{k+\left[\frac{i}{2}\right], \left[\frac{i-1}{2}\right]}, \quad \frac{P_i}{Q_i} = r_{k+\left[\frac{i+1}{2}\right], \left[\frac{i}{2}\right]} \quad \text{y} \quad \frac{P_{i+1}}{Q_{i+1}} = r_{k+\left[\frac{i+2}{2}\right], \left[\frac{i+1}{2}\right]}$$

por lo que utilizando la proposición (4.8) resulta

$$\frac{P_i Q_{i+1} - P_{i+1} Q_i}{P_i Q_{i-1} - P_{i-1} Q_i} = a_i (x - x_{i+k}) \quad , \quad i \geq 1$$

En efecto

$$P_i Q_{i+1} - P_{i+1} Q_i = v_1(x) \prod_{i=0}^{k+\left[\frac{i+1}{2}\right]+\left[\frac{i}{2}\right]} (x - x_i) \quad , \quad \partial v_1 = 0$$

y

$$P_i Q_{i-1} - P_{i-1} Q_i = v_2(x) \prod_{i=0}^{k+\left[\frac{i-1}{2}\right]+\left[\frac{i}{2}\right]} (x - x_i) \quad , \quad \partial v_2 = 0$$

La simplificación de los denominadores parciales se hace de la misma forma. ■

Aunque la proposición (4.9) nos permite escribir las

interpolantes racionales de la escalera T_k como fracciones continuas, queda por decir cómo calcular los coeficientes a 's y b 's.

Proposición 4.10. La fracción continua (4.19) se puede escribir como

$$\begin{aligned}
 g_k(x) = & c_0 + \sum_{i=1}^k c_i \frac{(x-x_0) \cdots (x-x_{i-1})}{(x-x_i)} + \frac{c_{k+1}(x-x_0) \cdots (x-x_k)}{1} + \\
 & + \frac{-q_1^{(k+1)}(x-x_{k+1})}{1 + q_1^{(k+1)}(x_0-x_{k+1})} + \frac{-e_1^{(k+1)}(x-x_{k+2})}{1 + e_1^{(k+1)}(x_0-x_{k+2})} + \\
 & + \frac{-q_2^{(k+1)}(x-x_{k+3})}{1 + q_2^{(k+1)}(x_0-x_{k+3})} + \frac{-e_2^{(k+1)}(x-x_{k+4})}{1 + e_2^{(k+1)}(x_0-x_{k+4})} + \dots \quad (4.19)
 \end{aligned}$$

donde los e 's y q 's se pueden arreglar en una tabla en la que los sub-índices indican una columna y los super-índices una diagonal,

Tabla QD

$e_0^{(1)}$				
$e_0^{(2)}$	$q_1^{(1)}$			
$e_0^{(3)}$	$q_1^{(2)}$	$e_1^{(1)}$		
$e_0^{(4)}$	$q_1^{(3)}$	$e_1^{(2)}$	$q_2^{(1)}$	
$e_0^{(5)}$	$q_1^{(4)}$	$e_1^{(3)}$	$q_2^{(2)}$...

Los elementos de la tabla se calculan utilizando las condiciones iniciales

$$e_0^{(k)} = 0, \quad q_1^{(k)} = \frac{f[x_0, \dots, x_{k+1}]}{f[x_1, \dots, x_{k+1}]}, \quad k \geq 1 \quad (4.20)$$

y las fórmulas

$$e_1^{(k)} = \frac{q_1^{(k+1)} - q_1^{(k)} + e_{l-1}^{(k+1)} \left[1 + q_1^{(k+1)}(x_0 - x_{k+2l-1}) \right]}{1 + q_1^{(k)}(x_0 - x_{k+2l-1})} \quad (4.21)$$

$$q_{l+1}^{(k)} = \frac{e_1^{(k+1)} q_1^{(k+1)} \left[1 + e_1^{(k)}(x_0 - x_{k+2l}) \right]}{e_1^{(k)} \left[1 + q_1^{(k+1)}(x_0 - x_{k+2l-1}) \right] + e_1^{(k+1)} (e_1^{(k)} - q_1^{(k+1)}) (x_0 - x_{k+2l+1})} \quad (4.22)$$

Demostración. Utilizando la fórmula de recurrencia para el denominador de la fracción continua $g_k(x)$ y la normalización de las interpolantes racionales se obtiene

$$\begin{aligned} 1 = Q_n(x_0) &= b_n Q_{n-1}(x_0) + a_n (x_0 - x_{k+n}) Q_{n-2}(x_0) = \\ &= b_n + a_n (x_0 - x_{k+n}) \end{aligned}$$

Por lo tanto $b_n = 1 - a_n (x_0 - x_{k+n})$. Ahora, si se escribe

$$a_{2i-1} = -q_i^{(k+1)} \quad \text{y} \quad a_{2i} = -e_i^{(k+1)}$$

queda comprobada la fórmula (3.6).

Para demostrar las fórmulas (4.21) y (4.22), se observa que por la relación que guardan g_k y g_{k-1} con las escaleras T_k y T_{k-1} respectivamente, las convergentes de la parte par de g_k son iguales a las convergentes de la parte impar de g_{k-1} , así que sus numeradores y denominadores tienen la misma forma. Utilizando las fórmulas (1.16) y (1.17) queda demostrada la proposición. ■

El algoritmo de Rutishauser es utilizado por PADE para construir y graficar interpolantes racionales. A continuación

hacemos unas observaciones que nos permiten comprender el listado de este programa.

Si queremos construir la interpolante racional $r_{m,n}$ a través del algoritmo cociente-diferencia, observamos que ella pertenece a la escalera $T_{m-n} = (r_{m-n,0}, r_{m-n+1,0}, r_{m-n+1,1}, r_{m-n+2,1}, \dots)$ y que es el $2n$ -ésimo elemento. Por lo tanto $r_{m,n}$ es igual a la $2n$ -ésima convergente C_{2n} de la fracción continua $g_{m-n}(x)$. De esto se infiere que para construir C_{2n} , necesitamos la siguiente región de la tabla qd.

Tabla 4.5.

$q_1^{(m-n+1)}$				
$q_1^{(m-n+2)}$	$e_1^{(m-n+1)}$			
$q_1^{(m-n+3)}$	$e_1^{(m-n+2)}$	$q_2^{(m-n+1)}$		
.	.	.	.	$e_{n-1}^{(m-n+1)}$
.	.	.	.	$q_n^{(m-n+1)}$
.	.	.	.	$e_{n-1}^{(m-n+2)}$
$q_1^{(m+n-3)}$	$e_1^{(m+n-3)}$	$q_2^{(m+n-3)}$		
$q_1^{(m+n-2)}$	$e_1^{(m+n-2)}$			
$q_1^{(m+n-1)}$				

Los elementos que forman la tabla (4.5) se encuentran aplicando las fórmulas (4.20), (4.21) y (4.22) por columnas de izquierda a derecha. Habiendo calculado los elementos de la diagonal $m-n+1$, estamos en posibilidad de construir $r_{m,n}$.

Como ya lo indicamos PADE grafica interpolantes racionales y la unidad ALG05.PAS es la encargada de realizar este proceso a través del algoritmo cociente-diferencia, a continuación mostramos una copia de él.

