



Universidad de Valladolid

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Matemáticas

Métodos de extrapolación y algunas aplicaciones

Autor: Alazne Angulo Conde

*Tutor: Ángel Durán Martín
Departamento de Matemática Aplicada*

Métodos de extrapolación y algunas aplicaciones

Alazne Angulo Conde

Alazne Angulo Conde *Métodos de extrapolación y algunas aplicaciones.*
Trabajo de fin de Grado. Curso académico 2021-2022.

**Responsable de
tutorización**

Ángel Durán Martín
Departamento de Matemática Aplicada

Grado en Matemáticas
Facultad de Ciencias
Universidad de Valladolid

Índice general

1	Introducción	1
1.1	El problema de la extrapolación	2
1.2	Ejemplos	2
1.2.1	Método Δ^2 de Aitken	2
1.2.2	Extrapolación de Richardson	4
1.2.3	Extrapolación vectorial	8
2	Extrapolación escalar	13
2.1	Introducción	13
2.1.1	Algunos tipos de convergencia	13
2.1.2	Métodos de extrapolación	15
2.2	El algoritmo de extrapolación general (GEA) y la transformación de Shanks	19
2.2.1	La transformación de Shanks	24
2.3	Métodos polinómicos y racionales	32
2.3.1	Método de extrapolación de Richardson polinómica (PREP)	34
2.3.2	Extrapolación racional. Algoritmo ρ	39
2.3.3	Algunas aplicaciones	41
2.4	Otros métodos	46
2.4.1	Métodos iterados	46
2.4.2	La transformación de Levin	46
2.4.3	El algoritmo θ	47
3	Extrapolación vectorial	51
3.1	Introducción	51
3.1.1	Conceptos básicos	51
3.1.2	Implementación en modo cíclico	52
3.1.3	Polinomios mínimos asociados a una matriz	53
3.2	Métodos de extrapolación vectorial	54
3.2.1	Algoritmos tipo ε vectorial	54
3.2.2	Métodos polinómicos	60
3.3	Convergencia y estabilidad de los métodos de extrapolación vectorial	68
3.3.1	Métodos vectoriales de tipo ε	68
3.3.2	Métodos polinómicos.	69
3.4	Aplicaciones de la extrapolación vectorial	72
3.4.1	Aceleración en procesos de Markov	72
3.4.2	Aproximación a soluciones especiales de ecuaciones en derivadas parciales.	79
	Bibliografía	89

1 Introducción

La presente memoria trata de los métodos de extrapolación para la aceleración de la convergencia de sucesiones escalares y vectoriales. Sus principales objetivos son:

1. Realizar un análisis de las principales familias de métodos de extrapolación escalar y vectorial. El estudio incluye aspectos de convergencia, estabilidad, implementación e ilustración con ejemplos.
2. Proporcionar una cierta perspectiva de los diferentes ámbitos de aplicación de los métodos, desde los clásicos (ya incorporados en los libros de texto) hasta los más actuales. Entre estos últimos, y no pudiendo, por razones obvias, abarcar todas las aplicaciones (véanse, en este sentido, las referencias [59, 18, 56]), nos centramos en el uso de la extrapolación para procesos de Markov y la aproximación de soluciones especiales de ecuaciones en derivadas parciales.

La memoria tiene la siguiente estructura.

El primer capítulo es introductorio y en él se presentan las ideas principales de la teoría para la aceleración de procesos secuenciales, así como la ilustración de las mismas, como iniciación, en métodos clásicos como la aceleración de Aitken y la extrapolación de Richardson.

El capítulo 2 está dedicado a la extrapolación escalar. Tras la presentación de algunos conceptos básicos, dividimos nuestro estudio en dos grandes familias de métodos: aquéllos basados en el algoritmo de extrapolación general (cuyo representante principal es la transformación de Shanks) y los métodos polinómicos y racionales, de los que destacaremos el método de extrapolación de Richardson polinómico y el algoritmo ρ . En todos los casos, el análisis seguirá un mismo desarrollo: presentación del método, discusión de aspectos de convergencia y estabilidad e implementación, con algunas aplicaciones clásicas. El capítulo se cierra con un resumen de otros procedimientos de extrapolación escalar y referencias al lector interesado en profundizar en ellos.

La aceleración de la convergencia para el caso vectorial se analiza en el capítulo 3. También aquí es posible distinguir dos tipos de procedimientos. El primer grupo procede de la extensión vectorial de los algoritmos de tipo ε escalares, entre los que destacan la generalización del esquema de Wynn y el algoritmo topológico de Brezinski. Otro punto de vista está fundamentado en la extrapolación exacta para recurrencias lineales basadas en el polinomio mínimo asociado a una matriz. Dentro de este grupo, nos centraremos en el método del polinomio mínimo y en la extrapolación de rango reducido. El análisis tendrá aquí elementos similares al del caso escalar (presentación, análisis de convergencia e implementación). La variedad de contextos en los que la extrapolación vectorial puede utilizarse (quizá algo menos conocidos que los del caso escalar) queda en este capítulo representado por dos aplicaciones: una está asociada a procesos lineales de recurrencia con cadenas de Markov; en el segundo caso se analiza el uso de extrapolación vectorial en esquemas iterativos no lineales para la aproximación de soluciones especiales de ecuaciones en derivadas parciales.

1.1. El problema de la extrapolación

Los métodos de extrapolación, objeto de estudio del presente trabajo, pueden utilizarse en aquellos procesos secuenciales en Matemática Aplicada, escalares o vectoriales, que requieran mejorar su rendimiento, bien mediante una aceleración de su convergencia o bien, en caso de divergencia, mediante su transformación a procesos convergentes. Su ámbito de aplicación es, pues, muy amplio, abarcando diferentes tópicos en Matemáticas (con especial incidencia en el Análisis Numérico) y, consecuentemente, en la modelización. A modo ilustrativo, podemos mencionar algunos:

- El tratamiento de series numéricas, incluyendo aspectos como la resolución de ecuaciones diferenciales por series de potencias, métodos asintóticos en teoría de perturbación o el fenómeno de Gibbs en el análisis de Fourier.
- La mejora del rendimiento en la resolución iterativa de sistemas lineales y no lineales, incluyendo el cómputo de autovalores, la aproximación numérica de ecuaciones diferenciales e integrales, o la implementación de problemas de optimización.

Si bien la idea de extrapolación está siempre presente en la formulación de los métodos de aceleración de la convergencia de procesos secuenciales, puede ser formativo mencionar otros puntos de vista. El fundamental tiene que ver con el hecho de que la aceleración es una transformación de una sucesión más lenta (o simplemente divergente) en otra que converge más rápidamente. (En el caso lineal, esto es conocido como proceso de sumación.) Esta perspectiva se encuentra en la base de la construcción de muchos métodos, a partir de la búsqueda de aquellas sucesiones que la transformación convierte en sucesiones constantes (el llamado núcleo de la transformación). Si tal constante tiene que ver con el límite (o, en el caso divergente, lo que se denominaría antilímite) de la sucesión original, la aplicación de la transformación a una sucesión que no esté en su núcleo generará al menos una aproximación a dicho valor. Esto sugiere la no existencia de una transformación ‘universal’, que acelere la convergencia de todas las sucesiones pero, al mismo tiempo, proporcione a la teoría el dinamismo necesario para su desarrollo actual.

Estos dos puntos de vista (extrapolación y transformación de sucesiones) son los que se utilizarán en el presente trabajo para abordar el estudio de los métodos de aceleración de convergencia. No hay que olvidar mencionar aquí, sin embargo, otros tópicos relacionados, bajo los que puede estudiarse el problema. Tal es el caso de la aproximación racional al límite con aproximantes de Padé (de tipo escalar o vectorial), fracciones continuas o la teoría generalizada de polinomios ortogonales, [18].

1.2. Ejemplos

Podemos ilustrar las ideas anteriores con los siguientes ejemplos.

1.2.1. Método Δ^2 de Aitken

Uno de los métodos de extrapolación clásicos es el algoritmo Δ^2 propuesto por el matemático neozelandés A. C. Aitken (1898-1967), [1]. Consideramos el operador Δ de diferencias progresivas que para cada sucesión $\{S_n\}_{n=0}^{\infty}$ genera la sucesión $\{\Delta S_n\}_{n=0}^{\infty}$ con

$$\Delta S_n := S_{n+1} - S_n, \quad n \geq 0.$$

El método Δ^2 de Aitken transforma $\{S_n\}_{n=0}^{\infty}$ en la sucesión $\{T_n\}_{n=0}^{\infty}$ con

$$T_n = S_n - \frac{(\Delta S_n)^2}{\Delta^2 S_n}, \quad n \geq 0, \quad (1.1)$$

donde $\Delta^2 S_n = \Delta(\Delta S_n) = S_{n+2} - 2S_{n+1} + S_n$.

En el capítulo 2 mencionaremos que el núcleo de la transformación (1.1) viene dado por aquellas sucesiones $\{S_n\}_{n=0}^{\infty}$ para las que existen $\lambda \neq 0, 1$ y S de modo que

$$S_{n+1} - S = \lambda (S_n - S), \quad n = 0, 1, \dots$$

o, explícitamente

$$S_n = S + \alpha \lambda^n, \quad n = 0, 1, \dots \quad (1.2)$$

con α arbitrario. (Recordemos que esto significa que $T_n = S$ si y solo si S_n es de la forma (1.2).) La aceleración de convergencia que proporciona (1.1) está basada en el siguiente resultado:

Teorema 1.1. Sea $S \in \mathbb{C}$ tal que

$$\lim_{n \rightarrow \infty} \left(\frac{S_{n+1} - S}{S_n - S} \right) = \lambda, \quad (1.3)$$

con $\lambda \neq 0, 1$. Entonces

$$\lim_{n \rightarrow \infty} \left(\frac{T_n - S}{S_n - S} \right) = 0.$$

Demostración:

Sea $e_n = S_n - S$, $n = 0, 1, \dots$. Utilizando la fórmula (1.1) podemos escribir

$$T_n - S = \frac{e_n e_{n+2} - e_{n+1}^2}{e_{n+2} - 2e_{n+1} + e_n}.$$

Entonces, operando, se tiene

$$\frac{T_n - S}{S_n - S} = \frac{A_n}{B_n},$$

con

$$A_n = \frac{e_{n+2}}{e_{n+1}} - \frac{e_{n+1}}{e_n}, \quad B_n = \frac{\frac{e_{n+1}}{e_n} \left(\frac{e_{n+2}}{e_{n+1}} - 1 \right) - \left(\frac{e_{n+1}}{e_n} - 1 \right)}{\frac{e_{n+1}}{e_n}}.$$

Se obtiene el resultado pasando al límite $n \rightarrow \infty$ y usando (1.3) de manera que

$$\lim_{n \rightarrow \infty} A_n = 0, \quad \lim_{n \rightarrow \infty} B_n = \frac{(\lambda - 1)^2}{\lambda} \neq 0. \quad \square$$

Algunos comentarios sobre el teorema 1.1 son los siguientes:

- La hipótesis (1.3) justifica la fórmula (1.1) en el sentido siguiente: (1.3) implica que para n grande

$$\frac{S_{n+1} - S}{S_n - S} \sim \lambda, \quad \frac{S_{n+2} - S}{S_{n+1} - S} \sim \lambda,$$

de manera que $(S_{n+1} - S)^2 \sim (S_{n+2} - S)(S_n - S)$. Esto nos lleva a que

$$S \sim \frac{S_{n+2}S_n - S_{n+1}^2}{S_{n+2} - 2S_{n+1} + S_n} = S_n - \frac{(\Delta S_n)^2}{S_{n+2} - 2S_{n+1} + S_n} = S_n - \frac{(\Delta S_n)^2}{\Delta^2 S_n} = T_n.$$

- En el teorema 1.1 no se supone que $\{S_n\}_{n=0}^\infty$ converge hacia S . En el caso de que $S = \lim_{n \rightarrow \infty} S_n$, el teorema afirma entonces que $\{T_n\}_{n=0}^\infty$ converge también hacia S y más rápidamente.

Para ilustrar estos puntos, podemos considerar el siguiente ejemplo elemental, [42]. La sucesión

$$S_0 = \frac{1}{2}, \quad S_{n+1} = e^{-S_n}, \quad n = 0, 1, \dots, \quad (1.4)$$

aproxima, mediante la iteración de punto fijo, la solución de la ecuación $x = e^{-x}$, véase la figura 1.1. Tanto la existencia de dicha solución como la convergencia de (1.4) hacia ella están

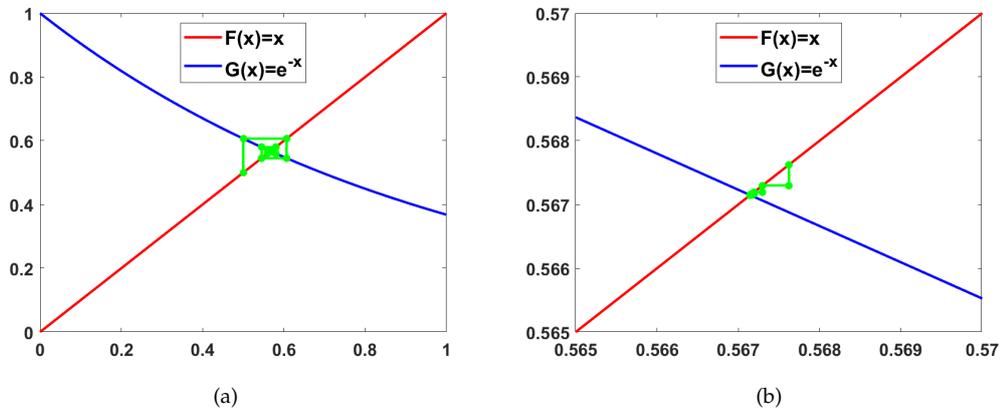


Figura 1.1: Representación conjunta de las funciones $F(x) = x$ y $G(x) = e^{-x}$.
 (a) Gráfico de telaraña para S_n ; (b) gráfico de telaraña para T_n dado por (1.1).

garantizadas por los clásicos teoremas de punto fijo. La segunda columna de la tabla 1.1 proporciona los primeros valores de la sucesión (1.4). Un mayor avance en la iteración sugiere tomar $\tilde{S} = 0.567143290$ como valor aproximado del límite S . Ello permite generar la tercera columna, donde el signo de los errores refleja el comportamiento de los iterantes en torno a \tilde{S} , ilustrado en el gráfico de telaraña de la figura 1.1(a). Por último, la sucesión de la cuarta columna converge a un valor de λ (cf. Teorema 1.1) cercano a $-\tilde{S} \approx G'(\tilde{S})$.

Por su parte, la tabla 1.2 proporciona los resultados asociados a la sucesión $\{T_n\}_{n=0}^\infty$. La aceleración de la convergencia se observa comparando las terceras columnas de ambas tablas (nótese el decrecimiento monótono de los errores en la tabla 1.2, véase figura 1.1(b)). En el capítulo 2 volveremos a este ejemplo para un último comentario referido a la última columna de la tabla 1.2.

1.2.2. Extrapolación de Richardson

Tal y como se menciona en [18], la teoría moderna de transformación de sucesiones para acelerar la convergencia tiene sus orígenes en el método de Aitken y en el de Richardson

n	S_n	$e_n = S_n - S$	$\Delta_n = \frac{e_n}{e_{n-1}}$
0	5e-01	-6.7143289999999967e-02	
1	6.0653065971263342e-01	3.9387369712633458e-02	-5.8661661817038568e-01
2	5.4523921189260505e-01	-2.1904078107394920e-02	-5.5611934148446607e-01
3	5.7970309487806826e-01	1.2559804878068292e-02	-5.7340029635066192e-01
4	5.6006462793890188e-01	-7.0786620610980844e-03	-5.6359649929424604e-01
5	5.7117214897721513e-01	4.0288589772151662e-03	-5.6915543395642554e-01
6	5.6486294698032347e-01	-2.2803430196765007e-03	-5.6600219381536221e-01
7	5.6843804757006622e-01	1.2947575700662517e-03	-5.6779070468526771e-01
8	5.6640945274692078e-01	-7.3383725307918368e-04	-5.6677579652354060e-01
9	5.6755963426224243e-01	4.1634426224246290e-04	-5.6735231210391801e-01
10	5.6690721293547142e-01	-2.3607706452855037e-04	-5.6702370114822953e-01

Tabla 1.1: Convergencia de la iteración del método de punto fijo (1.4). El valor del límite es aproximadamente $\tilde{S} = 0.567143290$.

n	T_n	$E_n = T_n - S$	$\Delta_n = \frac{E_n}{E_{n-1}}$
0	5.67252038878968e-01	1.08748878968168e-04	
1	5.67178136754967e-01	3.48467549667886e-05	3.204332338633913e-01
2	5.67154521727493e-01	1.12317274933993e-05	3.223177453425412e-01
3	5.67146898664407e-01	3.60866440751728e-06	3.212920193833083e-01
4	5.67144451783966e-01	1.16178396569744e-06	3.219429225054296e-01
5	5.67143663825010e-01	3.73825010058226e-07	3.217680920857028e-01
6	5.67143410545212e-01	1.20545211701462e-07	3.224642772902907e-01
7	5.67143329046794e-01	3.90467941313588e-08	3.239182509219931e-01
8	5.67143302838298e-01	1.28382983133335e-08	3.287926345539070e-01
9	5.67143294407277e-01	4.40727698780563e-09	3.432913677686050e-01
10	5.67143291695612e-01	1.69561231722071e-09	3.847301456006166e-01

Tabla 1.2: Convergencia de la sucesión $\{T_n\}_{n=0}^{\infty}$ obtenida con el método Δ^2 de Aitken (1.1). El valor del límite es aproximadamente $\tilde{S} = 0.567143290$.