Las variables m y n son para guardar el orden de la interpolante racional, F y $Nodo$ son vectores que contienen el valor de la función y los nodos respectivamente, $DifDiv$ es un procedimiento que nos proporciona las diferencias divididas de la fórmula (4.20) a través de los vectores $DifDiv1$ y $DifDiv2$, el procedimiento $QD2$ calcula la tabla 4.5 y nos proporciona los elementos de la diagonal ($m-n+1$) a través de los vectores e y q , el procedimiento $ForFrac$ forma los numeradores y denominadores parciales de la fracción continua (4.19), el procedimiento $ConEva$ evalúa convergentes de fracciones continuas por medio del algoritmo de sustitución hacia atrás y finalmente el procedimiento $Horner$ evalúa polinomios interpolantes de Lagrange.

```

Unit   Algo5;
Begin
  If    ((n[t]>1)and(m[t]>=n[t]))    Then
    Begin
      *****
      Se calculan las diferencias divididas que se van a utilizar
      en las fórmulas (4.20) y en el denominador parcial BO de la
      fracción continua (4.19)
      *****
      DifDiv(m[t]+n[t],F,Nodo,DifDiv1,DifDiv2);
      *****
      Se calculan los elementos de la tabla (4.5)
      *****
      QD2(DifDiv1,DifDiv2,e,q,m[t],n[t],Nodo);
      X1:=Xmin;
      *****
      Se forman los numeradores y denominadores parciales de la
      fracción continua (4.19) para x1 = xmin
      *****
      ForFracCont1(Num,Den,e,q,Nodo,n[t],X1,DifDiv1,m[t],n[t]);
      *****
      Se calcula el denominador parcial BO de (4.19)
      *****
      Horner(m[t]-n[t],DifDiv1,Nodo,X1,BO);
      *****
      Se evalua la convergente de orden 2*n de (4.19)
      *****
      ConEva(BO,Num,Den,Conv,2*n[t]);
      Y1:=Conv;
      *****
      Se repiten los pasos anteriores variando x en el intervalo
      [ xmin, xmax ]
      *****
      For r:=1 to 200 Do
        Begin
          X2:=Xmin+((Xmax-Xmin)*r)/200;
          ForFracCont1(Num,Den,e,q,Nodo,n[t],X2,DifDiv1,m[t],n[t]);
          Horner(m[t]-n[t],DifDiv1,Nodo,X2,BO);
          ConEva(BO,Num,Den,Conv,2*n[t]);
          Y2:=Conv;
          Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
          X1:=X2;   Y1:=Y2;
        End;
      End;
    End;
  End;

```

```

If      ((n[t]=1)and(m[t]>=n[t]))      Then
Begin
  DifDiv(m[t]+n[t],F,Nodo,DifDiv1,DifDiv2);
  q[m[t]]:=DifDiv1[m[t]+1]/DifDiv2[m[t]+1];
  X1:=Xmin;
  If m[t]=1 then
    Num[1]:=DifDiv1[m[t]]*(x1-Nodo[0]);
  If m[t]>1 then
    Begin
      Num[1]:=DifDiv1[m[t]]*(x1-Nodo[0]);
      For s:=1 to m[t]-1 do
        Num[1]:=Num[1]*(x1-Nodo[s]);
      End;
    End;
  Den[1]:=1;
  Horner(m[t]-n[t],DifDiv1,Nodo,X1,B0);
  Num[2]:=-q[m[t]]*(x1-Nodo[m[t]]);
  Den[2]:=1+q[m[t]]*(Nodo[0]-Nodo[m[t]]);
  ConEva(B0,Num,Den,Conv,2*n[t]);
  Y1:=Conv;
  For r:=1 to 200 Do
    Begin
      X2:=Xmin+((Xmax-Xmin)*r)/200;
      If m[t]=1 then
        Num[1]:=DifDiv1[m[t]]*(x2-Nodo[0]);
      If m[t]>1 then
        Begin
          Num[1]:=DifDiv1[m[t]]*(x2-Nodo[0]);
          For s:=1 to m[t]-1 do
            Num[1]:=Num[1]*(x2-Nodo[s]);
          End;
        End;
      Den[1]:=1;
      Horner(m[t]-n[t],DifDiv1,Nodo,X2,B0);
      Num[2]:=-q[m[t]]*(x2-Nodo[m[t]]);
      Den[2]:=1+q[m[t]]*(Nodo[0]-Nodo[m[t]]);
      ConEva(B0,Num,Den,Conv,2*n[t]);
      Y2:=Conv;
      Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
      X1:=X2;
      Y1:=Y2;
    End;
  End;
End;

```

```

If n[t]=0 then
Begin
DifDiv(m[t],F,Nodo,DifDiv1,DifDiv2);
x1:=xmin;
Horner(m[t],DifDiv1,Nodo,X1,BO);
y1:=BO;
For r:=1 to 200 Do
Begin
X2:=Xmin+((Xmax-Xmin)*r)/200;
Horner(m[t],DifDiv1,Nodo,x2,BO);
y2:=BO;
Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
X1:=X2;
Y1:=Y2;
End;
End;
If ((n[t]>m[t])and(m[t]>1)) then
Begin
For s:=0 to m[t]+n[t] do
Begin
If Abs(F[s])>MinimoReal then
FRec[s]:=1/F[s];
End;
DifDiv(m[t]+n[t],FRec,Nodo,DifDiv1,DifDiv2);
QD2(DifDiv1,DifDiv2,e,q,n[t],m[t],Nodo);
X1:=Xmin;
ForFracCont1(Num,Den,e,q,Nodo,m[t],X1,DifDiv1,n[t],m[t]);
Horner(n[t]-m[t],DifDiv1,Nodo,X1,BO);
ConEva(BO,Num,Den,Conv,2*m[t]);
If Abs(Conv)>MinimoReal then
Y1:=1/Conv;
For r:=1 to 200 Do
Begin
X2:=Xmin+((Xmax-Xmin)*r)/200;
ForFracCont1(Num,Den,e,q,Nodo,m[t],X2,DifDiv1,n[t],m[t]);
Horner(n[t]-m[t],DifDiv1,Nodo,X2,BO);
ConEva(BO,Num,Den,Conv,2*m[t]);
If Abs(Y2)>MinimoReal then
Y2:=1/Conv;
Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
X1:=X2;
Y1:=Y2;
End;
End;
End;

```

```

if ((n[t]>m[t])and(m[t]=1)) then
  Begin
    For s:=0 to m[t]+n[t] do
      FRec[s]:=1/F[s];
    DifDiv(m[t]+n[t],FRec,Nodo,DifDiv1,DifDiv2);
    q[n[t]]:=DifDiv1[n[t]+1]/DifDiv2[n[t]+1];
    X1:=Xmin;
    If n[t]=1 then
      Num[1]:=DifDiv1[n[t]]*(x1-Nodo[0]);
    If n[t]>1 then
      Begin
        Num[1]:=DifDiv1[n[t]]*(x1-Nodo[0]);
        For s:=1 to n[t]-1 do
          Num[1]:=Num[1]*(x1-Nodo[s]);
        End;
      End;
    Den[1]:=1;
    Horner(n[t]-m[t],DifDiv1,Nodo,X1,B0);
    Num[2]:=-q[n[t]]*(x1-Nodo[n[t]]);
    Den[2]:=1+q[n[t]]*(Nodo[0]-Nodo[n[t]]);
    ConEva(B0,Num,Den,Conv,2*m[t]);
    If Abs(Conv)>MinimoReal then
      Y1:=1/Conv;
    End
    For r:=1 to 200 Do
      Begin
        X2:=Xmin+((Xmax-Xmin)*r)/200;
        If n[t]=1 then
          Num[1]:=DifDiv1[n[t]]*(x2-Nodo[0]);
        If n[t]>1 then
          Begin
            Num[1]:=DifDiv1[n[t]]*(x2-Nodo[0]);
            For s:=1 to n[t]-1 do
              Num[1]:=Num[1]*(x2-Nodo[s]);
            End;
          End;
        Den[1]:=1;Horner(n[t]-m[t],DifDiv1,Nodo,X2,B0);
        Num[2]:=-q[n[t]]*(x2-Nodo[n[t]]);
        Den[2]:=1+q[n[t]]*(Nodo[0]-Nodo[n[t]]);
        ConEva(B0,Num,Den,Conv,2*m[t]);
        If Abs(Conv)>MinimoReal then
          Y2:=1/Conv;
        Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
        X1:=X2;Y1:=Y2;
      End;
    End;
  End;

```

```

If m[t]=0 then
Begin
  For s:=0 to n[t] do
    If Abs(F[s])>MinimoReal then
      Begin
        FRec[s]:=1/F[s];
      End
    Else
      FRec[s]:=MaximoReal;
    DifDiv(n[t],FRec,Nodo,DifDiv1,DifDiv2);
    x1:=xmin;
    Horner(n[t],DifDiv1,Nodo,x1,B0);
    If Abs(B0)>MinimoReal then
      Begin
        y1:=1/B0;
      End
    Else
      y1:=MaximoReal;
    For r:=1 to 200 Do
      Begin
        X2:=Xmin+((Xmax-Xmin)*r)/200;
        Horner(n[t],DifDiv1,Nodo,x2,B0);
        If Abs(B0)>Minimoreal then
          Begin
            y2:=1/B0;
          End
        Else
          y2:=MaximoReal;
        Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
        X1:=X2;
        Y1:=Y2;
      End;
    End;
  End;
End;
End.

```

Antes de pasar a otra cosa, es necesario hacer la siguiente observación. Cuando $m < n$, el método que se utiliza para construir la interpolante racional de orden (m,n) es como sigue. Se busca la interpolante de orden (n,m) que interpola en

los puntos $(x_0, 1/f(x_0)), (x_1, 1/f(x_1)), \dots, (x_{m+n}, 1/f(x_{m+n}))$, entonces $r_{m,n}(x) = 1/r_{n,m}(x)$. Un inconveniente de este procedimiento es que no se puede aplicar si $f(x_i) = 0$ en algun nodo.

4.8. Algoritmo epsilon.

El siguiente algoritmo que se considera es el algoritmo ϵ , la deducción de éste está relacionada con el algoritmo de Gragg para interpolación racional. Se discutirá lo necesario del algoritmo de Gragg [3] solamente.

Para $k \geq 1$, considere la escalera ascendente

$$S_k = \left\{ r_{k,0}, r_{k-1,0}, r_{k-1,1}, r_{k-2,1}, r_{k-2,2}, \dots \right\}$$

y una fracción continua de la forma

$$f_k(x) = c_0 + \sum_{i=1}^k c_i (x-x_0) \dots (x-x_{i-1}) - \frac{c_k (x-x_0) \dots (x-x_{k-1})}{1}$$

$$- \frac{f_1^{(k)}}{x-x_k} - \frac{s_1^{(k)}}{1} - \frac{f_2^{(k)}}{x-x_k} - \frac{s_2^{(k)}}{1} - \dots$$

Proposición 4.11. Si cada tres elementos consecutivos en S_k son diferentes entonces el n -ésimo elemento de S_k es igual a la n -ésima convergente de $f_k(x)$, para $n = 0, 1, 2, \dots, 2k$.

Demostración. De acuerdo a lo que se busca es necesario pedir que

$$C_n = \frac{P_n}{Q_n} = r_{k - \left[\frac{n+1}{2} \right], \left[\frac{n}{2} \right]}$$

donde $\left[\frac{n+1}{2} \right] + \left[\frac{n}{2} \right] = n$. Utilizando la proposición (2.5) para construir fracciones continuas, se encontrará una del tipo (4.18).

Las expresiones P_0 y $P_1 - P_0$ son fáciles de calcular, se procederá al cálculo del resto de los numeradores y denominadores. Obsérvese que :

$$\frac{P_i}{Q_i} = r_{k - \left[\frac{i+1}{2} \right], \left[\frac{i}{2} \right]}, \quad \frac{P_{i-1}}{Q_{i-1}} = r_{k - \left[\frac{i}{2} \right], \left[\frac{i-1}{2} \right]} \quad \text{y} \quad \frac{P_{i+1}}{Q_{i+1}} = r_{k - \left[\frac{i+2}{2} \right], \left[\frac{i+1}{2} \right]}.$$

Para simplificar

$$\frac{P_i Q_{i+1} - P_{i+1} Q_i}{P_i Q_{i-1} - P_{i-1} Q_i},$$

por medio de la proposición (4.8) deben hacerse las siguientes consideraciones : Para i impar se cumple

$$k - \left[\frac{i+1}{2} \right] = k - \left[\frac{i+2}{2} \right] \quad \text{y} \quad \left[\frac{i}{2} \right] + 1 = \left[\frac{i+1}{2} \right]$$

$$k - \left[\frac{i+1}{2} \right] + 1 = k - \left[\frac{i}{2} \right] \quad \text{y} \quad \left[\frac{i}{2} \right] = \left[\frac{i-1}{2} \right]$$

mientras que para i par

$$k - \left[\frac{i+1}{2} \right] = k - \left[\frac{i+2}{2} \right] + 1 \quad \text{y} \quad \left[\frac{i}{2} \right] = \left[\frac{i+1}{2} \right]$$

$$k - \left[\frac{i+1}{2} \right] = k - \left[\frac{i}{2} \right] \quad \text{y} \quad \left[\frac{i}{2} \right] = \left[\frac{i-1}{2} \right] + 1$$

De esto se sigue que, por ejemplo, cuando i es par se cumple

$$\frac{P_i Q_{i+1} - P_{i+1} Q_i}{P_i Q_{i-1} - P_{i-1} Q_i} = \frac{\nu_1(x) \prod_0 (x - x_j)^{k - \left[\frac{i+2}{2} \right] + \left[\frac{i+1}{2} \right]}}{\nu_2(x) \prod_0 (x - x_j)^{k - \left[\frac{i}{2} \right] + \left[\frac{i-1}{2} \right]}} = \text{constante} \quad (4.23)$$

porque $\partial \nu_1 = \partial \nu_2 = 0$ y $k - [i+2/2] + [i+1/2] = k - [i/2] + [i-1/2]$

Cuando i es impar también es fácil de comprobar (4.23).