(1881-1953), [47]. En el segundo caso, el método de extrapolación puede formularse de varias maneras. Aquí adoptaremos la utilizada en [56]. En muchas situaciones, la sucesión a extrapolar procede de evaluar una función s

$$S_n = s(h_n)$$

para cierta sucesión $\{h_n\}_{n=0}^{\infty}$, que decrece hacia cero, y de manera que

$$\lim_{h \rightarrow 0} s(h) = S \implies \lim_{n \rightarrow \infty} S_n = S.$$

Supongamos que s admite un desarrollo, para h suficientemente pequeño, de la forma

$$s(h) = S + \sum_{k=1}^l \alpha_k h^{\sigma_k}, \quad (1.5)$$

1 Introducción

con ciertos coeficientes α_k y exponentes $\sigma_k \neq 0$, tales que $h^{\sigma_k} \geq h^{\sigma_{k+1}}$, $k \geq 1$. Dado $w \in (0, 1)$, utilizando (1.5) podemos escribir, si $h' = wh$,

$$R(h', h) = \frac{s(wh) - w^{\sigma_1} s(h)}{1 - w^{\sigma_1}} = S + \sum_{k=2}^l \alpha'_k h^{\sigma_k}, \quad (1.6)$$

con $\alpha'_k = \frac{\alpha_k(w^{\sigma_k} - w^{\sigma_1})}{1 - w^{\sigma_1}}$. Observemos que, con respecto a (1.5), la función dada por (1.6) mejora la aproximación a S al eliminar el primer término $\alpha_1 h^{\sigma_1}$. Por otro lado, la función R en (1.6) tiene un desarrollo similar al de (1.5), de forma que el procedimiento de mejora puede aplicarse a R para ir eliminando las potencias dominantes h^{σ_k} de forma sucesiva. Ello sugiere la siguiente formulación del proceso: dado $R_0(h) = s(h)$, generamos las funciones

$$R_1(h, w) = \frac{R_0(wh) - w^{\sigma_1} R_0(h)}{1 - w^{\sigma_1}},$$

$$R_n(h, w) = \frac{R_{n-1}(hw, w) - w^{\sigma_n} R_{n-1}(h, w)}{1 - w^{\sigma_n}}, \quad n \geq 2, \quad (1.7)$$

que van eliminando sucesivamente las potencias en h^{σ_n} , $n \geq 2$. La recurrencia (1.7) permite además, al menos en teoría, extender la aplicación del procedimiento para el caso de funciones s en las que (1.5) se sustituye por un desarrollo asintótico

$$s(h) \sim S + \sum_{k=1}^{\infty} \alpha_k h^{\sigma_k}, \quad h \downarrow 0, \quad (1.8)$$

en el sentido de que para $l \geq 1$

$$s(h) = S + \sum_{k=1}^l \alpha_k h^{\sigma_k} + o(h^{\sigma_{l+1}}), \quad h \downarrow 0. \quad (1.9)$$

La formulación (1.7) es la base de la extrapolación de Richardson. Fijado $w \in (0, 1)$ y h_0 , sean $h_m = h_0 w^m$, $m = 0, 1, \dots$, de modo que $S_m = s(h_m)$, $m \geq 0$. Se tiene entonces el algoritmo:

1. Se define $R_0^{(j)} = R_0(h_j) = S_j$, $j = 0, 1, \dots$,

2. dado $C_n = w^{\sigma_n}$, calculamos

$$R_n^{(j)} = \frac{R_{n-1}^{(j+1)} - C_n R_{n-1}^{(j)}}{1 - C_n}, \quad j = 0, 1, \dots, n, \quad n \geq 1.$$

Las aproximaciones a S dadas por los $R_n^{(j)}$ suelen disponerse en forma matricial con la llamada Tabla de Romberg, donde los elementos $R_n^{(j)}$ se disponen según la estructura de tres términos

$$\begin{array}{ccc} R_{n-1}^{(j)} & & \\ & \searrow & \\ R_{n-1}^{(j+1)} & \rightarrow & R_n^{(j)} \end{array}$$

(cf. tabla 1.3). Alternativamente, podemos usar la estructura

$$\begin{array}{ccc} R_{n-1}^{(j)} & \rightarrow & R_n^{(j)} \\ & \nearrow & \\ R_{n-1}^{(j+1)} & & \end{array}$$

generando en este caso una tabla de Romberg donde el elemento $R_n^{(j)}$ se dispone en la posición (j, n) .

$n = 0$	$n = 1$	$n = 2$	$n = 3$
$R_0^{(0)}$			
\searrow			
$R_0^{(1)} \rightarrow$	$R_1^{(0)}$		
\searrow	\searrow		
$R_0^{(2)} \rightarrow$	$R_1^{(1)} \rightarrow$	$R_2^{(0)}$	
\searrow	\searrow	\searrow	
$R_0^{(3)} \rightarrow$	$R_1^{(2)} \rightarrow$	$R_2^{(1)} \rightarrow$	$R_3^{(0)}$
\vdots	\vdots	\vdots	\vdots

Tabla 1.3: Tabla de Romberg.

Richardson fue pionero en el uso de la extrapolación para acelerar la convergencia en sucesiones, una idea en principio diferente (si bien, como veremos, relacionada) a la basada en el concepto de transformación de sucesiones y su núcleo, iniciada por Aitken y desarrollada más adelante por Shanks (1917-1996), Wynn (1931-2017) y otros. La aplicación específica de la extrapolación de Richardson requiere entonces resolver algunos aspectos como:

1. La adaptación del problema a la formulación (1.5) o, más generalmente, (1.8)-(1.9).
2. Los elementos necesarios del algoritmo (como los términos C_n) frente a otros, típicamente artificiales (como la sucesión $\{h_j\}_{j=1}^\infty$).
3. La visualización de la aceleración de la convergencia en la Tabla de Romberg.

Podemos también, como se hizo en el primer ejemplo, ilustrar esta presentación del método de Richardson con un problema elemental, [56], para acelerar la convergencia de la aproximación a la derivada de una función f en un punto x_0 mediante la fórmula basada en diferencias centradas

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h}, \quad h > 0.$$

Para adaptar este caso a la descripción anterior, definimos

$$s(h) = \frac{f(x_0 + h) - f(x_0 - h)}{2h}, \quad S = f'(x_0).$$

El desarrollo requerido para $s(h)$ procede aquí del teorema de Taylor. Sea $l > 0$ y supongamos que f es de clase C^{2l+3} en un entorno de x_0 . Entonces, para $h \downarrow 0$, podemos escribir,

$$s(h) = \sum_{j=0}^l \frac{h^{2j}}{(2j+1)!} f^{(2j+1)}(x_0) + O(h^{2l+2}),$$

que es un desarrollo de la forma (1.9) con

$$\alpha_k = \frac{f^{(2k+1)}(x_0)}{(2k+1)!}, \quad \sigma_k = 2k, \quad k = 1, \dots, l.$$

Estamos entonces en condiciones de aplicar la extrapolación de Richardson a nuestro problema. Un valor típico para generar la tabla de Romberg es $w = 1/2$ (de manera que $C_n = 1/4^n$), [50]. La visualización de la convergencia del proceso para $f(x) = 2\sqrt{1+x}$, $x_0 = 0$, viene dada en la tabla 1.4 (donde, en el paso 1 del algoritmo, se ha tomado $h_0 = 1$). La tabla muestra los errores en la diferenciación numérica correspondientes a los valores de la tabla de Romberg y el valor exacto $S = f'(0) = 1$. Típicamente, la aceleración de la convergencia

3.527618e-02						
8.034340e-03	1.046274e-03					
1.966601e-03	5.597903e-05	1.004061e-05				
4.891176e-04	3.376709e-06	1.301127e-07	2.719678e-08			
1.221225e-04	2.092182e-07	1.947856e-09	8.650614e-11	1.980860e-11		
3.052084e-05	1.304791e-08	3.010703e-11	3.334000e-13	4.662937e-15	1.465494e-14	

Tabla 1.4: Errores de la diferenciación numérica de $f(x) = 2\sqrt{1+x}$

proporcionada por el método y a partir de la sucesión original (primera columna de la tabla de Romberg) puede observarse de dos formas: por columnas (la sucesión de una columna converge más rápidamente que la de la columna anterior y más lentamente que la de la columna siguiente) o bien por diagonales (los errores de una sucesión diagonal son menores que los de la anterior). Además, en general, la aceleración dada por las diagonales es mayor que la de las columnas. Discutiremos estos aspectos de convergencia en el capítulo 2.

1.2.3. Extrapolación vectorial

El desarrollo natural de la teoría llevó a la extensión de la extrapolación a procesos vectoriales a partir de la generalización al caso vectorial del algoritmo ε de Wynn [72], continuado por la variante propuesta por Brezinski, [13, 15, 70] y los llamados métodos polinómicos [60, 59]. Completamos la introducción con una descripción básica de uno de estos últimos, el llamado método de extrapolación de polinomio mínimo (MPE), que será desarrollada con más detalle en el capítulo 3.

Supongamos que el sistema lineal $N \times N$

$$Ax = b \tag{1.10}$$

admite una solución única $x = S$, y que podemos reescribir (1.10) en forma de sistema de punto fijo

$$x = Tx + d, \tag{1.11}$$

para cierto término independiente d y cierta matriz T que no tiene a $\lambda = 1$ como autovalor. A partir de un vector inicial S_0 generamos la sucesión

$$S_{m+1} = TS_m + d, \quad m = 0, 1, \dots, \tag{1.12}$$

que convergerá hacia S siempre que el radio espectral de T sea menor que uno, mientras que en el caso de divergencia, el vector S tiene el papel de antilímite de la sucesión $\{S_m\}_{m=0}^{\infty}$ (véase el capítulo 2). Observemos que de (1.11), (1.12) se tiene que, si $\varepsilon_m = S_m - S$

$$\varepsilon_{m+1} = T\varepsilon_m, \quad m = 0, 1, \dots$$

Por otro lado para todo i, n

$$T^i \varepsilon_n = \varepsilon_{n+i}. \quad (1.13)$$

Sea

$$P(z) = \sum_{i=0}^N p_i z^i, \quad p_N = 1,$$

el polinomio característico de T . Como, por el teorema de Cayley-Hamilton, es $p(T) = 0$, se tiene en particular que $p(T)\varepsilon_n = 0$, lo que, utilizando (1.13), implica que

$$0 = \sum_{i=0}^N p_i T^i \varepsilon_n = \sum_{i=0}^N p_i \varepsilon_{n+i} = \sum_{i=0}^N p_i S_{n+i} - S \sum_{i=0}^N p_i,$$

y como $\lambda = 1$ no es autovalor de T , entonces $p(1) = \sum_{i=0}^N p_i \neq 0$, de modo que

$$S = \frac{\sum_{i=0}^N p_i S_{n+i}}{\sum_{i=0}^N p_i}, \quad (1.14)$$

para n arbitrario. Definiendo,

$$\tilde{p}_i = \frac{p_i}{\sum_{j=0}^N p_j}, \quad i = 0, \dots, N,$$

entonces la fórmula (1.14) puede reescribirse como

$$S = \sum_{i=0}^N \tilde{p}_i S_{n+i}, \quad \sum_{i=0}^N \tilde{p}_i = 1. \quad (1.15)$$

De este modo, el proceso alcanzaría, teóricamente, la solución S mediante (1.15) y conocidos los coeficientes p_i del polinomio característico. (En la descripción del capítulo 3 se utilizará otro polinomio).

La determinación de los p_i puede hacerse de forma relativamente eficiente del siguiente modo. Definimos la sucesión $\{u_n\}_{n=0}^{\infty}$ con $u_m = \Delta S_m$, $m = 0, 1, \dots$. De (1.12) se tiene de nuevo la propiedad

$$T^i u_n = u_{n+i},$$

para n, i arbitrarios. Como $p(T)u_n = 0$ y $p_N = 1$, entonces

$$\sum_{i=0}^{N-1} p_i u_{n+i} = -u_{n+N}. \quad (1.16)$$

1 Introducción

De esta manera, los coeficientes p_0, \dots, p_{N-1} pueden calcularse como solución del sistema (1.16) para N valores $n = 0, 1, \dots, N-1$.

El procedimiento así descrito para calcular S exactamente no es aplicable en la práctica si N es muy grande. (No tiene mucho sentido, por ejemplo, intentar resolver un sistema lineal como (1.10) a partir de otro, (1.16), del mismo tamaño y sin, a priori, ser más sencillo.) El argumento puede, sin embargo, utilizarse para construir un método de aproximación a S . Elegimos K mucho menor que el tamaño N del sistema y calculamos coeficientes c_0, \dots, c_{K-1} a partir de un sistema del tipo (1.16)

$$\sum_{i=0}^{K-1} c_i u_{n+i} = -u_{n+K}, \quad (1.17)$$

que se resuelve, en general, en el sentido de mínimos cuadrados. Obtenidos c_0, \dots, c_{K-1} , fijamos $c_K = 1$ y definimos

$$\gamma_i = \frac{c_i}{\sum_{j=0}^K c_j}, \quad i = 0, \dots, K, \quad (1.18)$$

siempre que $\sum_{j=0}^K c_j \neq 0$. El argumento anterior, que nos llevó a (1.15), hace razonable pensar que

$$S_{n,K} = \sum_{i=0}^K \gamma_i S_{n+i}, \quad (1.19)$$

es una aproximación a S .

Como ilustración elemental podemos considerar el sistema (1.10) presentado en [60] con

$$A = \begin{pmatrix} 2 & 1 & 3 & 4 \\ 1 & -3 & 1 & 5 \\ 3 & 1 & 6 & -2 \\ 4 & 5 & -2 & -1 \end{pmatrix}, \quad b = (10, 4, 8, 6)^T, \quad (1.20)$$

de modo que $S = (1, 1, 1, 1)^T$. En (1.12), T será en este caso la matriz del método de Gauss-Seidel aplicado a (1.20), es decir

$$T = -(L+D)^{-1}U, \quad d = (L+D)^{-1}b$$

siendo $A = L + D + U$, con

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 4 & 5 & -2 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & 1 & 3 & 4 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Puede comprobarse que los autovalores de T son

$$\lambda_1 = \bar{\lambda}_2 = -2.3500 \pm 2.0506i, \quad \lambda_3 = -0.0228, \quad \lambda_4 = 0,$$

por lo que (1.12) no es convergente. Tomemos $K = 2$ e implementemos el procedimiento (1.17)-(1.19). Para diferentes valores de n , el error $S_{n,2} - S$ en norma del máximo se muestra en la

tabla 1.5.

n	$\ \mathbf{T}_k^{(n)} - \mathbf{x}\ _\infty$
1	7.2010e-03
2	1.6448e-04
3	3.7581e-06
4	8.5866e-08
5	1.9623e-09
6	4.2519e-11
7	5.6843e-12

Tabla 1.5: Errores en norma del máximo para el método (1.17)-(1.19) con $K = 2$ y el sistema (1.20).

Observamos que, a diferencia de la sucesión original $\{S_n\}_{n=0}^\infty$ dada por (1.12), la transformada $\{S_{n,2}\}_{n=0}^\infty$ converge a la solución S .

2 Extrapolación escalar

El presente capítulo está dedicado a los métodos de extrapolación para procesos secuenciales escalares. Se divide en cuatro secciones. La primera es introductoria; presenta algunas definiciones y elementos básicos de la teoría, que utilizaremos a lo largo del capítulo y que tendrán una extensión natural al tratar la extrapolación vectorial del capítulo 3.

Como ya se mencionó anteriormente, no es posible un análisis exhaustivo de toda la metodología presente en la literatura para la extrapolación escalar. Se ha optado entonces por una estructura alternativa, consistente en la descripción de dos grandes familias de métodos (que, como veremos, contienen aspectos comunes) junto con el análisis de sus representantes más utilizados en las aplicaciones. De esta manera, dedicamos la segunda sección al algoritmo de extrapolación general, destacando aquí la llamada transformación de y el algoritmo ε , mientras que la sección 2.3 se ocupa de los métodos polinómicos y racionales, destacando aquí el método de extrapolación de Richardson polinómico y el algoritmo ρ . Ambas secciones siguen una distribución similar, basada en una presentación, un análisis de convergencia, la implementación y algunas aplicaciones. Por último, cerraremos el capítulo con la sección 2.4, donde resumimos otros procedimientos de extrapolación escalar de relevancia, surgidos a partir de la búsqueda de una mejora en el rendimiento de los métodos de las secciones anteriores, bien en aspectos teóricos de convergencia, bien por motivos de eficiencia computacional.

Si bien la mayor parte de los resultados son válidos para el caso complejo (lo que permite su utilización en importantes aplicaciones, como la aceleración de la convergencia en series de Fourier) en este capítulo (y en el siguiente) consideraremos, por sencillez, sucesiones reales $\{S_n\}_{n=0}^{\infty}$.

2.1. Introducción

2.1.1. Algunos tipos de convergencia

Como se mencionó en la introducción, no es esperable conseguir una transformación que acelere la convergencia de todas las sucesiones. En la sección 2.1.2 veremos que la estrategia suele pasar por adaptar el método de extrapolación al tipo de sucesión a transformar, utilizando el concepto de núcleo. Por ello, destacamos en primer lugar varias formas de convergencia de sucesiones de uso común en la construcción y análisis de los esquemas de aceleración.

Definición 2.1. Sea $\{S_n\}_{n=0}^{\infty}$ una sucesión real y $S \in \mathbb{R}$, tales que

$$\rho = \lim_{n \rightarrow \infty} \left(\frac{S_{n+1} - S}{S_n - S} \right),$$

es finito. Entonces,

- si $0 < |\rho| < 1$, se dice que $\{S_n\}_{n=0}^{\infty}$ converge linealmente,

2 Extrapolación escalar

- si $\rho = 1$, se dice que la convergencia es logarítmica,
- si $\rho = 0$, se dice que la convergencia es hiperlineal,
- cuando $|\rho| > 1$, se dice que la sucesión es divergente.

Ejemplo 2.1. Un ejemplo clásico de convergencia lineal viene dada por la sucesión de sumas parciales de una serie geométrica

$$S_n = \sum_{k=0}^n \lambda^k = \frac{1 - \lambda^{n+1}}{1 - \lambda} = \frac{1}{1 - \lambda} - \frac{\lambda}{1 - \lambda} \lambda^n, \quad |\lambda| < 1, \quad (2.1)$$

con $S = \frac{1}{1 - \lambda}$. Entonces $\rho = \lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} = \lambda$.

Más generalmente, sucesiones de la forma

$$S_n = S + \lambda^n \sum_{k=0}^m \alpha_k n^{\sigma-k}, \quad (2.2)$$

con $0 < |\lambda| < 1$, $\sigma \in \mathbb{R}$, $m \in \mathbb{N} \cup \{0\}$, $\alpha_0 \neq 0$, tienen convergencia lineal pues

$$\frac{S_{n+1} - S}{S_n - S} = \lambda \frac{\alpha_0 (n+1)^\sigma + \dots + \alpha_m (n+1)^{\sigma-m}}{\alpha_0 n^\sigma + \dots + \alpha_m n^{\sigma-m}}.$$

Así, por ejemplo, la sucesión (2.1) es de la forma (2.2) con $S = \frac{1}{1 - \lambda}$, $m = \sigma = 0$, $\alpha_0 = \frac{-\lambda}{1 - \lambda}$.

La aproximación (2.2) puede sustituirse por un desarrollo asintótico

$$S_n \sim S + \lambda^n \sum_{k=0}^{\infty} \alpha_k n^{\sigma-k}, \quad n \rightarrow \infty,$$

en el sentido de que para todo $m \geq 1$

$$S_n = S + \lambda^n \sum_{k=0}^m \alpha_k n^{\sigma-k} + O(n^{\sigma-(m+1)}), \quad n \rightarrow \infty.$$

Ejemplo 2.2. Asimismo, sucesiones de la forma

$$S_n = S + \sum_{k=0}^m \alpha_k n^{\sigma-k}, \quad (2.3)$$

con $\alpha_0 \neq 0$, $\sigma \neq 0, 1, \dots, m$, o, más generalmente

$$S_n \sim S + \sum_{k=0}^{\infty} \alpha_k n^{\sigma-k}, \quad \sigma \neq 0, 1, 2, \dots,$$

son ejemplos de convergencia logarítmica.

Ejemplo 2.3. Por último, las llamadas sucesiones factoriales

$$S_n = S + \frac{\lambda^n}{(n!)^r} \sum_{k=0}^m \alpha_k n^{\sigma-k} \quad (2.4)$$

con $\alpha_0 \neq 0$, $r \in \mathbb{N}$, (o aquéllas que asintóticamente se comportan como (2.4) cuando $n \rightarrow \infty$) ilustran la convergencia hiperlineal.