Ahora se simplificará

$$\frac{P_{i+1} Q_{i-1} - P_{i-1} Q_{i+1}}{P_i Q_{i-1} - P_{i-1} Q_i} \quad (4.24)$$

Para esto se utiliza la fórmula que ya obtuvimos para el denominador de (4.23) y la observación de que $k - [i/2] + [i-1/2] = k - 1$. Por otro lado para simplificar el numerador de (4.24), obsérvese que

$$(P_{i+1} Q_{i-1} - P_{i-1} Q_{i+1})(x_j) = 0, \quad j = 0, \dots, k - [i+2/2] + [i+1/2]$$

junto con que $k - [i+2/2] + [i+1/2] = k - [i/2] + [i-1/2]$, para concluir que

$$(P_{i+1} Q_{i-1} - P_{i-1} Q_{i+1})(x) = \prod_0^{k - \left[\frac{i+2}{2} \right] + \left[\frac{i+1}{2} \right]} (x - x_j),$$

y como

$$\left[\frac{i+1}{2} \right] - \left[\frac{i+2}{2} \right] = \begin{cases} 0 & \text{si } i \text{ es impar} \\ -1 & \text{si } i \text{ es par} \end{cases}.$$

entonces

$$(P_{i+1} Q_{i-1} - P_{i-1} Q_{i+1})(x) = \begin{cases} \prod_{j=1}^k (x-x_j) & \text{si } i \text{ es impar} \\ 0 & \\ \prod_{j=1}^{k-1} (x-x_j) & \text{si } i \text{ es par} \\ 0 & \end{cases}$$

Al sustituir en (4.24) las fórmulas obtenidas se demuestra el resultado. ■

Con la ayuda del algoritmo de Gragg se desarrollará el algoritmo ϵ . Se empieza con la siguiente proposición que es el análogo de la regla cruzada en aproximación racional (proposición 3.13).

Proposición 4.12. Si $r_{m,n} = p_{m,n} / q_{m,n}$ entonces

$$(x-x_{m+n})^{-1}(r_{m-1,n} - r_{m,n})^{-1} + (x-x_{m+n+1})^{-1}(r_{m+1,n} - r_{m,n})^{-1} =$$

$$(x-x_{m+n})^{-1}(r_{m,n-1} - r_{m,n})^{-1} + (x-x_{m+n+1})^{-1}(r_{m,n+1} - r_{m,n})^{-1}$$

Demostración. Como $r_{m-1,n+1}$, $r_{m-1,n}$ y $r_{m,n}$ están en la escalera S_{m+n} y

$$r_{m-1, n} = r_{m+n-(n+1), n} = \frac{P_{2n+1}}{Q_{2n+1}}$$

$$r_{m-1, n+1} = r_{m+n-(n+1), n+1} = \frac{P_{2n+2}}{Q_{2n+2}}$$

Entonces, utilizando las fórmulas de recurrencia para los numeradores y denominadores de una fracción continua, se obtiene

$$p_{m-1, n+1} = (x - x_{m+n})p_{m-1, n} - f_{n+1}^{(n+m)} p_{m, n} \quad (4.25)$$

$$q_{m-1, n+1} = (x - x_{m+n})q_{m-1, n} - f_{n+1}^{(n+m)} q_{m, n}$$

y

$$p_{m-1, n} = p_{m, n} - s_n^{(m+n)} p_{m, n-1} \quad (4.26)$$

$$q_{m-1, n} = q_{m, n} - s_n^{(m+n)} q_{m, n-1}$$

Si de (4.25) se despeja $p_{m-1, n}$ y $q_{m-1, n}$ se obtiene

$$r_{m-1, n} - r_{m, n} = \frac{p_{m-1, n+1} + f_{n+1}^{(m+n)} p_{m, n}}{q_{m-1, n+1} + f_{n+1}^{(m+n)} q_{m, n}} - \frac{p_{m, n}}{q_{m, n}}$$

$$= \frac{p_{m-1, n+1} q_{m, n} - p_{m, n} q_{m-1, n+1}}{q_{m, n} (q_{m-1, n+1} + f_{n+1}^{(m+n)} q_{m, n})} \quad (4.27)$$

Por otro lado al despejar en (4.26) $p_{m, n-1}$ y $q_{m, n-1}$ se obtiene

$$r_{m, n} - r_{m, n-1} = \frac{p_{m, n}}{q_{m, n}} - \frac{p_{m-1, n} - p_{m, n}}{q_{m-1, n} - q_{m, n}}$$

Si ahora se despeja $p_{m-1, n}$ y $q_{m-1, n}$ en (4.25) y se sustituye en la ecuación anterior se llega a la siguiente fórmula

$$\begin{aligned} r_{m, n} - r_{m, n-1} &= \frac{p_{m, n}}{q_{m, n}} - \frac{p_{m, n} (x - x_{m+n} - f_{n+1}^{(m+n)}) - p_{m-1, n+1}}{q_{m, n} (x - x_{m+n} - f_{n+1}^{(m+n)}) - q_{m-1, n+1}} \\ &= \frac{p_{m-1, n+1} q_{m, n} - p_{m, n} q_{m-1, n+1}}{q_{m, n} [q_{m, n} (x - x_{m+n} - f_{n+1}^{(m+n)}) - q_{m-1, n+1}]} \quad (4.28) \end{aligned}$$

Ahora combinando (4.27) y (4.28) se llega al siguiente resultado

$$(r_{m-1, n} - r_{m, n})^{-1} + (r_{m, n} - r_{m, n-1})^{-1} = \frac{(x - x_{m+n}) q_{m, n}}{q_{m-1, n+1}} \frac{1}{r_{m-1, n+1} - r_{m, n}} \quad (4.29)$$

Los cálculos que se hicieron se pueden repetir, pero utilizando el hecho de que $r_{m,n+1}$, $r_{m,n}$ y $r_{m+1,n}$ pertenecen a la escalera S_{m+n+1} . Para llegar a la fórmula

$$(r_{m,n+1} - r_{m,n})^{-1} + (r_{m,n} - r_{m+1,n})^{-1} = \frac{(x-x_{m+n+1}) q_{m,n}}{q_{m-1,n+1} (r_{m-1,n+1} - r_{m,n})} \quad (4.30)$$

Las fórmulas (4.29) y (4.30) demuestran la proposición. ■

El siguiente paso es utilizar la regla cruzada y la fórmula de la proposición (4.12), para deducir el algoritmo ϵ . El camino es similar al caso de aproximación.

Si se hace $r_{m,n} = \epsilon_{2n}^{(m-n)}$ y se arreglan estos coeficientes en una tabla, llamada ϵ -tabla, donde los sub-índices indican columna y el super-índice diagonal. Junto con esto se definen los coeficientes $\epsilon'n$ en las columnas impares por :

$$\epsilon_{2n+1}^{(m-n-1)} = \epsilon_{2n-1}^{(m-n)} + \frac{1}{(x-x_{m+n}) (\epsilon_{2n}^{(m-n)} - \epsilon_{2n}^{(m-n-1)})} \quad (4.31)$$

Entonces se verifica :

Proposición 4.13. La tabla ϵ se puede calcular a través de las condiciones iniciales

$$\epsilon_{2n}^{(-n-1)}, \quad \epsilon_{-1}^{(m)} = 0 \quad \text{y} \quad \epsilon_0^{(m)} = r_{m,0}, \quad n, m = 0, 1, \dots$$

y la fórmula

$$\epsilon_{2n-1}^{(m-n+1)} = \epsilon_{2n-1}^{(m-n)} + \frac{1}{(x-x_{m+n})(\epsilon_{2n}^{(m-n)} - \epsilon_{2n-2}^{(m-n+1)})} \quad (4.32)$$