En algunas ocasiones, los métodos de extrapolación pueden transformar sucesiones divergentes en convergentes. Ello motiva el concepto de **antilímite**. Por ejemplo, la sucesión de sumas parciales de la serie geométrica (2.1) diverge cuando $|\lambda| > 1$, mientras que $S = \frac{1}{1-\lambda}$ tiene sentido para todo λ con $\lambda \neq 1$. Así, en el caso divergente, $S = \frac{1}{1-\lambda}$ se denominaría el antilímite de la sucesión $\{S_n\}_{n=0}^{\infty}$. Sin embargo, la aceleración generada por el método de Aitken para (2.1) satisface,

$$T_n = S_n - \frac{(\Delta S_n)^2}{\Delta^2 S_n} = \frac{1}{1-\lambda} - \frac{\lambda^{n+1}}{1-\lambda} - \frac{(\lambda^{n+1})^2}{\lambda^{n+1}(1-\lambda)} = \frac{1}{1-\lambda} = S, \quad \lambda \neq 1,$$

es decir, es exacta para S , transformando así el proceso divergente de S_n en convergente.

2.1.2. Métodos de extrapolación

Definición 2.2. Sea l un entero no negativo. Un método de extrapolación (o aceleración) escalar es una transformación $\phi : \mathbb{R}^{l+1} \rightarrow \mathbb{R}$ que, a partir de cualquier sucesión $\{S_n\}_{n=0}^{\infty}$, genera la sucesión transformada $\{T_n\}_{n=0}^{\infty}$ con

$$T_n = \phi(S_n, S_{n+1}, \dots, S_{n+l}), \quad n = 0, 1, 2, \dots$$

El entero l se llama orden de la transformación.

Algunas variantes de la definición, que aparecen en la literatura y que se utilizarán en este trabajo, son las siguientes:

1. La extrapolación puede escribirse también como, [56]

$$T_n = \phi(S_n, \dots, S_{n+l}) = \sum_{i=0}^l \theta_{n,i} S_{n+i}, \quad (2.5)$$

con coeficientes $\theta_{n,i} = \theta_{n,i}(S_n)$ que, en general, dependen de $\{S_n\}_{n=0}^{\infty}$, pero que satisfacen

$$\sum_{i=0}^l \theta_{n,i} = 1. \quad (2.6)$$

2. En algunos casos, [70], la transformación se genera a partir de una recurrencia con doble índice

$$T_k^{(n)} = \phi(S_n, \dots, S_{n+l})$$

con $k > 0$, $l = l(k)$ y $T_0^{(n)} = S_n$. El índice k es una medida de la complejidad de la transformación. Los términos de la nueva sucesión suelen disponerse en forma de tabla (cf. ejemplo 1.2.2)

$$\begin{array}{cccc} T_0^{(0)} & T_1^{(0)} & \dots & T_m^{(0)} \\ T_0^{(1)} & T_1^{(1)} & \dots & T_m^{(1)} \\ \vdots & \vdots & & \vdots \\ T_0^{(n)} & T_1^{(n)} & \dots & T_m^{(n)} \end{array}.$$

2 Extrapolación escalar

Definición 2.3. Sea $\{S_n\}_{n=0}^{\infty}$ una sucesión escalar. Sea $\{T_n\}_{n=0}^{\infty}$ la sucesión obtenida mediante un método de extrapolación a partir de $\{S_n\}_{n=0}^{\infty}$. Supongamos que $S = \lim_{n \rightarrow \infty} T_n$ y, si existe $\lim_{n \rightarrow \infty} S_n$, entonces es igual a S . Se dice que $\{T_n\}_{n=0}^{\infty}$ converge más rápidamente que $\{S_n\}_{n=0}^{\infty}$ si

$$\lim_{n \rightarrow \infty} \left| \frac{T_n - S}{S_n - S} \right| = 0.$$

El término $R_n = \left| \frac{T_n - S}{S_n - S} \right|$ se llama factor de aceleración de $\{T_n\}_{n=0}^{\infty}$.

Antes vimos que el método de Aitken alcanzaba exactamente el valor S para la sucesión (2.1). El concepto se formaliza en la siguiente definición.

Definición 2.4. Supongamos que S es el límite o antilímite de una sucesión $\{S_n\}_{n=0}^{\infty}$. Se dice que el método de extrapolación ϕ es exacto para la sucesión $\{S_n\}_{n=0}^{\infty}$ si existe n_0 tal que para todo $n \geq n_0$,

$$\phi(S_n, \dots, S_{n+l}) = S.$$

Aquellas sucesiones exactas para ϕ constituyen lo que se llama el núcleo de la transformación.

Si bien la aplicación ϕ que define un método de extrapolación es, en general, no lineal, una propiedad deseable es la **casi-linealidad** o **invariancia por traslaciones**:

$$\phi(\alpha x_0 + \beta, \dots, \alpha x_l + \beta) = \alpha \phi(x_0, \dots, x_l) + \beta, \quad \alpha, \beta \in \mathbb{R}, \quad x = (x_0, \dots, x_l)^T \in \mathbb{R}^{l+1}. \quad (2.7)$$

Algunas hipótesis sobre ϕ pueden servir para identificar la propiedad (2.7). Por ejemplo, la siguiente condición suficiente utiliza la formulación (2.5)-(2.6), [56].

Proposición 2.1. Sea ϕ un método de extrapolación definido por (2.5)-(2.6) y verificando que los coeficientes $\theta_{n,i}$ son funciones homogéneas de grado cero de las diferencias

$$\theta_{n,i} = \theta_{n,i}(\Delta S_m).$$

Entonces ϕ satisface (2.7).

Observemos que si ϕ es diferenciable, la condición (2.7) implica que

$$\langle x, \nabla \phi(x) \rangle = \phi(x), \quad \langle e, \nabla \phi(x) \rangle = 1, \quad (2.8)$$

donde $\langle \cdot, \cdot \rangle$ denota el producto escalar euclideo en \mathbb{R}^{l+1} y $e \in \mathbb{R}^{l+1}$ es el vector con todas sus componentes iguales a uno. Así, la primera igualdad (2.8) permite identificar los coeficientes $\theta_{n,i}$ en (2.5) como

$$\theta_{n,i} = \frac{\partial \phi}{\partial x_i}(S_n, \dots, S_{n+l}), \quad i = 0, \dots, l, \quad (2.9)$$

mientras que (2.6) corresponde a la segunda condición de (2.8).

Ejemplo 2.4. El método de Aitken puede escribirse en la forma (2.5)-(2.6) con

$$T_n = \theta_{n,0} S_n + \theta_{n,1} S_{n+1}, \quad \theta_{n,0} = \frac{\Delta S_{n+1}}{\Delta S_{n+1} - \Delta S_n}, \quad \theta_{n,1} = \frac{-\Delta S_n}{\Delta S_{n+1} - \Delta S_n}, \quad (2.10)$$

y, aplicando la proposición 2.1, se tiene que (2.10) satisface la propiedad (2.7).

Estabilidad

Además del concepto de convergencia, la extrapolación, como todo proceso numérico, requiere formalizar la idea de estabilidad. Su motivación puede venir de nuevo del caso diferenciable. Observemos que si la sucesión a extrapolar $\{S_n\}_{n=0}^{\infty}$ es calculada computacionalmente como $\{\tilde{S}_n\}_{n=0}^{\infty}$ con

$$\tilde{S}_n = S_n + \varepsilon_n, \quad |\varepsilon_n| \leq \varepsilon,$$

para $\varepsilon > 0$, entonces la extrapolación se calcula como

$$\begin{aligned} \tilde{T}_n &= \phi(\tilde{S}_n, \dots, \tilde{S}_{n+l}) = \phi(S_n, \dots, S_{n+l}) + \sum_{i=0}^l \varepsilon_i \frac{\partial \phi}{\partial x_i}(S_n, \dots, S_{n+l}) + O(\varepsilon^2) \\ &= T_n + \sum_{i=0}^l \varepsilon_i \frac{\partial \phi}{\partial x_i}(S_n, \dots, S_{n+l}) + O(\varepsilon^2). \end{aligned}$$

Por tanto,

$$|\tilde{T}_n - T_n| \leq \varepsilon \sum_{i=0}^l \left| \frac{\partial \phi}{\partial x_i}(S_n, \dots, S_{n+l}) \right| + O(\varepsilon^2).$$

De este modo, el control de la extrapolación debe incorporar el término

$$\sum_{i=0}^l \left| \frac{\partial \phi}{\partial x_i}(S_n, \dots, S_{n+l}) \right| = \sum_{i=0}^l |\theta_{n,i}|.$$

Definición 2.5 ([56]). Se dice que el método de extrapolación (2.5)-(2.6) es estable para la sucesión $\{S_n\}_{n=0}^{\infty}$ si

$$\sup_n \Gamma_n < \infty, \quad \Gamma_n := \sum_{i=0}^l |\theta_{n,i}|.$$

En otro caso, se dice que es inestable.

Observamos que, a partir de (2.6),

$$\sum_{i=0}^l |\theta_{n,i}| \geq \left| \sum_{i=0}^l \theta_{n,i} \right| = 1 \implies \Gamma_n \geq 1, \quad \forall n.$$

La definición implica que la estabilidad del algoritmo depende también de la sucesión $\{S_n\}_{n=0}^{\infty}$ a extrapolar. Por ejemplo, en el caso del método de Aitken, de (2.10) tenemos

$$\Gamma_n = |\theta_{n,0}| + |\theta_{n,1}| = \frac{1 + |g_n|}{|1 - g_n|}, \quad g_n = \frac{\Delta S_{n+1}}{\Delta S_n}.$$

Así, para la sucesión de sumas parciales de la serie geométrica (2.1), se tiene que $g_n = \lambda$ para todo n , luego

$$\Gamma_n = \frac{1 + |g_n|}{|1 - g_n|} = \frac{1 + |\lambda|}{|1 - \lambda|}.$$

Y el método es estable, si bien Γ_n puede ser grande cuando λ está muy cerca de 1. Por otro lado, para la sucesión

$$S_n = \sum_{j=0}^n \frac{1}{(j+1)^2},$$

2 Extrapolación escalar

se tiene que $g_n = \left(\frac{n+2}{n+3}\right)^2$ y como $\lim_{n \rightarrow \infty} g_n = 1$, el método de Aitken no es estable para la sucesión $\{S_n\}_{n=0}^{\infty}$.

Reglas de construcción

Como ya se mencionó anteriormente, el concepto de sucesión exacta puede utilizarse para la construcción de métodos de extrapolación, [70], a partir de las reglas siguientes:

1. Se considera una sucesión modelo de la forma

$$\tilde{S}_n = S + \tilde{R}_n, \quad (2.11)$$

donde $\{\tilde{R}_n\}_{n=0}^{\infty}$, llamada sucesión de restos, determina la formulación del método. Se supone que $\{\tilde{R}_n\}_{n=0}^{\infty}$ admite una estructura como producto término a término de dos sucesiones $\{w_n\}_{n=0}^{\infty}$ y $\{z_n\}_{n=0}^{\infty}$, es decir $\tilde{R}_n = w_n z_n$, $n \geq 0$.

2. Supongamos que encontramos una aplicación lineal $\hat{\phi}$, que anule z_n , es decir $\hat{\phi}(z_n) = 0$. Esto implica que

$$S = \frac{\hat{\phi}(\tilde{S}_n/w_n)}{\hat{\phi}(1/w_n)}.$$

Entonces,

$$\phi(\tilde{S}_n, w_n) = \frac{\hat{\phi}(\tilde{S}_n/w_n)}{\hat{\phi}(1/w_n)} = S,$$

es decir, la aplicación ϕ es exacta para $\{\tilde{S}_n\}_{n=0}^{\infty}$.

3. La representación de la fórmula del método de extrapolación se obtiene en la práctica sustituyendo $\{\tilde{S}_n\}_{n=0}^{\infty}$ por una sucesión genérica $\{S_n\}_{n=0}^{\infty}$ a transformar

$$T_n = \phi(S_n, w_n) = \frac{\hat{\phi}(S_n/w_n)}{\hat{\phi}(1/w_n)}.$$

La manera en la que aparece esta última expresión puede ser diferente: de forma explícita, como solución de un sistema lineal o no lineal, etc., todo con respecto a la sucesión original $\{S_n\}_{n=0}^{\infty}$ que debe extrapolarse.

La estructura concreta de los factores z_n y w_n es diversa. En general, se busca cierta sencillez en la formulación que permita una implementación eficiente y estimaciones del error para la convergencia. Una manera típica de conseguir esto consiste en escribir

$$\tilde{R}_n = w_n \sum_{j=0}^m C_j f_j(n), \quad n = 0, 1, \dots, \quad (2.12)$$

de modo que z_n es combinación lineal de ciertas funciones f_j , $j = 0, \dots, m$, para las que puede encontrarse una aplicación $\hat{\phi}$ que anule \tilde{R}_n/w_n de forma sencilla. Los coeficientes C_j suelen ser variables auxiliares y no juegan un papel relevante en la construcción del método de extrapolación. Una vez elegidas las funciones f_j , la formulación se completa con la elección de la sucesión w_n , con el objetivo de facilitar la búsqueda de estimaciones del error.

2.2. El algoritmo de extrapolación general (GEA) y la transformación de Shanks

Las reglas anteriores constituyen la base del método más general de extrapolación, o método GEA, introducido por Håvie, [32] y Brezinski, [14].

Utilizando (2.12) y eligiendo $w_n = 1$ para todo n , la sucesión modelo (2.11) es de la forma

$$\tilde{S}_n = S + \sum_{j=1}^k C_j f_j(n), \quad n = 0, 1, \dots \quad (2.13)$$

Para k fijo y funciones $f_j(n)$, $j = 1, \dots, k$, arbitrarias pero conocidas, la fórmula (2.13) determina un sistema lineal de $k + 1$ ecuaciones para las incógnitas S, C_1, \dots, C_k ,

$$\begin{cases} \tilde{S}_n &= S + C_1 f_1(n) + \dots + C_k f_k(n), \\ \tilde{S}_{n+1} &= S + C_1 f_1(n+1) + \dots + C_k f_k(n+1), \\ &\vdots \\ \tilde{S}_{n+k} &= S + C_1 f_1(n+k) + \dots + C_k f_k(n+k), \end{cases}$$

del cual, suponiendo que tiene solución única, obtenemos mediante las fórmulas de Cramer,

$$S = \frac{\begin{vmatrix} \tilde{S}_n & f_1(n) & f_2(n) & \dots & f_k(n) \\ \tilde{S}_{n+1} & f_1(n+1) & f_2(n+1) & \dots & f_k(n+1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{S}_{n+k} & f_1(n+k) & f_2(n+k) & \dots & f_k(n+k) \end{vmatrix}}{\begin{vmatrix} 1 & f_1(n) & f_2(n) & \dots & f_k(n) \\ 1 & f_1(n+1) & f_2(n+1) & \dots & f_k(n+1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & f_1(n+k) & f_2(n+k) & \dots & f_k(n+k) \end{vmatrix}}.$$

Siguiendo las reglas de construcción anteriores, tenemos la fórmula general de extrapolación de una sucesión $\{S_n\}_{n=0}^{\infty}$:

$$T_n = E_k(S_n) = \frac{\begin{vmatrix} S_n & f_1(n) & f_2(n) & \dots & f_k(n) \\ S_{n+1} & f_1(n+1) & f_2(n+1) & \dots & f_k(n+1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_{n+k} & f_1(n+k) & f_2(n+k) & \dots & f_k(n+k) \end{vmatrix}}{\begin{vmatrix} 1 & f_1(n) & f_2(n) & \dots & f_k(n) \\ 1 & f_1(n+1) & f_2(n+1) & \dots & f_k(n+1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & f_1(n+k) & f_2(n+k) & \dots & f_k(n+k) \end{vmatrix}}. \quad (2.14)$$

Conviene destacar que el carácter general de la transformación (2.14) viene determinado por la libertad de elección de las funciones f_j y el orden k . Así, gran parte de los métodos clásicos de extrapolación escalar se pueden formular como (2.14) utilizando convenientes funciones

2 Extrapolación escalar

base, véase [32]. Por ejemplo, si $k = 1$ y $f_1(n) = \Delta S_n$, se tiene

$$E_1(S_n) = \frac{\begin{vmatrix} S_n & \Delta S_n \\ S_{n+1} & \Delta S_{n+1} \end{vmatrix}}{\begin{vmatrix} 1 & \Delta S_n \\ 1 & \Delta S_{n+1} \end{vmatrix}} = \frac{\Delta S_{n+1}}{\Delta S_{n+1} - \Delta S_n} S_n - \frac{\Delta S_n}{\Delta S_{n+1} - \Delta S_n} S_{n+1},$$

que es el método de Aitken.

Este carácter general puede, sin embargo, ser un inconveniente para la implementación y el análisis de convergencia. Muchos métodos se formulan siguiendo otros caminos que permiten diseñar una programación más eficiente y analizar la convergencia de forma más directa. La comprobación de que los métodos pueden formularse como (2.14) suele venir a posteriori. El ejemplo más claro de esta situación es la transformación de Shanks, como veremos más adelante.

El algoritmo de implementación de (2.14), llamado E-algoritmo, [32], debe su construcción a la búsqueda de procedimientos para computar $E_k(S_n)$ sin utilizar determinantes. Su formulación es la siguiente:

$$E_0^{(n)} = S_n; \quad f_{0,i}^{(n)} = f_i(n), \quad i = 1, 2, \dots, \quad n = 0, 1, \dots \quad (2.15)$$

Para $k = 1, 2, \dots, m = 0, 1, \dots$

$$E_k^{(n)} = \frac{E_{k-1}^{(n)} f_{k-1,k}^{(n+1)} - E_{k-1}^{(n+1)} f_{k-1,k}^{(n)}}{f_{k-1,k}^{(n+1)} - f_{k-1,k}^{(n)}}, \quad (2.16)$$

donde

$$f_{k,i}^{(n)} = \frac{f_{k-1,i}^{(n)} f_{k-1,k}^{(n+1)} - f_{k-1,i}^{(n+1)} f_{k-1,k}^{(n)}}{f_{k-1,k}^{(n+1)} - f_{k-1,k}^{(n)}}, \quad i = k+1, k+2, \dots, \quad (2.17)$$

con $f_{k,i}^{(n)} = 0$, si $k \geq i$.

Observemos que, conocidos los coeficientes (2.17), la fórmula (2.16) para $E_k^{(n)}$ involucra $E_{k-1}^{(n)}, E_{k-1}^{(n+1)}$, lo que permite disponerla en la forma de la tabla 2.1, donde el subíndice k

$E_0^{(0)} = S_0$	→	$E_1^{(0)}$	→	$E_2^{(0)}$	→	...	→	$E_n^{(0)}$	
	↗		↗		↗		↗		
$E_0^{(1)} = S_1$	→	$E_1^{(1)}$	→	$E_2^{(1)}$	→	⋮			
	↗		↗		↗				$E_{k-1}^{(n)} \rightarrow E_k^{(n)}$
⋮	→	⋮	→	⋮					↗
	↗		↗		↗				$E_{k-1}^{(n+1)}$
$E_0^{(n-1)} = S_{n-1}$	→	$E_1^{(n-1)}$							
	↗								
$E_0^{(n)} = S_n$									

Tabla 2.1: Tabla del E-algoritmo y esquema de generación.

recorre las columnas y el superíndice n las filas. Las flechas indican el orden de generación de la tabla y a la derecha se dispone la ‘molécula’ computacional derivada de (2.16). La

identificación con (2.14) viene dada por el siguiente resultado, [32].

Proposición 2.2. Si se define $E_k^{(n)} = \frac{E_{k-1}^{(n)} f_{k-1,k}^{(n+1)} - E_{k-1}^{(n+1)} f_{k-1,k}^{(n)}}{f_{k-1,k}^{(n+1)} - f_{k-1,k}^{(n)}}$, $k, n = 0, 1, \dots$, entonces $E_k^{(n)} = E_k(S_n)$.

Demostración:

En la prueba se empleará la **identidad de Sylvester**, [67]: Dada C una matriz cuadrada, se denota por C_{ij} la submatriz de C que se obtiene eliminando la fila i y la columna j y por $C_{i'i',j'j'}$ a la submatriz de C que se obtiene eliminando las filas i, i' y las columnas j, j' , $i \neq i', j \neq j'$. Entonces si $|\cdot|$ representa el determinante de la correspondiente matriz se tiene que

$$|C| |C_{i'i',j'j'}| = |C_{ij}| |C_{i'j'}| - |C_{ij'}| |C_{i'j}|.$$

Para simplificar la notación, identificamos cada determinante por el primer elemento de cada columna. Así, escribimos (2.14) como

$$E_k(S_n) = \frac{|S_n f_1(n) \dots f_k(n)|}{|1 f_1(n) \dots f_k(n)|}. \quad (2.18)$$

Definimos, para $n = 0, 1, \dots$

$$g_{0,i}^{(n)} = f_i(n), \quad g_{k,i}^{(n)} = \frac{|f_i(n) f_1(n) \dots f_k(n)|}{|1 f_1(n) \dots f_k(n)|}, \quad i = k+1, k+2, \dots \quad (2.19)$$

Vamos primero a demostrar que $g_{n,i}^{(n)} = f_{k,i}^{(n)}$ con $f_{k,i}^{(n)}$ dado por (2.17), utilizando inducción sobre k . Para $k = 0$, se tiene, por (2.15),

$$g_{0,i}^{(n)} = f_i(n) = f_{0,i}^{(n)}, \quad n = 0, 1, \dots$$

Supongamos que la igualdad es cierta hasta el índice $k-1$. Entonces, para $i > k$

$$\begin{aligned} f_{k,i}^{(n)} &= \frac{f_{k-1,i}^{(n)} f_{k-1,k}^{(n+1)} - f_{k-1,i}^{(n+1)} f_{k-1,k}^{(n)}}{f_{k-1,k}^{(n+1)} - f_{k-1,k}^{(n)}} = \frac{g_{k-1,i}^{(n)} g_{k-1,k}^{(n+1)} - g_{k-1,i}^{(n+1)} g_{k-1,k}^{(n)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}} \\ &= \frac{|f_i(n) f_1(n) - f_{k-1}(n)| |f_k(n+1) f_1(n+1) \dots f_{k-1}(n+1)|}{|f_k(n+1) f_1(n+1) \dots f_{k-1}(n+1)| |1 f_1(n) \dots f_{k-1}(n)| - |f_k(n) f_1(n) \dots f_{k-1}(n+1)| |1 f_1(n+1) \dots f_{k-1}(n+1)|} \\ &= \frac{|f_i(n+1) f_1(n+1) - f_{k-1}(n+1)| |f_k(n) f_1(n) \dots f_{k-1}(n)|}{|f_k(n) f_1(n) \dots f_{k-1}(n)| |1 f_1(n) \dots f_{k-1}(n)| - |f_k(n+1) f_1(n+1) \dots f_{k-1}(n+1)| |1 f_1(n+1) \dots f_{k-1}(n+1)|}. \end{aligned}$$

La expresión anterior no cambia si desplazamos las columnas identificadas por $f_k(n)$ y $f_k(n+1)$ a la última posición en cada determinante en el que aparezcan. De este modo, si

$$C = (f_i(n) f_1(n)) \dots f_k(n), \quad D = (1 f_1(n)) \dots f_k(n),$$

denotan las matrices usadas para definir (2.19), entonces

$$f_{k,i}^{(n)} = \frac{|C_{k+1,k+1}| |C_{1,1}| - |C_{1,k+1}| |C_{k+1,1}|}{|D_{k+1,k+1}| |D_{1,1}| - |D_{1,k+1}| |D_{k+1,1}|} = \frac{|C|}{|D|} \frac{|C_{1,k+1,1,k+1}|}{|D_{1,k+1,1,k+1}|} = \frac{|C|}{|D|} = g_{k,i}^{(n)},$$

2 Extrapolación escalar

donde se ha utilizado la notación de la identidad de Sylvester y que se verifica

$$C_{1,k+1,1,k+1} = D_{1,k+1,1,k+1} = (f_1(n+1) \dots f_{k-1}(n+1)).$$

Por tanto, los $f_{k,i}^{(n)}$, definidos por (2.17), son de la forma (2.19). Teniendo en cuenta esta propiedad y que, por (2.18), $E_k(S_n)$ tiene una expresión similar a la de (2.19), cambiando la columna asociada a f_i por la de S_n , entonces los mismos pasos del razonamiento anterior, esta vez con $C = (S_n f_1(n) \dots f_k(n))$, permiten comprobar que $E_k(S_n) = E_k^{(n)}$. \square

Convergencia del GEA

La convergencia del método es analizada en [32] desde dos puntos de vista. Por un lado, se obtienen condiciones para asegurar, en caso de que exista $S = \lim_{n \rightarrow \infty} S_n$, la convergencia hacia S de las columnas $\{E_k(S_n)\}_{n=0}^{\infty}$ de la tabla 2.1. De otra parte, es necesario explicar de qué forma el E-algoritmo acelera la convergencia de la sucesión original.

En referencia al primer punto, el resultado general es el siguiente:

Teorema 2.1. *Supongamos que existe $\lim_{n \rightarrow \infty} S_n = S$, y dos constantes α, β con $0 < \alpha < 1 < \beta$ tales que*

$$\frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} \notin [\alpha, \beta], \quad (2.20)$$

para todo n a partir de un cierto n_0 . Entonces, para todo $k \geq 0$,

$$\lim_{n \rightarrow \infty} E_k(S_n) = S.$$

Demostración:

Escribimos la recurrencia de tres términos dada por (2.16) en la forma

$$E_k(S_n) = \frac{\frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}}}{\frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} - 1} E_{k-1}(S_n) - \frac{1}{\frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} - 1} E_{k-1}(S_{n+1}). \quad (2.21)$$

Para la demostración, usamos inducción sobre k . Como $E_0^{(n)} = S_n$, se satisface el resultado para $k = 0$ por hipótesis. Supongamos ahora que

$$\lim_{n \rightarrow \infty} E_{k-1}(S_n) = S.$$

A partir de (2.21), tendremos que

$$\lim_{n \rightarrow \infty} E_k(S_n) = S,$$

siempre que $\lim_{n \rightarrow \infty} \frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} \neq 1$. Pero si el límite es 1, entonces a partir de un n

$$1 - \varepsilon < \frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} < 1 + \varepsilon,$$

para $\varepsilon < 1$ arbitrario, lo que contradice (2.20). \square

El teorema 2.1 tiene el inconveniente de imponer una condición sobre los coeficientes (2.17) que no es fácil de verificar. El siguiente resultado traslada esta condición a una hipótesis sobre las funciones f_i que es más accesible.

Teorema 2.2. Si $\lim_{n \rightarrow \infty} S_n = S$ y existen los límites

$$\lim_{n \rightarrow \infty} \frac{f_i(n+1)}{f_i(n)} = b_i, \quad (2.22)$$

con $b_i \neq 1$, para todo i , y $b_i \neq b_j$ para $i \neq j$, entonces, para todo $k \geq 0$,

$$\lim_{n \rightarrow \infty} E_k(S_n) = S.$$

Demostración:

Buscamos aplicar el teorema 2.1; para ello, bastará con comprobar (2.20) sobre los coeficientes (2.17) bajo la condición (2.22). Escribimos

$$\frac{f_{k,i}^{(n+1)}}{f_{k,i}^{(n)}} = \frac{\frac{f_{k-1,k}^{(n+2)}}{f_{k-1,k}^{(n+1)}} - \frac{f_{k-1,i}^{(n+2)}}{f_{k-1,i}^{(n+1)}}}{\frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} - \frac{f_{k-1,i}^{(n+1)}}{f_{k-1,i}^{(n)}}} \cdot \frac{\frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} - 1}{\frac{f_{k-1,k}^{(n+2)}}{f_{k-1,k}^{(n+1)}} - 1} \cdot \frac{f_{k-1,i}^{(n+1)}}{f_{k-1,i}^{(n)}}. \quad (2.23)$$

Por otro lado, dadas las condiciones sobre los b_i , bastará con demostrar por inducción sobre k que

$$\lim_{n \rightarrow \infty} \frac{f_{k,i}^{(n+1)}}{f_{k,i}^{(n)}} = b_i, \quad i > k. \quad (2.24)$$

Como $f_{0,i}^{(n)} = f_i(n)$, se cumple (2.23) para $k = 0$ por la hipótesis (2.22). Suponiendo que se satisface (2.24) en el caso $k - 1$, utilizando (2.23) y las condiciones sobre los b_i , se tiene para $i > k$

$$\lim_{n \rightarrow \infty} \frac{f_{k,i}^{(n+1)}}{f_{k,i}^{(n)}} = \frac{b_k - b_i}{b_k - b_i} \frac{b_k - 1}{b_k - 1} b_i = b_i. \quad \square$$

El teorema 2.2 puede utilizarse para analizar la aceleración de la convergencia. Así, por ejemplo, se tiene:

Teorema 2.3. Bajo la hipótesis del teorema 2.2, supongamos que

$$\lim_{n \rightarrow \infty} \frac{E_{k-1}^{(n+1)} - S}{E_{k-1}^{(n)} - S} = b_k, \quad (2.25)$$

entonces

$$E_k(S_n) - S = o(E_{k-1}(S_n) - S), \quad n \rightarrow \infty. \quad (2.26)$$

Demostración:

Utilizando (2.16) podemos escribir

$$\frac{E_k(S_n) - S}{E_{k-1}(S_n) - S} = \frac{\frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} - \frac{E_{k-1}^{(n+1)} - S}{E_{k-1}^{(n)} - S}}{\frac{f_{k-1,k}^{(n+1)}}{f_{k-1,k}^{(n)}} - 1}$$

y por tanto del teorema 2.2 y (2.25) se obtiene (2.26). \square

El límite (2.25) permite establecer tipos de sucesiones $\{S_n\}_{n=0}^{\infty}$ para las que el algoritmo general acelere su convergencia, a través de las funciones f_i . Un resultado relevante en este sentido, cuya demostración puede verse en [32], es el siguiente:

Teorema 2.4. *Bajo la hipótesis del teorema 2.2, supongamos que:*

1. S_n admite un desarrollo de la forma $S_n \sim S + \sum_{j=1}^{\infty} a_j f_j(n)$, (cf. sección 2.1.1),
2. $f_{i+1}(n) = o(f_i(n))$ para todo i ,

entonces $E_k(n) - S = o(E_{k-1}(S_n) - S)$, para todo $k \geq 0$.

Observemos que la segunda condición del teorema 2.4 establece una limitación sobre las funciones f_i para la aceleración de la convergencia. Ello sugiere un criterio de selección de las mismas, que hasta el momento la teoría no había proporcionado.

2.2.1. La transformación de Shanks

Como se mencionó más arriba, diferentes elecciones de las funciones f_j del algoritmo general de extrapolación dan lugar a métodos conocidos independientemente y cuyo análisis puede ir más allá del limitado por los resultados del GEA. El caso más relevante en este sentido puede ser el de la transformación de Shanks, [53], correspondiente a tomar $f_j(n) = \Delta S_{n+j}$, de manera que

$$e_k(S_n) = \frac{|S_n \Delta S_n \dots \Delta S_{n+k-1}|}{|1 \Delta S_n \dots \Delta S_{n+k-1}|}. \quad (2.27)$$

Varios son los aspectos de la transformación (2.27) en los que se puede profundizar, más allá de que sea un elemento de la fórmula GEA. Notemos primero que el caso $k = 1$ corresponde al método de Aitken. Por otro lado, la caracterización del núcleo de la transformación viene dada por los lemas siguientes, [56].

Lema 2.1. $e_k(S_n) = S$ si y sólo si existen w_0, w_1, \dots, w_k , con $w_k \neq 0$ y tales que

$$\sum_{j=0}^k w_j (S_{n+j} - S) = 0, \quad \sum_{j=0}^k w_j = 1. \quad (2.28)$$

Demostración:

Supongamos que $\{S_n\}_{n=0}^{\infty}$ es exacto para (2.27). Entonces, de la construcción del método, existen $a_0, \dots, a_{k-1}, a_{k-1} \neq 0$ tales que

$$S_n = S + \sum_{j=0}^{k-1} a_j \Delta S_{n+j}, \quad (2.29)$$

es decir

$$S = S_n - \sum_{j=0}^{k-1} a_j (S_{n+j+1} - S_{n+j}) = \sum_{j=0}^k w_j S_{n+j},$$

donde

$$w_0 = 1 + a_0, \quad w_j = a_j - a_{j-1}, \quad 1 \leq j \leq k-1, \quad w_k = -a_{k-1}. \quad (2.30)$$

Entonces los w_j así definidos satisfacen (2.28).

Recíprocamente, si existen w_0, \dots, w_k verificando (2.28), definimos a_0, \dots, a_{k-1} a partir de las relaciones (2.30), de manera que la primera ecuación de (2.28) puede reescribirse en la forma (2.29), es decir $e_k(S_n) = S$. \square

Lema 2.2. $e_k(S_n) = S$ si y sólo si existen $\lambda_1, \dots, \lambda_t$ distintos con $\lambda_j \neq 1$, $j = 1, \dots, t$, tales que $S_n = S + \sum_{j=1}^t P_j(n) \lambda_j^n$, con $P_j(n)$ polinomio en n .

Demostración:

S_n es exacta para (2.27) si y solo si, del lema 2.1, existen w_0, \dots, w_k con $w_k \neq 0$ verificando (2.28), lo que equivale a que la sucesión $e_n = S_n - S$ satisface la ecuación en diferencias homogénea de orden k

$$\sum_{j=0}^k w_j e_{n+j} = 0. \quad (2.31)$$

Sean $\lambda_1, \dots, \lambda_t$ las raíces distintas del polinomio característico asociado

$$Q(z) = (z - \lambda_1)^{m_1} \dots (z - \lambda_t)^{m_t}, \quad m_1 + \dots + m_t = k,$$

donde podemos suponer que $w_k = 1$ y donde m_j denota la multiplicidad algebraica de la raíz característica λ_j , $j = 1, \dots, t$. Observamos que por la segunda ecuación de (2.28), $\lambda_j \neq 1$ para todo $j = 1, \dots, t$. De este modo, la solución general de (2.31) puede escribirse

$$e_n = \sum_{j=1}^t P_j(n) \lambda_j^n,$$

con $P_j(n)$ un polinomio de la forma $\alpha_0 + \alpha_1 n + \dots + \alpha_{m_j-1} n^{m_j-1}$, $j = 1, \dots, t$. \square

Los principales resultados que justifican el uso de la transformación de Shanks como método de extrapolación tienen que ver con su implementación y su convergencia. En el primer caso, la mejora con respecto al algoritmo general está basada en el siguiente teorema demostrado por Wynn, [72].

Teorema 2.5. Si $\varepsilon_{2k}^{(n)} := e_k(S_n)$ y $\varepsilon_{2k+1}^{(n)} := \frac{1}{e_k(\Delta S_n)}$, entonces

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\Delta \varepsilon_k^{(n)}}, \quad k = 1, 2, \dots, \quad n = 0, 1, \dots, \quad (2.32)$$

donde $\Delta \varepsilon_k^{(n)} = \varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}$ y siempre que esta cantidad no se anule.

La fórmula (2.32) permite elaborar una tabla de doble entrada para calcular el elemento $\varepsilon_j^{(m)}$ en la posición (m, j) a partir de la molécula computacional de cuatro términos

$$\begin{array}{ccccccc}
 & & & & k-1 & k & k+1 \\
 & & & & \vdots & \vdots & \vdots \\
 n & \dots & & & \varepsilon_k^{(n)} & \varepsilon_{k+1}^{(n)} & \\
 n+1 & \dots & \varepsilon_{k-1}^{(n+1)} & \varepsilon_k^{(n+1)} & & &
 \end{array} \tag{2.33}$$

y que genera el llamado **algoritmo ε** , cuyos pasos son:

1. $\varepsilon_{-1}^{(n)} = 0, \quad \varepsilon_0^{(n)} = S_n, \quad \varepsilon_1^{(n)} = \frac{1}{\Delta S_n}, \quad n = 0, 1, \dots$
2. Se calcula $\varepsilon_k^{(n)}$ usando (2.32).

Entonces, según el teorema 2.5, los términos del algoritmo ε que proporcionan la transformación de Shanks son los de la forma $\varepsilon_{2k}^{(n)}$.

La fórmula (2.32) de cuatro términos, que genera la tabla ε (2.33), proporciona una implementación eficiente, [70]; ésta se basa en la observación de que, a partir de la primera columna $\varepsilon_0^{(j)} = S_j, j = 0, \dots, m$, (2.32) permite calcular los valores

$$\varepsilon_k^{(\mu-k)}, \quad 0 \leq \mu \leq m, \quad 0 \leq k \leq \mu,$$

moviéndonos por las contradiagonales de la tabla, de manera que los elementos correspondientes a un nuevo término $\varepsilon_0^{(m+1)} = S_{m+1}$ de la primera columna son los de la siguiente contradiagonal, véase la tabla 2.2.

$$\begin{array}{cccccccc}
 \varepsilon_0^{(0)} = S_0 & \rightarrow & \varepsilon_1^{(0)} & \rightarrow & \dots & \rightarrow & \varepsilon_{m-1}^{(0)} & \rightarrow & \varepsilon_m^{(0)} & \rightarrow & \varepsilon_{m+1}^{(0)} \\
 & \nearrow & & \nearrow & \ddots & \nearrow & & \nearrow & & \nearrow & \\
 \varepsilon_0^{(1)} = S_1 & \rightarrow & \varepsilon_1^{(1)} & \rightarrow & \dots & \rightarrow & \varepsilon_{m-1}^{(1)} & \rightarrow & \varepsilon_m^{(1)} & & \\
 & \nearrow & & \nearrow & \ddots & \nearrow & & \nearrow & & & \\
 \varepsilon_0^{(2)} = S_2 & \rightarrow & \varepsilon_1^{(2)} & \rightarrow & \dots & \rightarrow & \varepsilon_{m-1}^{(2)} & & & & \\
 \vdots & & \vdots & & \vdots & & \ddots & & & & \\
 \varepsilon_0^{(m-1)} = S_{m-1} & \rightarrow & \varepsilon_1^{(m-1)} & \rightarrow & \varepsilon_2^{(m-1)} & & & & & & \\
 & & & \nearrow & & & & & & & \\
 \varepsilon_0^{(m)} = S_m & \rightarrow & \varepsilon_1^{(m)} & & & & & & & & \\
 & \nearrow & & & & & & & & & \\
 \varepsilon_0^{(m+1)} = S_{m+1} & & & & & & & & & &
 \end{array}$$

Tabla 2.2: Algoritmo ε .