Demostración. La fórmula (4.32) se demuestra por inducción en n . Para ésto primero escribáse la regla cruzada como

$$\begin{aligned} (x-x_{m+n+1})^{-1}(\epsilon_{2n+2}^{(m-n-1)} - \epsilon_{2n}^{(m-n)})^{-1} + (x-x_{m+n})^{-1}(\epsilon_{2n}^{(m-n)} - \epsilon_{2n}^{(m-n-1)})^{-1} = \\ (x-x_{m+n+1})^{-1}(\epsilon_{2n}^{(m+1-n)} - \epsilon_{2n}^{(m-n)})^{-1} + (x-x_{m+n})^{-1}(\epsilon_{2n}^{(m-n)} - \epsilon_{2n-2}^{(m-n+1)})^{-1} \end{aligned}$$

Ahora utilizando la definición de los ϵ 'n en las columnas impares, la fórmula anterior se puede reescribir como

$$(x-x_{m+n+1})^{-1}(\epsilon_{2n+2}^{(m-n-1)} - \epsilon_{2n}^{(m-n)})^{-1} + \epsilon_{2n+1}^{(m-n-1)} - \epsilon_{2n+1}^{(m-n)} =$$

$$\epsilon_{2n+1}^{(m-n)} = \epsilon_{2n+1}^{(m-n-1)} + (x-x_{m+n})^{-1} (\epsilon_{2n}^{(m-n)} - \epsilon_{2n-2}^{(m-n+1)})^{-1} \quad (4.33)$$

Ahora si se supone que (4.32) es cierta y se utiliza (4.33) entonces es facil demostrar (4.32) cuando n se sustituye por $n+1$. ■

El algoritmo epsilon tambien se programó en PADE, a continuación presentamos el listado de este programa.

```

Unit   Algo6;
Procedure  Int3;
Begin
  If  n[t]=0  then
    Begin
      DifDiv(m[t],F,Nodo,DifDiv1,DifDiv2);
      X1:=Xmin;
      Horner(m[t],DifDiv1,Nodo,X1,Evaluacion1);
      Y1:=Evaluacion1;
      For  r:=1  to  200  do
        Begin
          X2:=Xmin+(Xmax-Xmin)*r/200;
          Horner(m[t],DifDiv1,Nodo,X2,Evaluacion2);
          Y2:=Evaluacion2;
          Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
          X1:=X2;
          Y1:=Y2;
        End;
      End;
    End;
End;

```

```

If ((m[t]<n[t])and(m[t]>0)) then
Begin
For r:=0 to n[t]+m[t] do
Begin
If Abs(F[r])>MinimoReal then
FRec[r]:=1/F[r];
If Abs(F[r])<MinimoReal then
FRec[r]:=MaximoReal;
End;
DifDiv(m[t]+n[t],FRec,Nodo,DifDiv1,DifDiv2);
X1:=Xmin;
For s:=n[t]-m[t] to m[t]+n[t] do
Begin
Horner(s,DifDiv1,Nodo,X1,Evaluacion1);
Parcial1[s]:=Evaluacion1;
End;
Epsilondos(n[t],m[t],Evaluacion1,Parcial1,X1,Nodo);
If Abs(Evaluacion1)>MinimoReal then
Y1:=1/Evaluacion1;
For r:=1 to 200 do
Begin
X2:=Xmin+(Xmax-Xmin)*r/200;
For s:=n[t]-m[t] to m[t]+n[t] do
Begin
Horner(s,DifDiv1,Nodo,X2,Evaluacion2);
Parcial2[s]:=Evaluacion2;
End;
Epsilondos(n[t],m[t],Evaluacion2,Parcial2,X2,Nodo);
If Abs(Evaluacion2)>MinimoReal then
Y2:=1/Evaluacion2;
Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
X1:=X2;
Y1:=Y2;
End;
End;
End;
End.

```

```

If ((m[t]>=n[t])and(n[t]>0)) then
Begin
DifDiv(m[t]+n[t],F,Nodo,DifDiv1,DifDiv2);
X1:=Xmin;
For s:=m[t]-n[t] to m[t]+n[t] do
Begin
Horner(s,DifDiv1,Nodo,X1,Evaluacion1);
Parcial1[s]:=Evaluacion1;
End;
Epsilon(m[t],n[t],Evaluacion1,Parcial1,X1,Nodo);
Y1:=Evaluacion1;
For r:=1 to 200 do
Begin
X2:=Xmin+(Xmax-Xmin)*r/200;
For s:=m[t]-n[t] to m[t]+n[t] do
Begin
Horner(s,DifDiv1,Nodo,X2,Evaluacion2);
Parcial2[s]:=Evaluacion2;
End;
Epsilon(m[t],n[t],Evaluacion2,Parcial2,X2,Nodo);
Y2:=Evaluacion2;
Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
X1:=X2;Y1:=Y2;
End;
End;
If m[t]=0 then
Begin
For r:=0 to n[t] do
Begin
If Abs(F[r])>MinimoReal then
FRec[r]:=1/F[r];
End;
DifDiv(n[t],FRec,Nodo,DifDiv1,DifDiv2);X1:=Xmin;
Horner(n[t],DifDiv1,Nodo,X1,Evaluacion1);
If Abs(Evaluacion1)>MinimoReal then
Y1:=1/Evaluacion1;
For r:=1 to 200 do
Begin
X2:=Xmin+(Xmax-Xmin)*r/200;
Horner(n[t],DifDiv1,Nodo,X2,Evaluacion2);
If Abs(Evaluacion2)>MinimoReal then
Y2:=1/Evaluacion2;
Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
X1:=X2;Y1:=Y2;
End;
End;
End;

```

4.9. Interpolación osculatoria

4.9.1. Introducción.

El tema que se estudiará ahora, trata sobre la interpolación racional, en la que además se conocen en cada nodo la derivada de la función hasta determinado orden. Es de esperar que este tipo de interpolación de mejores aproximaciones a la función. Este tema es generalización de los de aproximación e interpolación y nos referiremos a él como interpolación racional de Hermite u osculatoria.

4.9.2. Definición.

Definición 4.4. Sean $(x_i)_{i=0}^{\infty}$ y $(s_i)_{i=0}^{\infty}$ sucesiones en \mathbb{C} y \mathbb{N} respectivamente. Supongamos que la derivada $f^{(l)}(x_i)$ de la función f evaluadas en x_i son dadas para $l=0, 1, \dots, s_i - 1$. Consideremos enteros fijos j, k, m y n tales que

$$1 \leq k \leq s_{j+1} \quad \text{y} \quad m+n+1 = \sum_0^j s_i + k$$

El problema de interpolación de Hermite de orden (m, n) para f

consiste en encontrar dos polinomios

$$p(x) = \sum_0^m a_i x^i \quad y \quad q(x) = \sum_0^n b_i x^i$$

con p/q irreducible y que satisfagan el siguiente criterio de aproximación a f

$$f^{(l)}(x_i) = \left[\frac{p}{q} \right]^{(l)}(x_i), \quad \begin{array}{l} l=0,1,\dots,s_i-1 \\ i=0,1,\dots,j \end{array} \quad (4.34)$$

$$f^{(l)}(x_{j+1}) = \left[\frac{p}{q} \right]^{(l)}(x_{j+1}), \quad l=0,1,\dots,k-1 \quad \blacksquare$$

4.9.3. Formas equivalentes del problema de interpolación de Hermite.

En esta sección se presentan varias formas de escribir el problema de interpolación de Hermite, con el fin de facilitar su solución.