Utilizando esta propiedad, reescribimos (2.32) en la forma

$$\varepsilon_j^{(n-j)} = \varepsilon_{j-2}^{(n-j+1)} + \frac{1}{\varepsilon_{j-1}^{(n-j+1)} - \varepsilon_{j-1}^{(n-j)}},$$

y la implementamos con un array E y la asignación

$$\varepsilon_j^{(n-j)} \rightarrow E(n-j), \quad n \geq 0, \quad 0 \leq j \leq n.$$

El vector E debe actualizarse a medida que se construyen las contradiagonales, teniendo en cuenta que se deben preservar las dos últimas columnas. La fórmula (2.33) quedaría, en la programación del algoritmo, de la siguiente forma

$$\begin{array}{ccc} & E(n-j) & \rightarrow E(n-j) \\ & \nearrow & \nearrow \\ E(n-j+1) & \rightarrow E(n-j+1) & \end{array}$$

y la tabla 2.2 quedaría

$$\begin{array}{ccccccc} E(0) & E(0) & E(0) & \dots & E(0) & E(0) & E(0) \\ E(1) & E(1) & E(1) & \dots & E(1) & E(1) & \\ E(2) & E(2) & E(2) & \dots & E(2) & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ E(n-2) & E(n-2) & E(n-2) & & & & \\ E(n-1) & E(n-1) & & & & & \\ E(n) & & & & & & \end{array}$$

Esto indica la necesidad de utilizar un valor auxiliar E' para la actualización, con la asignación siguiente:

$$\begin{array}{ccccccc} \varepsilon_j^{(n-j)} & = & \varepsilon_{j-2}^{(n-j+1)} & + & \frac{1}{\varepsilon_{j-1}^{(n-j+1)} - \varepsilon_{j-1}^{(n-j)}} \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ E(n-j) & & E'(n-j+1) & & E(n-j+1) & & E'(n-j) \end{array}$$

Así, dados S_0, \dots, S_{m+2k} , los elementos $\varepsilon_{2k}^{(n)} = e_k(S_n)$, $n = 0, \dots, m$, de la columna $2k+1$ de la tabla ε pueden calcularse a partir del pseudocódigo:

1. Para $n = 0$,

$$E(0) \leftarrow S_0, \quad E'(0) \leftarrow E(0)$$

2. Para $n = 1$,

$$\begin{aligned} E(1) \leftarrow S_1, \quad E(0) &\leftarrow \frac{1}{(E(1) - E'(0))} \\ E'(0) \leftarrow E(0), \quad E'(1) &\leftarrow E(1) \end{aligned}$$

3. Para $l = 0, \dots, m$

Para $n = 2, \dots, 2k+l$

$$E(n) \leftarrow S_n, \quad E(n-1) \leftarrow \frac{1}{(E(n) - E'(n-1))}$$

Para $j = 2, \dots, n-l$

$$E(n-j) \leftarrow E'(n-j+1) + \frac{1}{(E(n-j+1) - E'(n-j))}$$

Para $i = 0, \dots, n-1$

$$E'(i) \leftarrow E(i).$$

Como ya se mencionó anteriormente, un segundo aspecto más desarrollado en la transformación de Shanks, con respecto al algoritmo general, se refiere a su convergencia. Para un análisis exhaustivo, véase [56]. En nuestro caso, vamos a destacar aquellos resultados que nos parecen más representativos de la aplicabilidad del método, con algunas demostraciones que ilustren las técnicas de las que se hace uso.

Convergencia y estabilidad para sucesiones exponenciales

Buena parte de los resultados de convergencia para la transformación de Shanks se refieren a las sucesiones exponenciales, que son aquéllas de la forma

$$S_m \sim S + \sum_{k=1}^{\infty} C_k \lambda_k^m, \quad m \rightarrow \infty, \quad (2.34)$$

con

$$\lambda_k \neq 0, 1, \quad |\lambda_k| \geq |\lambda_{k+1}|, \quad \lim_{k \rightarrow \infty} \lambda_k = 0, \quad (2.35)$$

y C_k constantes. (Recordemos que la expresión anterior significa que para todo N

$$S_m - S = \sum_{k=1}^N C_k \lambda_k^m + O(\lambda_N^m), \quad m \rightarrow \infty,$$

véase sección 2.1.) El principal resultado es el siguiente:

Teorema 2.6. *Supongamos que $\{S_n\}_{n=0}^{\infty}$ es una sucesión del tipo (2.34), (2.35), y que existen n y r , enteros positivos, para los cuales*

$$|\lambda_n| > |\lambda_{n+1}| = |\lambda_{n+2}| = \dots = |\lambda_{n+r}| > |\lambda_{n+r+1}|. \quad (2.36)$$

Entonces

$$\varepsilon_{2n}^{(j)} - S = O(\lambda_{n+1}^j), \quad j \rightarrow \infty, \quad (2.37)$$

que se verifica sea S el límite o el antilímite de la sucesión $\{S_n\}_{n=0}^{\infty}$.

Nótese que, en particular, si $|\lambda_{n+1}| < 1$, entonces

$$\lim_{j \rightarrow \infty} \varepsilon_{2n}^{(j)} = S,$$

obteniéndose así la convergencia de la columna $2n$ (y posteriores) de la tabla ε hacia S , con independencia del comportamiento de la sucesión $\{S_n\}_{n=0}^{\infty}$.

Para la demostración del teorema 2.6 necesitamos una nueva notación y dos lemas auxiliares. Dada una sucesión $\{S_m\}_{m=0}^{\infty}$ introducimos los determinantes de Hankel,

$$H_0^{(n)}(S_m) = 1, \quad H_p^{(n)}(S_m) = \begin{vmatrix} S_n & S_{n+1} & \dots & S_{n+p-1} \\ S_{n+1} & S_{n+2} & \dots & S_{n+p} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n+p-1} & S_{n+p} & \dots & S_{n+2p-2} \end{vmatrix}, \quad p \geq 1, \quad n \geq 0.$$

Así, la transformación de Shanks puede escribirse, después de algunas manipulaciones alge-

braicas, de la forma siguiente:

$$e_k(S_n) = \frac{H_{k+1}^{(n)}(S_m)}{H_k^{(n)}(\Delta^2 S_m)}. \quad (2.38)$$

Lema 2.3. Si $C_m = S_m - S$, entonces

$$\varepsilon_{2k}^{(n)} - S = \frac{H_{k+1}^{(n)}(C_m)}{H_k^{(n)}(\Delta^2 C_m)}.$$

Demostración:

Notemos que

$$\Delta C_m = \Delta(S_m - S) = \Delta S_m.$$

Entonces, esta propiedad y (2.38) implican que

$$\begin{aligned} \varepsilon_{2k}^{(n)} - S &= e_k(S_n) - S = \frac{|S_n \Delta S_n \dots \Delta S_{n+k-1}|}{|1 \Delta S_n \dots \Delta S_{n+k-1}|} - S \frac{|1 \Delta S_n \dots \Delta S_{n+k-1}|}{|1 \Delta S_n \dots \Delta S_{n+k-1}|} = \\ &= \frac{|S_n - S \Delta S_n \dots \Delta S_{n+k-1}|}{|1 \Delta S_n \dots \Delta S_{n+k-1}|} = \frac{|C_n \Delta C_n \dots \Delta C_{n+k-1}|}{|1 \Delta C_n \dots \Delta C_{n+k-1}|} = e_k(C_n) = \frac{H_{k+1}^{(n)}(C_m)}{H_k^{(n)}(\Delta^2 C_m)}. \quad \square \end{aligned}$$

Lema 2.4. Sea $\{S_n\}_{n=0}^\infty$ una sucesión de la forma (2.34), (2.35). Entonces

$$H_n^{(j)}(f_m) \sim \sum_{1 \leq k_1 < \dots < k_n} \left(\prod_{p=1}^n C_{k_p} \lambda_{k_p}^j \right) [V(\lambda_{k_1}, \dots, \lambda_{k_n})]^2,$$

donde $V(\lambda_1, \dots, \lambda_n)$ denota el determinante de Vandermonde de orden n basado en $\lambda_1, \dots, \lambda_n$, esto es

$$V(\lambda_1, \dots, \lambda_n) = \begin{vmatrix} \lambda_1^0 & \lambda_2^0 & \dots & \lambda_n^0 \\ \lambda_1 & \lambda_2 & \dots & \lambda_n \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \dots & \lambda_n^{n-1} \end{vmatrix} = \prod_{1 \leq i < j \leq n} (\lambda_j - \lambda_i).$$

Demostración: Véase [56].

Demostración del teorema 2.6:

A partir del lema 2.3, se tiene que

$$\varepsilon_{2n}^{(j)} - S = \frac{H_{n+1}^{(j)}(C_m)}{H_n^{(j)}(\Delta^2 C_m)}, \quad C_m = S_m - S \approx \sum_{k=1}^{\infty} C_k \lambda_k^m, \quad m \rightarrow \infty.$$

Por otro lado, $\{C_n\}_{n=0}^\infty$ es también exponencial, por lo que aplicando el lema 2.4 al numerador, se tiene

$$H_{n+1}^{(j)}(C_m) \sim \sum_{1 \leq k_1 < \dots < k_{n+1}} \left(\prod_{p=1}^{n+1} C_{k_p} \lambda_{k_p}^j \right) [V(\lambda_{k_1}, \dots, \lambda_{k_{n+1}})]^2, \quad j \rightarrow \infty.$$

Asímismo, de (2.36)

$$\Delta^2 C_m = \sum_{k=1}^{\infty} \beta_k \lambda_k^m, \quad m \rightarrow \infty,$$

2 Extrapolación escalar

con $\beta_k = C_k(\lambda_k - 1)^2$. De este modo, $\{\Delta^2 C_n\}_{n=0}^\infty$ es también exponencial, de manera que aplicando el lema 2.4 para $f_m = \Delta^2 C_m$ y un razonamiento similar al utilizado para el numerador, tenemos un desarrollo para el denominador de la forma

$$H_n^{(j)}(\Delta^2 C_m) \leq \sum_{1 \leq k_1 < \dots < k_n} \left(\prod_{p=1}^n \beta_{k_p} \lambda_{k_p}^j \right) [V(\lambda_{k_1} \dots \lambda_{k_n})]^2 \sim \left(\prod_{p=1}^n \beta_p \lambda_p^j \right) [V(\lambda_1 \dots \lambda_n)]^2.$$

Si dividimos las dos expresiones asintóticas, el coeficiente que multiplica a $C_p \lambda_p^j$, $p = n + 1, \dots, n + r$, es

$$\frac{\prod_{p=1}^n C_p \lambda_p^j [V(\lambda_1 \dots \lambda_n, \lambda_p)]^2}{\prod_{p=1}^n \beta_p \lambda_p^j [V(\lambda_1 \dots \lambda_n)]^2} = \prod_{k=1}^n \frac{(\lambda_p - \lambda_k)^2}{(1 - \lambda_k)^2},$$

obteniéndose la expresión

$$\varepsilon_{2n}^{(j)} - S \sim \sum_{p=n+1}^{n+r} C_p \lambda_p^j \prod_{k=1}^n \frac{(\lambda_p - \lambda_k)^2}{(\lambda_k - 1)^2} = \mathcal{O}(\lambda_{n+1}^j). \quad \square$$

Una sencilla ilustración de la interpretación del teorema 2.6 como aceleración de convergencia viene dada por la aplicación de la transformación de Shanks a sucesiones generadas a partir de una iteración de punto fijo

$$S_{m+1} = g(S_m), \quad m \geq 0, \quad (2.39)$$

para g suficientemente regular en un entorno del punto fijo $S = g(S)$. En estas condiciones, puede demostrarse la existencia de un desarrollo de la forma

$$S_m \sim S + \sum_{k=1}^{\infty} \alpha_k \lambda^{km}, \quad m \rightarrow \infty,$$

con $\lambda = g'(S)$, α_k dependientes de g y S_0 , $\alpha_1 \neq 0$, [23]. Entonces $\{S_m\}_{m=0}^\infty$ es una sucesión de la forma (2.34)-(2.35) con $\lambda_k = \lambda^k$. El teorema 2.6 afirma que la transformación de Shanks acelera la convergencia de la iteración, en el sentido de que, según (2.37)

$$\varepsilon_{2n}^{(j)} - S = \mathcal{O}(\lambda^{j(n+1)}).$$

Así, por ejemplo, si aplicamos el método de Aitken (correspondiente a la transformación de Shanks con $n = 1$) a (2.39), se tiene que $e_j = S_j - S = \mathcal{O}(\lambda^j)$, mientras que $E_j = \varepsilon_2^{(j)} - S = \mathcal{O}(\lambda^{2j})$. Como consecuencia,

$$\frac{e_{j+1}}{e_j} = \mathcal{O}(\lambda), \quad \frac{E_{j+1}}{E_j} = \mathcal{O}(\lambda^2). \quad (2.40)$$

Esto puede observarse en las últimas columnas de las tablas 1.1 y 1.2 correspondientes al ejemplo de la sección 1.2.1. Nótese que (2.40) permite también estimar $\lambda = g'(S)$.

A la convergencia de la transformación de Shanks para sucesiones exponenciales, dada por el teorema 2.6, debe añadirse un análisis de la estabilidad del método para esta clase de procesos secuenciales. A partir de la representación alternativa

$$\varepsilon_n^{(j)} = \sum_{i=0}^n \gamma_{n,i}^{(j)} S_{j+i},$$

con

$$\sum_{i=0}^n \gamma_{n,i}^{(j)} z^i = \frac{R_n^{(j)}(z)}{R_n^{(j)}(1)} = \frac{\begin{vmatrix} 1 & z & \dots & z^n \\ \Delta S_j & \Delta S_{j+1} & \dots & \Delta S_{j+n} \\ \vdots & \vdots & \vdots & \vdots \\ \Delta S_{j+n-1} & \Delta S_{j+n} & \dots & \Delta S_{j+2n-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \dots & 1 \\ \Delta S_j & \Delta S_{j+1} & \dots & \Delta S_{j+n} \\ \vdots & \vdots & \vdots & \vdots \\ \Delta S_{j+n-1} & \Delta S_{j+n} & \dots & \Delta S_{j+2n-1} \end{vmatrix}}$$

se tiene, [56],

Teorema 2.7. *Bajo las condiciones del teorema 2.6*

$$\lim_{j \rightarrow \infty} \sum_{i=0}^n \gamma_{n,i}^{(j)} z^i = \prod_{i=1}^n \frac{z - \lambda_i}{1 - \lambda_i} = \sum_{i=0}^n \rho_{ni} z^i,$$

esto es $\rho_{ni} = \lim_{j \rightarrow \infty} \gamma_{n,i}^{(j)}$, $i = 0, \dots, n$. Por tanto, la transformación de Shanks es estable con respecto a las sucesiones columna de la tabla ε , con

$$\lim_{j \rightarrow \infty} \Gamma_n^{(j)} = \sum_{i=0}^n |\rho_{ni}| \leq \prod_{i=1}^n \frac{1 + |\lambda_i|}{|1 - \lambda_i|},$$

siendo $\Gamma_n^{(j)} = \sum_{i=0}^n |\gamma_{n,i}^{(j)}|$.

Del teorema 2.7 se deduce que la estabilidad puede deteriorarse cuando los λ_i están cercanos a uno.

Convergencia y estabilidad para otras sucesiones

Resumimos aquí el comportamiento de la transformación de Shanks con respecto a otros tipos de sucesiones. Los resultados pueden consultarse en varias referencias, por ejemplo [56, 70].

Teorema 2.8. *(Sucesiones lineales.) Si $\{S_n\}_{n=0}^{\infty}$ es una sucesión del tipo (2.2), con $\sigma \neq 0, \dots, n-1$, entonces*

$$\varepsilon_{2n}^{(j)} - S \sim (-1)^n \alpha_0 \frac{n! [\sigma]_n}{(\lambda - 1)^{2n}} \lambda^{j+2n} j^{\sigma-2n},$$

donde $[\sigma]_n = \sigma(\sigma-1)\dots(\sigma-n+1)$, $n > 1$, $[\sigma]_0 = 1$. Si σ es entero, $0 \leq \sigma \leq n-1$ y $\alpha_{\sigma+1} \neq 0$, entonces

$$\varepsilon_{2n}^{(j)} - S \sim \alpha_{\sigma+1} \frac{(n-\sigma-1)!(n+\sigma+1)!}{(\lambda-1)^{2n}} \lambda^{j+2n} j^{-2n-1}, \quad j \rightarrow \infty.$$

Además,

$$\lim_{j \rightarrow \infty} \sum_i \gamma_{n,i}^{(j)} z^i = \left(\frac{z - \lambda}{1 - \lambda} \right)^n,$$

y la transformación de Shanks es estable con respecto a las sucesiones columna de la tabla ε , verificándose que

$$\lim_{j \rightarrow \infty} \Gamma_n^{(j)} = \left(\frac{1 + |\lambda|}{|1 - \lambda|} \right)^n.$$

2 Extrapolación escalar

A la vista del teorema 2.8, la convergencia por columnas de la transformación de Shanks para sucesiones lineales depende de λ , de manera que:

1. Si $|\lambda| < 1$, entonces $\{\varepsilon_{2n}^{(j)}\}_{j=0}^{\infty}$ converge hacia S . La aceleración es mayor a medida que se avanza en las columnas.
2. Si $|\lambda| = 1$, $\lambda \neq 1$, entonces $\{\varepsilon_{2n}^{(j)}\}_{j=0}^{\infty}$ converge hacia S para $n > \frac{1}{2}$ cuando $\sigma \neq 0, 1, \dots, n-1$ o bien, si σ es un entero no negativo, para $n \geq \sigma + 1$.
3. En el resto de los casos, $\{\varepsilon_{2n}^{(j)}\}_{j=0}^{\infty}$ diverge.

Teorema 2.9. (Sucesiones logarítmicas.) Si $\{S_n\}_{n=0}^{\infty}$ es de la forma (2.3), entonces

$$\varepsilon_{2n}^{(j)} - S \sim (-1)^n \alpha_0 \frac{n!}{[\sigma-1]_n} j^\sigma, \quad j \rightarrow \infty,$$

lo que implica que la transformación de Shanks no acelera la convergencia de este tipo de sucesiones.

Teorema 2.10. (Sucesiones factoriales.) Si $\{S_n\}_{n=0}^{\infty}$ es de la forma (2.4), entonces

$$\varepsilon_{2n}^{(j)} - S \sim (-1)^n \alpha_0 r^n \frac{\lambda^{j+2n}}{(j!)^r} j^{\sigma-2nr-n}, \quad j \rightarrow \infty,$$

$$\lim_{j \rightarrow \infty} \sum_{i=0}^n \gamma_{n,i}^{(j)} z^i = z^n,$$

(y, por tanto, $\lim_{j \rightarrow \infty} \Gamma_n^{(j)} = 1$). De este modo

$$\frac{\varepsilon_{2n}^{(j)} - S}{S_{j+2n} - S} \sim C j^{-n}, \quad j \rightarrow \infty,$$

con C constante. Así, las columnas de la tabla ε aceleran la convergencia de $\{S_n\}_{n=0}^{\infty}$, con cada columna convergiendo más rápidamente que la anterior.

2.3. Métodos polinómicos y racionales

Las diferentes formulaciones presentadas en la sección 2.1 justifican el hecho de que, con frecuencia, se hable indistintamente de transformación para acelerar la convergencia y de extrapolación. La familia de algoritmos de la sección 2.2 hace un uso más explícito de la idea de los algoritmos como transformaciones entre sucesiones. El concepto de extrapolación se hace más evidente en los esquemas que presentamos en esta sección.

El planteamiento inicial asume la existencia de una función $s(x)$ que coincide con la sucesión $\{S_n\}_{n=0}^{\infty}$ a transformar al evaluarse en un conjunto discreto de argumentos $\{x_n\}_{n=0}^{\infty}$,

$$s(x_n) = S_n.$$

Esto reduce el problema de aceleración de convergencia a un problema de extrapolación: conocidos $S_m, S_{m+1}, \dots, S_{m+k}$, se busca construir una aproximación $s_k(x)$ a $s(x)$ verificando

las $k + 1$ condiciones de interpolación

$$s_k(x_{m+j}) = S_{m+j}, \quad 0 \leq j \leq k. \quad (2.41)$$

Requerimientos mínimos de regularidad aseguran que si $S = \lim_{n \rightarrow \infty} S_n$, entonces $S = \lim_{m \rightarrow \infty} S_m = \lim_{m \rightarrow \infty} s(x_m) = s(\lim_{m \rightarrow \infty} x_m) = s(x_\infty)$, con $x_\infty = \lim_{m \rightarrow \infty} x_m$.

El siguiente paso es entonces determinar el valor del aproximante $s_k(x)$ cuando $x \rightarrow x_\infty$, con la esperanza de que sea una aproximación a S más precisa que el último término S_{m+k} de la sucesión utilizada para la construcción de $s_k(x)$.

Puesto que las funciones interpolantes más comunes son los polinomios y las funciones racionales (son las más sencillas, flexibles tanto teóricamente como en aspectos de computación y dan buenos resultados para una clase amplia de sucesiones a transformar), la literatura suele distinguir dos grandes familias de métodos de extrapolación:

1. Polinómica. Esto es, considerando que $s_k(x)$ es un polinomio de grado k ,

$$s_k(x) = c_0 + c_1x + \cdots + c_kx^k.$$

En este caso, los puntos de interpolación suelen tomarse verificando las condiciones, [70]

$$x_m > x_{m+1}, \quad x_m > 0, \quad m \geq 0; \quad x_\infty = \lim_{m \rightarrow \infty} x_m = 0. \quad (2.42)$$

De este modo, la aproximación al límite S se identifica con el término independiente $c_0 = s_k(0)$.

Analizaremos el método de extrapolación de Richardson polinómica (PREP) como ejemplo de esta formulación.

2. Racional. En este caso, se asume que s_k es una función racional,

$$s_k(x) = \frac{a_0 + a_1x + \cdots + a_lx^l}{b_0 + b_1x + \cdots + b_mx^m},$$

para enteros positivos $l > m$. La función s_k contiene así $l + m + 2$ coeficientes, de los que $l + m + 1$ son independientes, al determinarse s_k salvo un factor común no nulo. (Esto suele plasmarse fijando $b_0 = 1$ o bien $b_m = 1$.) De este modo, para determinar los coeficientes de s_k necesitamos $k + 1$ condiciones de interpolación (2.41), con $k = l + m$. Aquí, podemos aplicar el procedimiento con dos posibles situaciones. En la primera, el punto de interpolación es $x_\infty = 0$, asumiendo (2.42) y de manera que la aproximación al límite se identifica con el cálculo de $s_k(0) = a_0/b_0$. En la segunda situación se asume $l = m$, adaptándose el procedimiento a problemas en los que $x_\infty = +\infty$, de forma que la sucesión $\{x_n\}_{n=0}^\infty$ satisface, en lugar de (2.42),

$$0 < x_m, \quad x_m < x_{m+1}, \quad m \geq 0; \quad \lim_{m \rightarrow \infty} x_m = +\infty, \quad (2.43)$$

y la aproximación al límite se identifica con $\lim_{x \rightarrow +\infty} s_k(x) = a_l/b_l$.

El ejemplo que utilizaremos para ilustrar esta familia es el llamado ρ -algoritmo, [73, 13].

2.3.1. Método de extrapolación de Richardson polinómica (PREP)

Para la construcción del método, suponemos que, cuando $h \rightarrow 0$, la función $s(h)$ admite un desarrollo (cf. ejemplo 1.2.2)

$$s(h) = S + \sum_{j=1}^k \alpha_j h^{\sigma_j} + O(h^{\sigma_{k+1}}),$$

donde $\sigma_k \neq 0$ son tales que $h^{\sigma_i} > h^{\sigma_{i+1}}$, $h \downarrow 0$ y $\alpha_j \in \mathbb{R}$, $j = 1, \dots, k$. La extrapolación de Richardson polinómica corresponde a tomar exponentes de la forma

$$\sigma_j = jr, \quad j \geq 1, \quad (2.44)$$

para cierta constante $r > 0$. Asumiendo la existencia de una sucesión $h_m \downarrow 0 = h_\infty$, con $h_m > 0$, $m \geq 0$ y denotando $S_m = s(h_m)$, se define la aproximación $S_k^{(n)}$ a través de las ecuaciones

$$\begin{cases} S_n = S_k^{(n)} + \sum_{j=1}^k \alpha_j h_n^{jr}, \\ \vdots \\ S_{n+k} = S_k^{(n)} + \sum_{j=1}^k \alpha_j h_{n+k}^{jr}. \end{cases} \quad (2.45)$$

La resolución de (2.45) puede abordarse desde dos puntos de vista:

1. Con el cambio de variable $x = h^r$, $s(x) = s(h)$, entonces (2.45) tiene la forma de un problema de interpolación

$$s(x_{n+l}) = S_k^{(n)} + \sum_{j=1}^k \alpha_j x_{n+l}^j, \quad 0 \leq l \leq k, \quad x_{n+l} = h_{n+l}^r,$$

de manera que $S_k^{(n)} = s_{k,n}(0)$, donde $s_{k,n}(x)$ es el polinomio de grado menor o igual que k que interpola $s(x)$ en los nodos x_{n+l} , $0 \leq l \leq k$. Estos polinomios pueden calcularse de forma recursiva con el algoritmo de interpolación de Neville-Aitken, [45, 2, 48],

$$\begin{aligned} s_{0,n}(x) &= S_n, \quad n = 0, 1, \dots, \\ s_{k,n}(x) &= \frac{(x - x_{n+k})s_{k-1,n}(x) - (x - x_n)s_{k-1,n+1}(x)}{x_n - x_{n+k}}. \end{aligned} \quad (2.46)$$

La evaluación de (2.46) en el punto límite $x = 0$ se conoce como algoritmo de Burlisch y Stoer, [20],

$$\begin{aligned} S_0^{(n)} &= S_n, \quad n = 0, 1, \dots, \\ S_k^{(n)} &= \frac{x_n S_{k-1}^{(n+1)} - x_{n+k} S_{k-1}^{(n)}}{x_n - x_{n+k}}, \quad n = 0, 1, \dots, \quad k = 1, 2, \dots \end{aligned} \quad (2.47)$$

El algoritmo (2.47) permite construir la tabla de Romberg para los $S_k^{(n)}$, con k como índice para las columnas y n para las filas. Alternativamente, algunos resultados de

convergencia y la implementación suelen hacer uso de una disposición de $S_k^{(n)}$ en la fila $n+k$ y la columna k . De este modo, la molécula computacional de tres términos puede construir la tabla de dos formas:

$$\begin{array}{ccc} S_{k-1}^{(n)} & \rightarrow & S_k^{(n)} \\ & \nearrow & \\ S_{k-1}^{(n+1)} & & \end{array}, \quad \begin{array}{ccc} S_{k-1}^{(n)} & & \\ & \searrow & \\ S_{k-1}^{(n+1)} & \rightarrow & S_k^{(n)} \end{array}.$$

2. Asimismo, el método puede presentarse como un caso particular dentro de la familia del algoritmo general explicado en la sección 2.2.1, utilizando las funciones $f_j(n) = h_n^{r_j} = x_n^j$, $j = 1, \dots, k$. De este modo, podemos escribir

$$S_m^{(k)} = \left| \begin{array}{cccc|cccc} S_m & S_{m+1} & \dots & S_{m+k} & 1 & 1 & \dots & 1 \\ x_m & x_{m+1} & \dots & x_{m+k} & x_m & x_{m+1} & \dots & x_{m+k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_m^k & x_{m+1}^k & \dots & x_{m+k}^k & x_m^k & x_{m+1}^k & \dots & x_{m+k}^k \end{array} \right|,$$

y, por construcción, la transformación es exacta para sucesiones de la forma

$$S_m = S + \sum_{j=1}^k \alpha_j x_m^j, \quad m, k \in \mathbb{N}.$$

Pasamos a analizar la convergencia y la estabilidad del método PREP. Tal y como se adelantó en la sección 1.2.2 y en referencia a la tabla de Romberg, podemos tener convergencia de dos tipos, por columnas y por diagonales, [56]. Para estudiar la convergencia por columnas, hacemos uso de la fórmula para el error de interpolación

$$s(x) - s_{n,k}(x) = s[x, x_n, \dots, x_{n+k}] \prod_{i=0}^k (x - x_{n+i}),$$

donde $s[x_0, \dots, x_l]$ denota la diferencia dividida de orden l de s en los nodos x_0, \dots, x_l . Haciendo $x = 0$, tenemos:

$$S - S_k^{(n)} = (-1)^k s[0, x_n, \dots, x_{n+k}] \prod_{i=0}^k x_{n+i}. \quad (2.48)$$

Si s es de clase C^k en un intervalo que contiene a los puntos $0, x_n, \dots, x_{n+k}$ y existe la derivada de orden $k+1$ de s , $s^{(k+1)}$, en dicho intervalo, entonces existe $x_{n,k}$ con $0 < x_{n,k} < \max(x_n, \dots, x_{n+k})$ y tal que

$$s[0, x_n, \dots, x_{n+k}] = \frac{s^{(k+1)}(x_{n,k})}{(k+1)!}.$$

Podemos entonces escribir (2.48) como

$$S - S_k^{(n)} = (-1)^k \frac{s^{(k+1)}(x_{n,k})}{(k+1)!} \prod_{i=0}^k x_{n+i}$$

2 Extrapolación escalar

de esta forma,

$$S - S_k^{(n)} = \mathcal{O}(x_n x_{n+1} \cdots x_{n+k}). \quad (2.49)$$

Como $x_m = h_m^r \downarrow 0$, si $m \rightarrow \infty$, la expresión (2.49) proporciona la convergencia por columnas de la tabla de Romberg, es decir, de las sucesiones $\{S_k^{(n)}\}_{n=0}^\infty$. De (2.49) y el decrecimiento hacia cero de la sucesión $\{x_n\}_{n=0}^\infty$ se deduce también la aceleración de la convergencia al pasar de una columna k a la siguiente. Por otro lado, en algunos casos se puede dar una argumentación alternativa para obtener (2.49) requiriendo menos regularidad a la función s , cf. [56].

En relación a la convergencia por diagonales, analizamos el caso, frecuente en las aplicaciones, en que la sucesión $\{h_n\}_{n=0}^\infty$ es de la forma

$$h_m = h_0 w^m, \quad h_0 > 0, \quad w \in (0, 1), \quad m \geq 0. \quad (2.50)$$

Definimos $c_j = w^{jr}$, $j = 1, 2, \dots$. Entonces se tiene el siguiente

Teorema 2.11. *Se consideran los polinomios*

$$U_n(z) = \prod_{j=1}^n \frac{z - c_j}{1 - c_j} = \sum_{i=0}^n \rho_{n,i} z^i, \quad n \geq 1, \quad U_0(z) = 1. \quad (2.51)$$

Entonces

$$S_k^{(n)} = \sum_{i=0}^k \rho_{k,i} s(h_{n+i}). \quad (2.52)$$

Demostración:

De (2.51) se tiene la recurrencia

$$U_k(z) = \frac{z - c_k}{1 - c_k} U_{k-1}(z), \quad k \geq 1.$$

Entonces, igualando coeficientes se deduce que

$$\rho_{k,i} = \frac{\rho_{k-1,i-1} - c_k \rho_{k-1,i}}{1 - c_k}, \quad 0 \leq i \leq k,$$

donde se supone $\rho_{j,i} = 0$, si $i < 0$ ó $i > j$. De este modo, si $R_k^{(n)}$ denota el lado derecho de (2.52), podemos escribir

$$\begin{aligned} R_k^{(n)} &= \sum_{i=0}^k \rho_{k,i} s(h_{n+i}) = \sum_{i=0}^k \frac{\rho_{k-1,i-1} - c_k \rho_{k-1,i}}{1 - c_k} s(h_{n+i}) \\ &= \frac{1}{1 - c_k} \left(\sum_{i=0}^k \rho_{k-1,i-1} s(h_{n+i}) - c_k \sum_{i=0}^k \rho_{k-1,i} s(h_{n+i}) \right) \\ &= \frac{1}{1 - c_k} \left(\sum_{j=0}^{k-1} \rho_{k-1,j} s(h_{n+j+1}) - c_k \sum_{j=0}^{k-1} \rho_{k-1,j} s(h_{n+j}) \right) \\ &= \frac{1}{1 - c_k} \left(R_{k-1}^{(n+1)} - c_k R_{k-1}^{(n)} \right), \end{aligned}$$

de manera que $R_k^{(n)}$ satisface la recurrencia de tres términos (2.47) con $x_n = h_n^r = h_0^r w^{nr} = h_0^r c_n$

y, por tanto, coincide con $S_k^{(n)}$, obteniéndose (2.52). □

Teorema 2.12. En las condiciones del teorema 2.11, los coeficientes $\rho_{k,i}$ satisfacen

$$\sum_{i=0}^k |\rho_{n,i}| |z|^i \leq \prod_{i=1}^k \frac{|z| + |c_i|}{|1 - c_i|}. \quad (2.53)$$

En particular

$$\sum_{i=0}^k |\rho_{k,i}| \leq \prod_{i=1}^k \frac{1 + |c_i|}{|1 - c_i|}. \quad (2.54)$$

Demostración:

Sea $Q(z) = \prod_{i=1}^k \frac{z + |c_i|}{|1 - c_i|} = \sum_{i=0}^k a_{k,i} z^i$, donde, operando, identificamos

$$a_{k,i} = \frac{1}{(1 - c_1) \cdots (1 - c_k)} \sum_{k_1 \leq k_2 \leq \cdots \leq k_i} \prod_{j=1}^i |c_{k_j}|.$$

Por otro lado, si $U_k(z)$ es el polinomio definido en (2.51), cálculos similares nos llevan a

$$\rho_{k,i} = \frac{1}{(1 - c_1)(1 - c_2) \cdots (1 - c_n)} (-1)^{n-i} \sum_{k_1 \leq k_2 \leq \cdots \leq k_i} \prod_{j=1}^i c_{k_j}.$$

Por tanto, $|\rho_{k,i}| \leq a_{k,i}$, $1 \leq i \leq k$, y

$$\sum_{i=0}^k |\rho_{n,i}| |z|^i \leq \sum_{i=0}^k a_{k,i} |z|^i = Q(|z|) = \prod_{i=1}^k \frac{|z| + |c_i|}{|1 - c_i|},$$

lo que demuestra (2.53). La desigualdad (2.54) se obtiene evaluando (2.53) en $z = 1$. □

Teorema 2.13. Sea $k_0 \in \mathbb{R}$ tal que

$$s(h) = S + \sum_{j=1}^{k_0} \alpha_j h^{j r} + O(h^{r(k_0+1)}), \quad h \rightarrow 0, \quad (2.55)$$

y sea

$$\alpha_{k_0+1} = \max_{0 \leq h \leq h_0} \left| \frac{1}{h^{r(k_0+1)}} \left(s(h) - S - \sum_{j=1}^{k_0} \alpha_j h^{j r} \right) \right|. \quad (2.56)$$

Entonces, para cada n y cada $k \geq k_0$

$$|S_k^{(n)} - S| \leq \alpha_{k_0+1} h_n^{r(k_0+1)} \prod_{i=1}^k \frac{|c_{k_0+1}| + |c_i|}{|1 - c_i|}. \quad (2.57)$$

Demostración:

Sea

$$R_{k_0}(h) = s(h) - S - \sum_{j=1}^{k_0} \alpha_j h^{j r}.$$

2 Extrapolación escalar

Entonces, de la definición de (2.56), es

$$|R_{k_0}(h)| \leq \alpha_{k_0+1} h^{r(k_0+1)}, \quad 0 < h \leq h_0. \quad (2.58)$$

Sustituyendo (2.55) en (2.52) para $k \geq k_0$, se tiene

$$\begin{aligned} S_k^{(n)} &= \sum_{i=0}^k \rho_{k,i} S(h_{n+i}) = \sum_{i=0}^k \rho_{k,i} \left(S + \sum_{j=1}^{k_0} \alpha_j h_{n+i}^{j_r} + R_{k_0}(h_{n+i}) \right) \\ &= S + \sum_{j=1}^{k_0} \alpha_j \sum_{i=0}^k \rho_{k,i} h_0^{j_r} w^{j_r(n+i)} + \sum_{i=0}^k \rho_{k,i} R_{k_0}(h_{n+i}) \\ &= S + \sum_{j=1}^{k_0} \alpha_j h_0^{j_r} w^{j_r n} U_k(c_j) + \sum_{i=0}^k \rho_{k,i} R_{k_0}(h_{n+i}), \end{aligned}$$

donde $U_k(z)$ está definido en (2.51) y en particular $\sum_{i=0}^k \rho_{k,i} = U_k(1) = 1$. Observamos que, como $k \geq k_0$, $U_k(c_j) = 0$, $1 \leq j \leq k_0$ y por tanto,

$$S_k^{(n)} = S + \sum_{i=0}^k \rho_{k,i} R_{k_0}(h_{n+i}).$$

A partir de (2.58) y la definición de los c_j , se tiene entonces

$$|S_k^{(n)} - S| \leq \sum_{i=0}^k |\rho_{k,i}| |R_{k_0}(h_{n+i})| \leq \alpha_{k_0+1} \sum_{i=0}^k |\rho_{k,i}| h_{n+i}^{r(k_0+1)} \leq \alpha_{k_0+1} h_n^{r(k_0+1)} \sum_{i=0}^k |\rho_{k,i}| |c_{k_0+1}|^i,$$

de donde obtenemos (2.57) por el teorema 2.12. □

Teorema 2.14. En las condiciones del teorema 2.13,

$$S_k^{(n)} - S = \mathcal{O}(w^{rk(k_0+1)}), \quad k \rightarrow \infty. \quad (2.59)$$

Si, además, el desarrollo de (2.55) es válido para todo k_0 entonces

$$S_k^{(n)} - S = \mathcal{O}(w^{\alpha k}), \quad k \rightarrow \infty,$$

para todo $\alpha > 0$.

Demostración:

Reescribimos (2.57) en la forma

$$|S_k^{(n)} - S| \leq \alpha_{k_0+1} h_n^{r(k_0+1)} |c_{k_0+1}|^k \prod_{i=1}^k \frac{1 + \left| \frac{c_i}{c_{k_0+1}} \right|}{|1 - c_i|}, \quad (2.60)$$

con $k \geq k_0$. Como $c_i = w^{ri}$, entonces $\frac{|c_{i+1}|}{|c_i|} = w^r < 1$, por lo que la serie $\sum_{i=1}^{\infty} c_i$ converge absolutamente, lo que implica que los productos infinitos $\prod_{i=1}^{\infty} |1 \pm c_i|$ convergen, de modo que $\inf_k \prod_{i=1}^k |1 - c_i| > 0$.