Proposición 4.14. El sistema de ecuaciones (4.34) es equivalente al sistema

$$\begin{aligned}
 (fq - p)^{(l)}(x_i) &= 0, & l = 0, 1, \dots, s_i - 1 \\
 & & i = 0, 1, \dots, j \\
 (fq - p)^{(l)}(x_{j+1}) &= 0, & l = 0, 1, \dots, k-1
 \end{aligned}
 \tag{4.35}$$

Demostración. Supongamos que (4.34) es verdadera pero (4.35) no. Entonces $(fq - p)^{(l)}(x_i) \neq 0$, lo cual implica que

$$\left[q \left[f - \frac{p}{q} \right] \right]^{(l)}(x_i)$$

es cero y no es cero por la regla de Leibniz. La demostración de la otra parte es parecida. ■

El sistema de ecuaciones (4.35) tiene $m+n+2$ incógnitas con $m+n+1$ ecuaciones. Así tal sistema tiene por lo menos una solución no trivial. Se demostrará que dos soluciones son equivalentes como funciones racionales.

Proposición 4.15. Si p_1/q_1 y p_2/q_2 son solución del problema de interpolación racional de Hermite de orden (m, n) , entonces $p_1q_2 = p_2q_1$.

Demostración. La función $\phi(x) = (p_1q_2 - p_2q_1)(x)$ es un

polinomio que satisface $\partial\phi \leq m + n$. Pero por otro lado como (p_1, q_1) y (p_2, q_2) son solución del problema de interpolación de Hermite, se satisface

$$\phi^{(l)}(x_i) = (-q_2(fq_1 - p_1) + q_1(fq_2 - p_2))^{(l)}(x_i) = 0$$

para $l = 0, 1, \dots, s_i - 1$, $i = 0, 1, \dots, j$, $l = k$ y $i=j+1$

Lo cual implica que

$$\phi(x) = (x - x_0)^{s_0} (x - x_1)^{s_1} \dots (x - x_{j+1})^k$$

Por lo tanto $\partial\phi > m + n + 1$. Pero de estas desigualdades sobre el grado de ϕ nos obligan a deducir que $\phi \equiv 0$, con lo que queda demostrado el resultado. ■

Definición 4.5. La única fracción irreducible que se obtiene de la solución del problema de interpolación racional de Hermite para f , con la normalización $q_0(x_0) = 1$, se llama la interpolante racional de Hermite de orden (m, n) para f . ■

Ejemplo 4.6. Para encontrar la interpolante racional de Hermite de orden $(3, 2)$ para $f(x) = \ln(1+x)$, y para la información

de la tabla 4.5, se necesita resolver el sistema de ecuaciones generado por las ecuaciones (4.35), donde

$$p(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0 \quad y \quad q(x) = b_2 x^2 + b_1 x + 1$$

TABLA 4.5.			
x_1	0	0.5	1
y_1	0	0.4054651	0.6931471
y_1'	1	0.6666666	0.5

Resolviendo el sistema (4.35), se obtienen los coeficientes de los polinomios

$$a_0 = 0, \quad a_1 = 1, \quad a_2 = 3.8128797648E-01, \quad a_3 = -1.109490387292E-04$$

$$b_0 = 1, \quad b_1 = 8.8022511942E-01 \quad y \quad b_2 = 1.1239236025E-01$$

Para comparar las estimaciones dadas por nuestra interpolante de Hermite y otra, tomaremos el polinomio mini-máx de grado cinco en $[0,1]$ [4]. Sus coeficientes son

$$a_1 = 0.99949556 \quad a_3 = 0.28947478 \quad a_5 = 0.03215845$$

$$a_2 = -0.49190896 \quad a_4 = -0.13606275$$

En la tabla (4.6) se muestran algunas de las aproximaciones dadas por ambos métodos.

TABLA 4.6.

y_i	PADE	$\ln(1+y_i)$	Mini-Max
-5.00E-01	-6.8822125097E-01	-6.9314718056E-01	-6.6841824093E-01
-4.00E-01	-5.0907121796E-01	-5.1082562376E-01	-5.0084255245E-01
-3.00E-01	-3.5611805950E-01	-3.5667494394E-01	-3.5311654677E-01
-2.00E-01	-2.2300373447E-01	-2.2314355131E-01	-2.2211925974E-01
-1.00E-01	-1.0534099286E-01	-1.0536051566E-01	-1.0517204824E-01
0.00E+00	0.0000000000E+00	0.0000000000E+00	0.0000000000E+00
1.00E-01	9.5315712771E-02	9.5310179803E-02	9.5306656489E-02
2.00E-01	1.8233223821E-01	1.8232155679E-01	1.8233114214E-01
3.00E-01	2.6237437015E-01	2.6236426447E-01	2.6236871742E-01
4.00E-01	3.3647771633E-01	3.3647223662E-01	3.3646527245E-01
5.00E-01	4.0546506428E-01	4.0546510811E-01	4.0545591719E-01
6.00E-01	4.6999964013E-01	4.7000362925E-01	4.7000357155E-01
7.00E-01	5.3062303680E-01	5.3062825106E-01	5.3063755555E-01
8.00E-01	5.8778273312E-01	5.8778666490E-01	5.8779217945E-01
9.00E-01	6.4185239799E-01	6.4185388617E-01	6.4184533388E-01
1.00E+00	6.9314710000E-01	6.9314718056E-01	6.9315708000E-01
1.10E+00	7.4193485875E-01	7.4193734473E-01	7.4210823961E-01
1.20E+00	7.8844552883E-01	7.8845736036E-01	7.8913898534E-01
1.30E+00	8.3287771175E-01	8.3290912293E-01	8.3478743074E-01
1.40E+00	8.7540419103E-01	8.7546873735E-01	8.7972822045E-01

A partir de estos resultados podemos hacer los siguientes comentarios. Para empezar, es necesario estar

concientes del trabajo que nos costo obtener cada aproximante. Sin lugar a dudas la interpolante racional de Hermite supera rotundamente a la mini-max. Otro aspecto importante que es necesario comparar, es el error absoluto generado por las aproximaciones de ambos métodos. A partir de la tabla 4.6., se observa que las aproximaciones dadas por la interpolante racional de Hermite coinciden en cuatro cifras con los valores exactos desde -0.1 hasta 1.2 . Mientras que las aproximaciones dadas por la mini-max lo hacen sólo de 0 a 1 . Claro que esta comparación adolece de ser sólo en un número finito de puntos. Sin embargo en [4] nos dicen que el error máximo absoluto en el caso de la mini-max es de 0.00001 . Para el caso de la interpolante racional de Hermite, el error absoluto máximo se puede calcular fácilmente. Sea $e(x)$ el error, es decir

$$e(x) = \ln(1+x) - p_0(x)/q_0(x).$$

Para encontrar los puntos donde $e(x)$ alcanza sus valores extremos en $[0,1]$, aplicamos el criterio de la primer derivada. Como la derivada de la función y la derivada de la interpolante racional son iguales en los nodos, tenemos tres puntos críticos automáticamente. Pero hay dos más, el numerador de $e'(x)$ es de grado cinco, estos son $2.382378734E-01$ y $6.93618445E-01$.

Evaluando la función de error en estos puntos, concluimos que el máximo error esperado es $1.11203589E-05$. ■

Aunque la fórmula (4.35) es adecuada para encontrar interpolantes racionales de Hermite, es posible mejorarla aún más. Para esto necesitamos escribir las funciones en términos de los polinomios de Newton.

Definición 4.6. Sean $(x_i)_{i=0}^{\infty}$ y $(y_i)_{i=0}^{\infty}$ sucesiones en \mathbb{C} y \mathbb{N} respectivamente. Se definen

$$\begin{aligned}
 y_1 &= x_0 & , & \quad l = 0, 1, \dots, s_0 - 1 \\
 y_{d(i)+1} &= x_i & , & \quad l = 0, 1, \dots, s_i - 1 \\
 & & & \quad d(i) = s_0 + s_1 + \dots + s_{i-1} \\
 c_{i,j} &= 0 & , & \quad i > j \\
 c_{i,j} &= f[y_i, \dots, y_j] & , & \quad i \leq j
 \end{aligned}$$

También definimos los polinomios de Newton, con respecto a la sucesión $(y_i)_{i=0}^{\infty}$, como

$$\begin{aligned}
 B_0(x) &= 1 \\
 B_j(x) &= (x-y_0)(x-y_1) \dots (x-y_{j-1}) & , & \quad j = 1, 2, \dots \quad \blacksquare
 \end{aligned}$$

Sin entrar en detalles, supondremos que una función f se puede expresar en términos de los polinomios de Newton. Es decir

$$f(x) = \sum_0^{\infty} c_{0i} B_i(x) \quad (4.36)$$

donde $c_{0i} = f(y_0, \dots, y_i)$. Decimos que (4.36) es la serie de Newton de f .

Utilizando la escritura (4.36) para una función, se tiene la siguiente forma equivalente del problema de interpolación racional de Hermite.

Proposición 4.16. El problema de interpolación racional de Hermite (4.35), es equivalente a encontrar polinomios

$$p(x) = \sum_{i=0}^m a_i B_i(x) \quad y \quad q(x) = \sum_{i=0}^n b_i B_i(x)$$

tales que

$$(fq - p)(x) = \sum_{i=m+n+1}^{\infty} d_i B_i(x) \quad (4.37)$$

donde $d_i = (fq - p)[y_0, \dots, y_i]$.

Demostración. Si escribimos

$$(fq - p)(x) = \sum_{i=0}^{\infty} d_i B_i(x) \quad (4.38)$$

Bastará comprobar que $d_i = 0$, para $i = 0, 1, \dots, m+n$. Pero esto es fácil de obtener si calculamos las derivadas (4.35) en la fórmula (4.38). ■

4.9.4. Algoritmo de Gauss.

Utilizando la ecuación (4.37) se puede dar un algoritmo que nos permite construir los coeficientes de los numerador y denominador de la interpolante racional de Hermite. Para esto necesitamos la siguiente proposición, su demostración se puede encontrar en [9].

Proposición 4.17. (Fórmula de Leibnitz) Se cumple

$$(fq)[y_0, \dots, y_i] = \sum_{l=0}^i f[y_0, \dots, y_l] q[y_l, \dots, y_i] \quad (4.39)$$

Proposición 4.18. El problema de interpolación racional de Hermite es equivalente a resolver los dos sistemas de ecuaciones siguientes

$$\begin{aligned}
 c_{00} b_0 &= a_0 \\
 c_{01} b_0 + c_{11} b_1 &= a_1 \\
 &\vdots \\
 &\vdots \\
 c_{0,m} b_0 + c_{1m} b_1 + \dots + c_{n,m} b_n &= a_m
 \end{aligned}
 \tag{4.40}$$

$$\begin{aligned}
 c_{0,m+1} b_0 + c_{1,m+1} b_1 + \dots + c_{n,m+1} b_n &= 0 \\
 c_{0,m+2} b_0 + c_{1,m+2} b_1 + \dots + c_{n,m+2} b_n &= 0 \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 c_{0,m+n} b_0 + c_{1,m+n} b_1 + \dots + c_{n,m+n} b_n &= 0
 \end{aligned}
 \tag{4.41}$$

Demostración. Se utilizan las fórmulas (4.37) y (4.39). ■

En PADE hicimos un programa mediante el cual se encuentran los coeficientes de los polinomios que forman una interpolante racional de Hermite. Este programa se basa en resolver los sistemas de ecuaciones (4.40) y (4.41). A continuación presentamos el listado de este procedimiento.

```

Unit   Algo71;
Procedure   IntHer1;
Begin
  If   ((n[s]>1)and(m[s]>1))   then
  Begin
    For t:=1 to n[s] do
      Begin
        For r:=1 to n[s] do
          Begin
            If   ((r<n[s])and(r<=(m[s]+t)))   then
            Begin
              Coef1[t,r]:=DifeDivididas[m[s]+t,m[s]-(r-t)];
            End;
            If   ((r>n[s])or(r>(m[s]+t)))   then
              Coef1[t,r]:=0;
            End;
          End;
        End;
      For t:=1 to n[s] do
        Coef2[t]:=-DifeDivididas[m[s]+t,m[s]+t];
      End;
      Gauss(Coef2,Coef1,n[s],Incognita);
      b[0]:=1;
      For t:=1 to n[s] do
        b[t]:=Incognita[t];
      End;
      a[0]:=DifeDivididas[0,0];
      If m[s]>=1 then
      Begin
        For r:=1 to m[s] do
          Begin
            Intermedio:=0;
            For t:=0 to n[s] do
              Begin
                If r>=t then
                  Intermedio:=Intermedio+b[t]*DifeDivididas[r,r-t];
                End;
              a[r]:=Intermedio;
            End;
          End;
        End;
      X1:=Xmin;
      Horner(m[s],a,Y,X1,Evaluacion1);
      Horner(n[s],b,Y,X1,Evaluacion2);
      If   Abs(Evaluacion2)>MinimoReal   then
        Y1:=Evaluacion1/Evaluacion2;
      For r:=1 to 200 do
        Begin

```

```

        X2:=Xmin+((Xmax-Xmin)*r)/200;
        Horner(m[s],a,Y,X2,Evaluacion1);
        Horner(n[s],b,Y,X2,Evaluacion2);
        If Abs(Evaluacion2)>MinimoReal then
            Y2:=Evaluacion1/Evaluacion2;
            Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
            X1:=X2;
            Y1:=Y2;
        End;
    End;
End;
If n[s]=0 then
    Begin
        For r:=1 to m[s] do
            Begin
                a[r]:=DifeDivididas[r,r];
            End;
        a[0]:=DF[0,0];
        X1:=Xmin;
        Horner(m[s],a,Y,X1,Evaluacion1);
        Y1:=Evaluacion1;
        For r:=1 to 200 do
            Begin
                X2:=Xmin+((Xmax-Xmin)*r)/200;
                Horner(m[s],a,Y,X2,Evaluacion2);
                Y2:=Evaluacion2;
                Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
                X1:=X2;
                Y1:=Y2;
            End;
        End;
    End;
End;
Unit Algo72;
Procedure IntHer2;
Begin
    If n[s]=1 then
        Begin
            b[0]:=1;
            b[1]:=-DifeDivididas[m[s]+1,m[s]+1]/DifeDivididas[m[s]+1,m[s]];
            For r:=1 to m[s] do
                Begin
                    a[r]:=DifeDivididas[r,r]+DifeDivididas[r,r-1]*b[1];
                End;
            a[0]:=DifeDivididas[0,0];
            X1:=Xmin;
            Horner(m[s],a,Y,X1,Evaluacion1);