Por otro lado, para $i \geq k_0 + 1$, se tiene que $\left| \frac{c_i}{c_{k_0+1}} \right| = w^{(i-k_0-1)r}$, luego si $k \geq k_0$, entonces

$$\prod_{i=1}^k \left(1 + \frac{|c_i|}{|c_{k_0+1}|} \right) = \prod_{i=1}^{k_0} \left(1 + \frac{|c_i|}{|c_{k_0+1}|} \right) \prod_{i=k_0+1}^k (1 + w^{r(i-k_0-1)}) = \prod_{i=1}^{k_0} \left(1 + \frac{|c_i|}{|c_{k_0+1}|} \right) \prod_{i=1}^{k-(k_0+1)} (1 + w^{ri}) < \infty.$$

De la última expresión y (2.60), se deduce (2.59). \square

Notemos que el último teorema proporciona la convergencia por diagonales para el caso de sucesiones de la forma (2.50) y teniendo en cuenta, tal y como se mencionó anteriormente, que en esta disposición $S_k^{(n)}$ se encuentra en la posición $(n+k, k)$ de la tabla. En referencia a la estabilidad, tenemos que, de la representación del método formulado dentro del algoritmo general y usando (2.55),

$$\Gamma_k^{(n)} = \sum_{i=0}^k |\rho_{k,i}| \leq \prod_{i=1}^k \frac{1+|c_i|}{|1-c_i|}, \quad (2.61)$$

donde la cota no depende de n . Obtenemos entonces el siguiente resultado, [56]:

Teorema 2.15. *En las condiciones del teorema 2.13:*

a) *El proceso de generar $\{S_k^{(n)}\}_{n=0}^{\infty}$ es estable en el sentido de que*

$$\sup_n \Gamma_k^{(n)} = \sum_{i=0}^k |\rho_{k,i}| < \infty.$$

b) *El proceso de generar $\{S_k^{(n)}\}_{k=0}^{\infty}$ es estable en el sentido de que*

$$\sup_k \Gamma_k^{(n)} < \infty. \quad (2.62)$$

Demostración:

Para el apartado a) se utiliza (2.61). En el caso b), observamos que la convergencia de los productos infinitos $\prod_{i=1}^{\infty} (1 \pm c_i)$, demostrada en el teorema anterior, y de nuevo (2.61) implican que

$$\sup_k \Gamma_k^{(n)} \leq \sup_k \prod_{i=0}^k \frac{1+|c_i|}{|1-c_i|} < \infty,$$

obteniéndose (2.62). \square

2.3.2. Extrapolación racional. Algoritmo ρ

La ilustración de la extrapolación racional viene aquí dada por el algoritmo ρ . Para su presentación, pueden combinarse los dos puntos de vista utilizados en el método PREP: un problema de interpolación, en este caso basado en nodos x_m verificando (2.43), y la formulación a través del algoritmo general de la sección 2.2.1.

Sea $k \geq 0$. Para una función racional de la forma

$$s_{2k}(x) = \frac{a_0 + a_1x + \cdots + a_kx^k}{b_0 + b_1x + \cdots + b_{k-1}x^{k-1} + x^k}, \quad (2.63)$$

2 Extrapolación escalar

imponemos las $2k + 1$ condiciones de interpolación

$$S_{n+j} = s_{2k}(x_{n+j}), \quad 0 \leq j \leq 2k. \quad (2.64)$$

Reescribiendo (2.63) como

$$s_{2k}(x) = a_k + \left(\frac{a_{k-1}}{x} - \frac{b_{k-1}}{x} s_{2k}(x) \right) + \left(\frac{a_{k-2}}{x^2} - \frac{b_{k-2}}{x^2} s_{2k}(x) \right) + \cdots + \left(\frac{a_0}{x^k} - \frac{b_0}{x^k} s_{2k}(x) \right), \quad (2.65)$$

la imposición de (2.64) sobre (2.65) lleva a una formulación del tipo del algoritmo general

$$S_m = \rho_{2k}^{(n)} + \sum_{j=1}^k (c_{2j-1} f_{2j-1}(m) + c_{2j} f_{2j}(m)), \quad m = n, \dots, n + 2k, \quad (2.66)$$

donde $\rho_{2k}^{(n)} = a_k = \lim_{x \rightarrow +\infty} s_{2k}(x)$ y

$$f_{2j-1}(n) = S_n/x_n^j, \quad c_{2j-1} = -b_{k-j}, \quad f_{2j}(n) = 1/x_n^j, \quad c_{2j} = a_{k-j}, \quad 1 \leq j \leq k.$$

De este modo, $\rho_{2k}^{(n)}$ puede calcularse teóricamente a partir de las fórmulas de Cramer para el sistema (2.66), si bien, como en el caso de la transformación de Shanks, es más realista construir un algoritmo tipo ε utilizando valores auxiliares $\rho_{2k+1}^{(n)}$. El resultado es el llamado **algoritmo ρ** y tiene la forma

$$\rho_{-1}^{(n)} = 0; \quad \rho_0^{(n)} = S_n, \quad n \geq 0; \quad \rho_{k+1}^{(n)} = \rho_{k-1}^{(n+1)} + \frac{x_{n+k+1} - x_n}{\rho_k^{(n+1)} - \rho_k^{(n)}}, \quad n \geq 0, \quad k \geq 1, \quad (2.67)$$

donde recordemos que los nodos de interpolación x_m satisfacen (2.43). El algoritmo fue derivado por Wynn, [72], para el caso $x_m = m + \beta$ con $\beta > 0$, lo que lleva a la forma estándar de (2.67)

$$\rho_{-1}^{(n)} = 0; \quad \rho_0^{(n)}, \quad n \geq 0; \quad \rho_{k+1}^{(n)} = \rho_{k-1}^{(n+1)} + \frac{k+1}{\rho_k^{(n+1)} - \rho_k^{(n)}}, \quad k \geq 1 \quad n \geq 0. \quad (2.68)$$

(El algoritmo (2.67) puede también deducirse de una recurrencia del tipo Neville-Aitken para la extrapolación racional, debida a Burlisch y Stoer, [20].)

Son varias las similitudes con el algoritmo ε . En primer lugar, la construcción de la tabla ρ , similar a la tabla ε , a partir de la fórmula recursiva con los cuatro términos

$$\begin{array}{ccc} & \rho_k^{(n)} & \rightarrow \rho_{k+1}^{(n)} \\ & \nearrow & \nearrow \\ \rho_{k-1}^{(n+1)} & \rightarrow \rho_k^{(n+1)} & \end{array}$$

por lo que el algoritmo puede también implementarse de forma similar. El cambio de variables $k+1 \rightarrow j$, $n+k+1 \rightarrow n$, permite reescribir (2.67) como

$$\begin{aligned} \rho_0^{(n)} &= S_n, \quad n \geq 0; \quad \rho_1^{(n-1)} = \frac{x_{n+1} - x_n}{S_{n+1} - S_n}, \quad n \geq 1, \\ \rho_j^{(n-j)} &= \rho_{j-2}^{(n-j+1)} + \frac{x_n - x_{n-j}}{\rho_{j-1}^{(n-j+1)} - \rho_{j-1}^{(n-j)}}, \quad n \geq 2, \quad 1 \leq j \leq n. \end{aligned} \quad (2.69)$$

Podemos entonces seguir la estrategia explicada en la sección 2.2.1 para el algoritmo ε , avanzando por las contradiagonales de la tabla ρ . Si almacenamos los valores $\rho_j^{(n-j)}$ en un vector

$$\rho_j^{(n-j)} \rightarrow R(n-j), \quad 0 \leq j \leq n,$$

y definimos otro vector para los nodos de interpolación

$$x_n \rightarrow \zeta(n), \quad n \geq 0,$$

(lo que no es necesario para la forma estándar (2.68)), se tiene el pseudocódigo para (2.69), dado por las fórmulas

$$\begin{aligned} R(n) &\leftarrow S(n), \quad n \geq 1; \quad R(n-1) \leftarrow \frac{\zeta(n) - \zeta(n-1)}{R(n) - R'(n-1)}, \quad n \geq 1; \\ R(n-j) &\leftarrow R'(n-j+1) + \frac{\zeta(n) - \zeta(n-j)}{R(n-j+1) - R'(n-j)}, \quad n \geq 2, \quad 2 \leq j \leq n, \end{aligned}$$

con R' un vector auxiliar que actualiza el vector R en cada paso, similar al vector E' utilizado en la implementación del algoritmo ε .

Observación 2.1. El algoritmo ρ permite a su vez generalizar el algoritmo ε ,

$$\begin{aligned} \varepsilon_{-1}^{(n)} &= 0, \quad \varepsilon_0^{(n)} = S_n, \quad n \geq 0, \\ \varepsilon_{k-1}^{(n)} &= \varepsilon_{k-1}^{(n+1)} + \frac{x_{n+k+1} - x_n}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}}, \quad k \geq 1, \quad n \geq 0, \end{aligned}$$

incorporando nodos de interpolación x_n (no necesariamente de la forma (2.43)), [15]. Ello relaciona también, en cierta manera, el algoritmo ε con los métodos de extrapolación racional.

Propiedades adicionales del algoritmo ρ pueden consultarse en varias referencias, como [13, 56, 71]. Las más relevantes para esta memoria son las relacionadas con la convergencia. En este sentido, el algoritmo ρ es complementario al algoritmo ε , puesto que no acelera (en general) la convergencia de sucesiones lineales (como sí hace la transformación de Shanks) y, en cambio, es útil en el caso de algunas sucesiones logarítmicas, donde falla el algoritmo ε , [56, 70].

2.3.3. Algunas aplicaciones

Para completar el estudio de la extrapolación polinómica y racional, en esta sección mencionamos dos aplicaciones clásicas: la integración de Romberg y los métodos de extrapolación para la resolución numérica de ecuaciones diferenciales ordinarias (EDOs).

2.3.3.1. Integración de Romberg

Una de las aplicaciones de la extrapolación de Richardson más frecuente consiste en mejorar la aproximación de integrales por reglas de cuadratura. Supongamos que

$$I[f] = \int_a^b f(x) dx, \quad (2.70)$$

2 Extrapolación escalar

con f definida e integrable en un intervalo $[a, b]$, es aproximada mediante la regla del trapecio

$$Q(h) = h \left(\frac{1}{2}f(a) + \sum_{j=1}^{n-1} f(a + jh) + \frac{1}{2}f(b) \right), \quad (2.71)$$

o del punto medio

$$Q(h) = h \sum_{j=0}^{n-1} f \left(a + \left(j + \frac{1}{2} \right) h \right), \quad (2.72)$$

con $h = (b - a)/n$, $n \geq 1$ entero. El grado de convergencia, cuando $h \rightarrow 0$, de $Q(h)$ hacia (2.70) depende de la regularidad del integrando f .

Hay diferentes formas de expresar el error de cuadratura $Q(h) - I[f]$ como función de h . La que aquí utilizamos se conoce como desarrollo de Euler-McLaurin, [4].

Teorema 2.16. *Sea $m \geq 0$ entero y supongamos que $f \in \mathcal{C}^{2m}([a, b])$. Entonces, existe $\xi_m(h) \in [a, b]$ tal que*

$$Q(h) = I[f] + \sum_{k=1}^{m-1} \frac{B_{2k}(\theta)}{(2k)!} (f^{2k-1}(b) - f^{2k-1}(a)) h^{2k} + R_m(h), \quad (2.73)$$

donde:

1. $R_m(h) = (b - a) \frac{B_{2m}(\theta)}{2m!} f^{2m}(\xi_m(b)) h^{2m}$,
2. $B_n(x)$ es el polinomio de Bernoulli de grado n

$$B_n(x) = \sum_{k=0}^n \binom{n}{k} B_k x^{n-k},$$

con B_k denotando el k -ésimo número de Bernoulli, definido por

$$\sum_{j=0}^k \binom{k+1}{j} B_j = 0, \quad k \geq 1.$$

3. $\theta = 0$ para la regla (2.71) y $\theta = 1/2$ para (2.72).

Observamos que (2.73) es un desarrollo de la forma (1.5) con $\sigma_j = 2j$. En particular, si $f \in \mathcal{C}^\infty([a, b])$, (2.73) es válido para todo $m \geq 0$ y puede sustituirse por un desarrollo asintótico infinito en el sentido especificado en (1.8)-(1.9).

Podemos entonces aplicar la extrapolación de Richardson a cualquiera de las reglas $Q(h)$ dadas por (2.71) ó (2.72). El esquema (2.47) correspondiente lleva el nombre de integración de Romberg, que fue el primero en proponerlo para el caso $a = 0$, $b = 1$, con $h_j = h_0 w^j$, $h_0 = 1$, $w = 1/2$, [50]. La tabla 2.3 ilustra los correspondientes errores de la tabla de Romberg para $f(x) = 1/(1 + x^2)$, $x \in [0, 1]$.

La convergencia por columnas viene justificada, además de por los resultados presentados en la sección 2.3.1, por el siguiente teorema que, por sencillez, se enuncia para $[a, b] = [0, 1]$ y cuya demostración puede consultarse en [56]. El teorema proporciona una expresión del error $S_k^{(n)} - I[f]$ con $S_k^{(n)}$ generado a partir de (2.71) ó (2.72), válido para una sucesión $h_n \downarrow 0$ arbitraria.

Teorema 2.17. *Supongamos que $f \in \mathcal{C}^\infty([0, 1])$. Entonces:*

1.039816e-02						
2.604046e-03	6.006535e-06					
6.510398e-04	3.778277e-08	3.601340e-07				
1.627604e-04	5.912427e-10	1.888193e-09	3.798249e-09			
4.069010e-05	9.239165e-12	2.956113e-11	5.906387e-14	1.495437e-11		
1.017252e-05	1.447731e-13	4.615197e-13	4.440892e-16	6.661338e-16	1.532108e-14	

Tabla 2.3: Errores de la integración de Romberg de $f(x) = 1/(1 + x^2)$.

1. Existe una función $w(t) \in C^\infty([0, 1])$ tal que $w(m^{-2}) = Q(m^{-1})$, $m = 1, 2, \dots$, y de manera que

$$S_k^{(n)} - I[f] = (-1)^k \frac{w^{(k+1)}(t_{k,n})}{(k+1)!} \left(\prod_{i=0}^k h_{n+i} \right)^2,$$

para ciertos $t_{k,n} \in (t_{n+k}, t_n)$, con $t_n = h_n^2$. En particular, para h_n arbitrario

$$S_k^{(n)} - I[f] = \mathcal{O}((h_n \cdots h_{n+k})^2), \quad n \rightarrow \infty.$$

2. Si $\frac{h_{m+1}}{h_m} \sim 1$, $m \rightarrow \infty$, entonces existe $\mu \geq 0$ tal que

$$S_k^{(n)} - I[f] = \mathcal{O}(h_n^{2(k+1+\mu)}), \quad n \rightarrow \infty, \quad k \geq 1.$$

Así, cada columna de la tabla de Romberg converge al menos tan rápidamente como la precedente.

Como se indica en [56], la elección de la sucesión h_n es importante a efectos computacionales. El número de evaluaciones del integrando para $Q(h_n)$ cuando $h_n = w^n$ crece como $1/w^n$. Por ejemplo, para la elección clásica $w = 1/2$, calcular $S_0^{(n)}$ a partir de $Q(h_k)$, $0 \leq i \leq n$, requiere 2^n evaluaciones del integrando. Por tanto, el paso de n a $n+1$ dobla el número de evaluaciones. Para reducir este coste computacional, se suelen utilizar sucesiones alternativas como $h_n = 1/(n+1)$ o bien verificando $h_{n+1} \leq h_n w$ para algún $w \in (0, 1)$.

2.3.3.2. Método de extrapolación para la resolución numérica de EDOs

Otra de las aplicaciones clásicas de la extrapolación de Richardson polinómica se encuentra en la resolución numérica de EDOs. Consideremos un problema de valores iniciales

$$\begin{cases} y'(x) = f(x, y) \\ y(x_0) = y_0 \end{cases} \quad (2.74)$$

con f suficientemente regular, y que suponemos con solución única $y(x)$. Para la aproximación numérica de (2.74), tomemos un método de un paso

$$y_{n+1} = y_n + h\phi(x_n, y_n, h), \quad (2.75)$$

con $h > 0$ la longitud de paso y ϕ la función incremento. Es conocido, [31], que para ϕ suficientemente regular, el método (2.75) admite, en cada punto de la solución $y(x)$, un error

2 Extrapolación escalar

local en la forma

$$y(x+h) - y(x) - h\phi(x, y(x), h) = d_{p+1}(x)h^{p+1} + \dots + d_{N+1}(x)h^{N+1} + \mathcal{O}(h^{N+2}), \quad (2.76)$$

para ciertas funciones $d_j(x)$ que dependen de f y sus derivadas, siendo p el orden de consistencia del método. La existencia de (2.76) y la condición $\phi(x, y, 0) = f(x, y)$ implican, para ϕ suficientemente regular, la existencia de un desarrollo del error global

$$y(x) - y_h(x) = e_p(x)h^p + \dots + e_N(x)h^N + E_N(x)h^{N+1}, \quad (2.77)$$

para ciertas funciones $e_j(x)$, $j = p, \dots, N$, con $e_j(x_0) = 0$, y $E_N(x)$ acotada, siendo $y_n := y_h(x)$ la solución numérica en $x = x_0 + nh$.

El desarrollo (2.77) es la base para los llamados métodos de extrapolación, [31]: fijados enteros positivos $m_0 < m_1 < \dots$, definimos longitudes de paso $h_j = h/m_j$, $j \geq 0$, y calculamos aproximaciones numéricas a (2.74) con m_i pasos de longitud h_i usando (2.75), de modo que generamos la sucesión

$$S_0^{(n)} := y_{h_n}(x_0 + h), \quad n = 0, 1, \dots \quad (2.78)$$

Buscamos ahora mejorar las aproximaciones (2.78) a partir del desarrollo (2.77) y los problemas de interpolación

$$S_0^{(l)} = S_k^{(n)} - \sum_{j=0}^{k-1} e_{p+j} h_1^{p+l}, \quad l = n, n+1, \dots, n+k, \quad (2.79)$$

consistentes en $k+1$ ecuaciones lineales para $S_k^{(n)}, e_p, \dots, e_{p+k-1}$. Si redefinimos las incógnitas $\tilde{e}_{p+j} = h^p e_{p+j}$, $j = 0, \dots, k-1$, entonces (2.79) se ajusta a la formulación (2.45) del método PREP con $r = 1$. De este modo, $S_k^{(n)}$ puede entenderse como la extrapolación de Richardson polinómica en el límite $h \downarrow 0$. El ejemplo clásico $m_0 = 1$, $m_1 = 2$, $k = 1$, es también conocido como extrapolación de Richardson, [31]: calculados

$$S_0^{(0)} = y_h(x_0 + h), \quad S_1^{(0)} = y_{h/2}(x_0 + h),$$

como las aproximaciones a $y(x_0 + h)$ con (2.75) usando un paso de longitud h y dos pasos de longitud $h/2$ respectivamente, la tabla de Romberg proporciona el valor

$$S_0^{(1)} = \frac{S_1^{(0)} - c_1 S_0^{(0)}}{1 - c_1}, \quad c_1 = \frac{1}{2^p}, \quad (2.80)$$

que es una aproximación de orden $p+1$ a $y(x_0 + h)$, [31].

A continuación se ilustran estos resultados con la comparación de la solución numérica al siguiente problema de valores iniciales, [31]

$$\begin{cases} y' = (-y \operatorname{sen}(x) + 2 \tan(x))y, & 0 \leq x \leq \frac{\pi}{6}, \\ y(0) = 1, \end{cases} \quad (2.81)$$

con solución $y(x) = 1/\cos(x)$. En la primera columna de la tabla 2.4 se presentan los errores generados por el método de Euler explícito para pasos $h_n = 2^{-n} 0.2 \pi/6$, $n = 1, 2, \dots, 8$, en el punto $x = \pi/6$. Se puede observar cómo el orden del método de Euler es $p = 1$. La segunda

columna de la tabla 2.4 muestra los errores calculados con el método (2.80), que mejoran los obtenidos con el método de Euler. Para este método de extrapolación el orden es de $p + 1 = 2$, confirmando el comentario anterior.