```

```

Horner(n[s],b,Y,X1,Evaluacion2);
If Abs(Evaluacion2)>MinimoReal then
  Y1:=Evaluacion1/Evaluacion2;
For r:=1 to 200 do
  Begin
    X2:=Xmin+((Xmax-Xmin)*r)/200;
    Horner(m[s],a,Y,X2,Evaluacion1);
    Horner(n[s],b,Y,X2,Evaluacion2);
    If Abs(Evaluacion2)>MinimoReal then
      Y2:=Evaluacion1/Evaluacion2;
      Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
      X1:=X2;
      Y1:=Y2;
    End;
  End;
If ((m[s]=1)and(n[s]>1)) then
  Begin
    For r:=1 to n[s] do
      Begin
        For t:=1 to n[s] do
          Begin
            If r<=t then
              Coef1[r,t]:=DifeDivididas[r+1,r+1-t];
            If r>t then
              Coef1[r,t]:=0;
            End;
          End;
        End;
      For r:=1 to n[s] do
        Begin
          Coef2[r]:=DifeDivididas[r+1,r+1];
        End;
      Gauss(Coef2,Coef1,n[s],Incognita);
      b[0]:=1;
      For t:=1 to n[s] do
        Begin
          b[t]:=Incognita[t];
        End;
      a[0]:=DifeDivididas[0,0];
      a[1]:=DifeDivididas[1,1]+DifeDivididas[1,0]*b[1];
      X1:=Xmin;
      Horner(m[s],a,Y,X1,Evaluacion1);
      Horner(n[s],b,Y,X1,Evaluacion2);
      If Abs(Evaluacion2)>MinimoReal then
        Y1:=Evaluacion1/Evaluacion2;
      For r:=1 to 200 do
        Begin

```

```

X2:=Xmin+((Xmax-Xmin)*r)/200;
Horner(m[s],a,Y,X2,Evaluacion1);
Horner(n[s],b,Y,X2,Evaluacion2);
If Abs(Evaluacion2)>MinimoReal then
Y2:=Evaluacion1/Evaluacion2;
Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);
X1:=X2;
Y1:=Y2;
End;
End;
If m[s]=0 then
Begin
a[0]:=DifeDivididas[0,0];
For r:=1 to n[s] do
Begin
For t:=1 to n[s] do
Begin
If r<=t then
Coef1[r,t]:=DifeDivididas[r,r-t];
If r>t then
Coef1[r,t]:=0;
End;
End;
For r:=1 to n[s] do
Begin
Coef2[r]:=DifeDivididas[r+1,r+1];
End;
Gauss(Coef2,Coef1,n[s],Incognita);
b[0]:=1;
For t:=1 to n[s] do
Begin
b[t]:=Incognita[t];
End;
X1:=Xmin;
Horner(m[s],a,Y,X1,Evaluacion1);
Horner(n[s],b,Y,X1,Evaluacion2);
If Abs(Evaluacion2)>MinimoReal then
Y1:=Evaluacion1/Evaluacion2;
For r:=1 to 200 do
Begin
X2:=Xmin+((Xmax-Xmin)*r)/200;
Horner(m[s],a,Y,X2,Evaluacion1);
Horner(n[s],b,Y,X2,Evaluacion2);
If Abs(Evaluacion2)>MinimoReal then
Y2:=Evaluacion1/Evaluacion2;
Grafica(X1,Y1,X2,Y2,Xmin,Ymax,D1,D2);

```

```
X1:=X2;  
Y1:=Y2;  
End;  
End;  
End;  
End.
```

Conclusiones.

Algunos comentarios finales sobre este trabajo se pueden hacer. Como se observó mi trabajo consistió en estudiar la teoría de aproximación de Padé, interpolación racional e interpolación racional osculatoria y programar algunos de los algoritmos que hay en esas teorías. Como se marco a lo largo de este trabajo, una limitación de los algoritmos cociente-diferencia y epsilon es que sólo se pueden aplicar si la aproximante es normal. De esta situación se desprenden dos cosas; primero, la posibilidad de generalizar estos algoritmos para el caso cuando no hay normalidad; segundo, dar criterios sencillos de aplicar que nos permitan conocer cuando una aproximante es normal. También queda como algo a completar la extensión de los algoritmos cociente-diferencia y epsilon a interpolación racional de Hermite. Otro aspecto importante por cubrir es la de obtener buenas fórmulas para estimar el error de aproximación, pues como vimos a través de comparación con tablas, las aproximantes racionales proporcionan buenas aproximaciones. También es necesario desarrollar estudios que nos permitan estimar los errores de redondeo, la eficiencia y estabilidad de los algoritmos.

Desde el punto de vista de programación es posible

mejorar el trabajo en el sentido de que pueda detectar cualquier anomalia y sepa que hacer ante una eventualidad de esas. También se puede mejorar para que sea más versátil y ágil. Posiblemente a mucha gente no le guste que se tenga que hacer un archivo para graficar la función que se aproxima, le hubiera gustado más que la función fuera dada a través del teclado, sin embargo este problema no es nada fácil. Otro aspecto importante en que se debe mejorar PADE, es que debería proporcionarnos la información numérica a través de diferentes opciones como un archivo o la pantalla, y no tener que meterse a los programas fuente para hacerlo.

La creación de programas para aproximar funciones de variable compleja a través de los métodos que se vieron en este trabajo es importante. Sería interesante poder aproximar las función de Bessel complejas a través de funciones racionales. Utilizar interpolación racional en \mathbb{C} para reproducir el ejemplo 4.1. para la función de error compleja, etc.

También es importante ver si las funciones racionales se pueden utilizar para generar aproximaciones por mínimos cuadrados o interpolación segmentaria. Las aplicaciones que puedan tener los temas de este trabajo en otras áreas como ecuaciones

diferenciales, integración, aceleración de convergencia, etc.
es también importante.

BIBLIOGRAFIA

[1] Abramowitz M. and Stegun I. Handbook of Mathematical functions. Dover publications, New York, 1968.

[2] Cuyt A. and Wuytack L. Nonlinear methods in numerical analysis. North-Holland Mathematics Studies, 1987.

[3] Gragg W. The Padé table and its relation to certain algorithms of numerical analysis. SIAM Rev. 14, 1972, 1-62.

[4] Hastings C. Jr. Approximations for digital computers. Princeton New Jersey, 1955.

[5] Nutall J. The convergence of Padé approximants of meromorphic functions. J. Math. Anal. Appl. 31, 1970, 147-153.

[6] Perron O. Die Lehre von den Kettenbrüchen II. Teubner, Stuttgart, 1977.

[7] Rutishauser H. Der quotienten-differenzen algorithmus. Mitteilungen Institut für a angewandte mathematik (ETH) 7, Birkhäuser Verlag, Basel, 1957.

[8] Shanks D. Nonlinear transformations of divergent and slowly convergent sequences. J. Math. and Phys. 34, 1955, 1-42.

[9] Verde-Star L. Interpolation and combinatorial functions. Studies in Mathematics 79, 65-92, 1988.

[10] Viscovatov B. De la méthode générale pour reduire toutes sortes de quantités en fractions continues. Mém. Acad. Impériale Sci. St.-Petersburg 1, 1803-1806, 226-247.

[11] Wynn P. On a device for computing the $e_m(S_n)$ transformation. MTAC 10, 1956, 91-96.