Euler explícito	Método (2.80)
1.7256e-02	
8.6776e-03	9.9779e-05
4.3511e-03	2.4622e-05
2.1786e-03	6.1101e-06
1.0901e-03	1.5216e-06
5.4523e-04	3.7962e-07
2.7266e-04	9.4808e-08
1.3634e-04	2.3690e-08

Tabla 2.4: Comparación del error en $x = \pi/6$ para el problema (2.81) con el método de Euler explícito y el método (2.80) y longitudes de paso $h_n = 2^{-n} 0.2 \pi/6$, $n = 1, 2, \dots, 8$.

Generalizando el resultado de convergencia de la extrapolación (2.80), se tiene:

Teorema 2.18. El valor $S_k^{(n)}$ obtenido de (2.79) representa una aproximación a $y(x_0 + h)$ de orden $p + k$.

Demostración:

Comparamos (2.79) y el desarrollo (2.77), con $N = p + k$, escrito en la forma

$$y_{h_l}(x_0 + h) = S_0^{(l)} = y(x_0 + h) - e_p(x_0 + h)h_l^p - \dots - e_p(x_0 + h)h_l^{p+k-1} - \Delta_l, \quad l = n, n+1, \dots, n+k, \quad (2.82)$$

con

$$\Delta_l = e_{p+k}(x_0 + h)h_l^{p+k} + E_{h_l}(x_0 + h)h_l^{p+k+1} = \mathcal{O}(h^{p+k}).$$

(La última igualdad se debe a que $e_{p+k}(x_0) = 0$ y $h_l \leq h$, $l = n, \dots, n+k$.) Notemos que (2.82) puede verse como un sistema lineal para las incógnitas

$$y(x_0 + h), h^p e_p(x_0 + h), \dots, h^{p+k-1} e_{p+k-1}(x_0 + h),$$

y con matriz de Vandermonde

$$A = \begin{pmatrix} 1 & \frac{-1}{m_n^p} & \dots & \frac{-1}{m_n^{p+k-1}} \\ 1 & \frac{-1}{m_{n+1}^p} & \dots & \frac{-1}{m_{n+1}^{p+k-1}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \frac{-1}{m_{n+k}^p} & \dots & \frac{-1}{m_{n+k}^{p+k-1}} \end{pmatrix},$$

(donde $h_j = h/m_j$, $j \geq 0$). Se trata entonces de una perturbación de (2.79) con términos $\Delta_l = \mathcal{O}(h^{p+k})$. Puede comprobarse que A es invertible y, comparando los dos sistemas, se tiene

$$|y(x_0 + h) - S_k^{(n)}| \leq \|A^{-1}\|_\infty \max |\Delta_l| = \mathcal{O}(h^{k+n}). \quad \square$$

2 Extrapolación escalar

Los términos $h_j = h/m_j$, $j \geq 0$, más utilizados en la práctica son:

- Sucesión de Romberg: $m_j = 2^j$, $j \geq 0$. De este modo, $h_j = hw^j$ con $w = 1/2$.
- Sucesión de Burlisch: $m_0 = 1$ y

$$m_{2j+1} = 2^j, \quad j \geq 0; \quad m_{2j} = 3 \cdot 2^{j-1} - \frac{3}{2} \cdot 2^j, \quad j \geq 1.$$

En este caso,

$$h_{2j+1} = \frac{h}{2^j} = \frac{h\sqrt{2}}{(\sqrt{2})^{2j+1}}, \quad h_{2j} = \frac{2h}{3} \frac{1}{2^j} = \frac{2h}{3} \frac{1}{(\sqrt{2})^{2j}}.$$

- Sucesión armónica: $m_j = j + 1$, con $h_j = h/(j + 1)$, $j \geq 0$.

2.4. Otros métodos

En la línea de los comentarios sobre el propósito de esta memoria, realizados en la introducción, presentamos aquí una breve descripción de otros procedimientos y técnicas utilizadas en la extrapolación escalar.

2.4.1. Métodos iterados

Para incrementar el orden de una transformación se suele hacer uso de la llamada técnica de iteración. Partiendo de un método de extrapolación escrito de la forma

$$T^{(n)} = \phi_n(\{S_m\}_m), \quad n \geq 0,$$

se define el método iterado asociado a partir de las fórmulas

$$\mathcal{T}_0^{(n)} = S_n, \quad n \geq 0; \quad \mathcal{T}_{k+1}^{(n)} = \phi_n(\{\mathcal{T}^{(m)}\}_m), \quad n, k \geq 0.$$

El ejemplo más frecuente en la literatura obtenido con este procedimiento es el basado en el método de Aitken, [56, 33, 63].

2.4.2. La transformación de Levin

Con el objetivo de mejorar el rendimiento de métodos clásicos, como la transformación de Shanks y la extrapolación de Richardson, la literatura ofrece varias posibilidades. En el primer caso podemos mencionar, a título representativo, la llamada transformación de Levin, [40]. Ésta puede verse como una generalización del proceso de Aitken y de la transformación E_1 de la primera columna del algoritmo de extrapolación general.

Ya hemos mencionado que el método de Aitken es exacto para sucesiones de la forma

$$S_n = S + a\Delta S_n, \quad n = 0, 1, \dots,$$

mientras que el caso de E_1 el núcleo es

$$S_n = S + af(n), \quad n = 0, 1, \dots,$$

con $f(n)$ una función conocida, y donde en ambas sucesiones a es constante. La generalización propuesta por la transformación de Levin se basa en modificar estos núcleos. Específicamente, su construcción parte de que el método es exacto para sucesiones

$$S_n = S + w_n \sum_{j=0}^{k-1} \frac{c_j}{(n+\beta)^j}, \quad n = 0, 1, \dots \quad (2.83)$$

donde $\beta > 0$, c_j es constante y los estimativos w_n se suponen distintos para todo n .

Tomando de (2.83) el correspondiente sistema con S_n, \dots, S_{n+k} , para las $k+1$ incógnitas S, c_0, \dots, c_{k-1} y asumiendo que tiene solución única, la transformación puede escribirse, como en el caso del algoritmo general, en la forma de cociente de dos determinantes

$$\mathcal{T}_k^{(n)}(\beta, S_n, w_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \dots & S_{n+k} \\ w_n & w_{n+1} & \dots & w_{n+k} \\ \vdots & \vdots & \ddots & \vdots \\ w_n & w_{n+1} & \dots & w_{n+k} \end{vmatrix}}{\begin{vmatrix} (n+\beta)^{k-1} & (n+1+\beta)^{k-1} & \dots & (n+\beta+k)^{k-1} \\ 1 & 1 & \dots & 1 \\ w_n & w_{n+1} & \dots & w_{n+k} \\ \vdots & \vdots & \ddots & \vdots \\ w_n & w_{n+1} & \dots & w_{n+k} \end{vmatrix}}. \quad (2.84)$$

(Para una representación alternativa, véase [56].) Notemos que la transformación depende de la sucesión a extrapolar S_n , de β y de los estimativos w_n .

Si bien, para la definición, es suficiente con que $\beta \neq -n$ para todo n , se suele elegir $\beta > 0$ por sencillez. Por otro lado, en su artículo original, [40], Levin consideró tres tipos de sucesiones w_n ; si $a_n = \Delta S_{n-1}$ (con $a_1 = S_1$), entonces:

- $w_n = a_n$. El método (2.84) resultante fue llamado transformación t .
- $w_n = na_n$ ó $(n+\beta)a_n$. El método (2.84) así obtenido se llama transformación u .
- $w_n = \frac{a_n a_{n+1}}{a_{n+1} - a_n}$, obteniéndose la llamada transformación v .

Sidi, [56], demostró algunos resultados de convergencia para estos tres casos. En particular, comprobó que las transformaciones u y v aceleran la convergencia lineal y logarítmica, mientras que la transformación t rinde bien en el caso de convergencia lineal. También confirmó la estabilidad de los tres procedimientos. Por último, en [26] puede encontrarse una programación eficiente de (2.84), basada en una fórmula de recurrencia de tres términos, véase también [70].

2.4.3. El algoritmo θ

Un segundo ejemplo de mejora de rendimiento es el llamado algoritmo θ , cuyo origen procede de las propiedades complementarias de convergencia de los algoritmos ε y ρ , y la consiguiente búsqueda de un método que las combine. Con este fin, Brezinski, [11], consideró un esquema recursivo de la forma

$$T_{-1}^{(n)} = 0, \quad T_0^{(n)} = S_n, \quad T_{k+1}^{(n)} = T_{k-1}^{(n+1)} + w_k D_k^{(n)}, \quad k, n = 0, 1, \dots \quad (2.85)$$

2 Extrapolación escalar

con $D_k^{(n)}$ una cantidad dependiente de los elementos $T_j^{(n)}$, mientras que w_k es indeterminado. Casos particulares de (2.85) son precisamente el algoritmo ε (con $w_k = 1$ y $D_k^{(n)} = \frac{1}{T_k^{(n+1)} - T_k^{(n)}}$) y el algoritmo ρ (con $w_k = 1$ y $D_k^{(n)} = \frac{x_{n+k+1} - x_n}{T_k^{(n+1)} - T_k^{(n)}}$).

Suponiendo que $D_k^{(n)}$ es conocido, Brezinski se centró en la búsqueda de valores w_k que garanticen para $T_k^{(n)}$ la aceleración de convergencia. Para ello, comparó con los algoritmos ε y ρ . Como en estos casos, si uno asume que sólo los términos $T_k^{(n)}$ con k par proporcionan aproximaciones al límite (o antilímite), entonces parece natural la elección $w_{2k} = 1$. Por otro lado, los parámetros con índice impar w_{2k+1} pueden determinarse a partir de imponer la propiedad de que $T_{2k+2}^{(n)}$ converja más rápidamente que $T_{2k}^{(n+1)}$, en el sentido de que

$$\lim_{n \rightarrow \infty} \frac{\Delta T_{2k+2}^{(n)}}{\Delta T_{2k}^{(n+1)}} = 0, \quad (2.86)$$

donde recordamos que Δ representa el operador de diferencias progresivas con respecto a n . Utilizando (2.85), la condición (2.86) se cumple eligiendo

$$w_{2k+1} = - \lim_{n \rightarrow \infty} \frac{\Delta T_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}}.$$

En la práctica, Brezinski sugiere tomar los ratios

$$w_{2k+1} = - \frac{\Delta T_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}}.$$

Sustituyendo en (2.85), tenemos el esquema recursivo

$$T_{-1}^{(n)} = 0; \quad T_0^{(n)} = S_n,$$

$$T_{2k+1}^{(n)} = T_{2k-1}^{(n+1)} + D_{2k}^{(n)}; \quad T_{2k+2}^{(n)} = T_{2k}^{(n+1)} - \frac{\Delta T_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}} D_{2k+1}^{(n)}, \quad k, n = 0, 1, \dots,$$

que debe distinguir entre subíndices pares e impares, debido a la diferente definición de los w_k . De este modo, eligiendo $D_{2k}^{(n)}$ como en el algoritmo ε , se obtiene el llamado **algoritmo θ** :

$$\theta_{-1}^{(n)} = 0, \quad \theta_0^{(n)} = S_n,$$

$$\theta_{2k+1}^{(n)} = \theta_{2k-1}^{(n+1)} + \frac{1}{\Delta \theta_{2k}^{(n)}}; \quad \theta_{2k+2}^{(n)} = \theta_{2k}^{(n+1)} + \frac{\Delta \theta_{2k}^{(n+1)}}{\Delta^2 \theta_{2k+1}^{(n)}} \Delta \theta_{2k+1}^{(n+1)}, \quad k, n = 0, 1, \dots \quad (2.87)$$

Brezinski comprobó que (2.87) acelera la convergencia de sucesiones linealmente convergentes y de sucesiones con convergencia logarítmica, siendo estable en el primer caso pero no en el segundo, [56]. Al igual que en su derivación, la programación del algoritmo θ puede tener también en cuenta la correspondiente a los algoritmos ε y ρ . De este modo, los elementos

$\theta_j^{(m)}$ pueden disponerse en forma de tabla (la tabla θ) y distinguir la programación para los subíndices pares e impares

$$\begin{array}{ccc}
 & \nearrow & \\
 & \theta_{2k}^{(n)} & \rightarrow \theta_{2k+1}^{(n)} \\
 \theta_{2k-1}^{(n+1)} & \rightarrow \theta_{2k}^{(n+1)} & \nearrow
 \end{array}
 \qquad
 \begin{array}{ccc}
 & \nearrow & \\
 & \theta_{2k+1}^{(n)} & \rightarrow \theta_{2k+2}^{(n)} \\
 \theta_{2k}^{(n+1)} & \rightarrow \theta_{2k+1}^{(n+1)} & \nearrow \\
 & \nearrow & \\
 \theta_{2k}^{(n+2)} & \rightarrow \theta_{2k+1}^{(n+2)} &
 \end{array}$$

La implementación requiere entonces dos arrays; los detalles pueden consultarse en [70].

3 Extrapolación vectorial

La extrapolación es también aplicable a procesos vectoriales de convergencia lenta o directamente divergentes. A juzgar por la literatura consultada, da la impresión de que este tópico es menos conocido y, al mismo tiempo, donde la teoría de aceleración de convergencia tiene un mayor desarrollo en la actualidad. Ello puede deberse a que muchos procesos secuenciales vectoriales aparecen en la aproximación a modelos continuos (como la discretización de ecuaciones en derivadas parciales, o de ecuaciones integrales) y suelen ser, en general, de tamaño grande. En otras situaciones, el proceso a estudio es ya discreto, pero puede tener una estructura especial que influya en la elección del método de extrapolación a aplicarse. El tercer capítulo de esta memoria se dedica entonces a la extrapolación vectorial. Contiene tres partes. La primera es introductoria, y en ella presentaremos los elementos básicos de la teoría, algunos de los cuales son extensiones, al caso vectorial, de los mencionados en la sección 2.1 para la extrapolación escalar. En las secciones 3.2 y 3.3 analizaremos dos grupos de métodos de aceleración para el caso vectorial: los de tipo ε (basados en el algoritmo ε escalar) y los llamados polinómicos. Seguiremos aquí una estructura de presentación similar a la del capítulo 2: presentación de los métodos, implementación y convergencia. La sección 3.4 está dedicada a las aplicaciones. Su objetivo es ilustrar el uso de la extrapolación vectorial en las situaciones mencionadas más arriba y se centrará en dos problemas: la aceleración de procesos de Markov discretos y la búsqueda de aproximaciones eficientes a soluciones especiales de ecuaciones en derivadas parciales.

3.1. Introducción

3.1.1. Conceptos básicos

Comenzamos extendiendo algunas definiciones dadas en la sección 2.1 al caso vectorial. Como en el capítulo 2, consideraremos, por sencillez, sucesiones vectoriales reales.

Definición 3.1. Fijados enteros $N > 1$, $l \geq 0$, un método de extrapolación vectorial es una aplicación

$$\phi : \mathbb{R}^N \times \overset{(l+1)}{\dots} \times \mathbb{R}^N \longrightarrow \mathbb{R}^N$$

de modo que, a partir de una sucesión $\{S_n\}_{n=0}^{\infty}$ con $S_n \in \mathbb{R}^N$, genera una sucesión transformada $\{T_n\}_{n=0}^{\infty}$, con

$$T_n = \phi(S_n, S_{n+1}, \dots, S_{n+l}),$$

siendo $l \geq 0$ el orden del método.

Como en el caso escalar, la representación alternativa del método consiste en incorporar el orden l como segundo índice y escribir

$$T_{n,l} = T_n^{(l)} = \sum_{i=0}^l \gamma_{n,i} S_{n+i}, \quad (3.1)$$

3 Extrapolación vectorial

con $\gamma_{n,i} = \gamma_{n,i}(S_m)$ verificando $\sum_{i=0}^l \gamma_{n,i} = 1$ (cf. sección 1.2.3). La formulación anterior permite asimismo disponer la sucesión transformada en forma de tabla de extrapolación, donde cada elemento representa un vector en \mathbb{R}^N .

Como en el caso escalar, aunque la aplicación ϕ que define un método de extrapolación es, en general, no lineal, una propiedad deseable es la casi-linealidad o invariancia por traslaciones

$$\phi(\alpha x_0 + \beta, \dots, \alpha x_l + \beta) = \alpha \phi(x_0, \dots, x_l) + \beta, \quad (3.2)$$

donde $\beta \in \mathbb{R}^N$, $\alpha \in \mathbb{R}$, $x_0, \dots, x_l \in \mathbb{R}^N$.

Definición 3.2. Supongamos que $S = \lim_{n \rightarrow \infty} S_n$. Se dice que un método de extrapolación vectorial ϕ es exacto para S_n , si a partir de un n ,

$$\phi(S_n, S_{n+1}, \dots, S_{n+l}) = S.$$

Aquellas sucesiones exactas para ϕ constituyen el núcleo de la transformación.

Definición 3.3. Se dice que el método de extrapolación (3.1) es estable si $\sup_n \Gamma_n < \infty$ donde

$$\Gamma_n := \sum_{i=0}^l |\gamma_{n,i}|.$$

En otro caso, se dice que es inestable.

3.1.2. Implementación en modo cíclico

La computación de $T_{n,l}$ en (3.1) requiere, en principio, el almacenamiento de $l+1$ vectores de tamaño N . Por otra parte, la convergencia de la extrapolación se obtiene típicamente manteniendo l fijo y aumentando n . Así, si el tamaño N de los vectores es grande, es necesario reducir el coste computacional en la implementación de (3.1). Para ello, se introduce el llamado modo cíclico [59, 60], que puede ser de dos tipos:

- Modo cíclico completo. Consta de los pasos siguientes:

C₁. Fijamos n, l y un vector S_0 .

C₂. Generamos $S_1, S_2, \dots, S_{n+l+1}$ con el procedimiento que elijamos para la sucesión a extrapolar.

C₃. Extrapolamos para obtener $T_{n,l}$ usando (3.1).

C₄. Si $T_{n,l}$ no satisface el criterio de parada establecido sobre el proceso, fijamos $S_0 = T_{n,l}$ y volvemos al paso (C₂).

Cada aplicación de los pasos (C₁)-(C₄) se llama ciclo. Si es necesario, suele denotarse por $T_{n,l}^{(r)}$ al vector $T_{n,l}$ calculado en el ciclo r -ésimo. Más adelante se justificará la necesidad de calcular $n+l+1$ términos en (C₂) para implementar el ciclo, a pesar de que la fórmula (3.1) sólo involucra los términos S_0, S_1, \dots, S_{n+l} .

- Modo cíclico 'congelado'. En este caso, se implementa inicialmente un número p de ciclos completos para calcular $T_{n,l}^{(r)}$, $r = 1, \dots, p$, guardándose los últimos coeficientes $\gamma_{n,i} = \gamma_{n,i}^{(p)}$ usados para calcular $T_{n,l}^{(p)}$. El cómputo de los ciclos siguientes $T_{n,l}^{(r)}$, $r =$

$p+1, p+2, \dots$, se realiza a través de los pasos (C₁)-(C₄) pero donde (C₃) se modifica calculando

$$T_{n,l} = \sum_{i=0}^l \gamma_{n,i}^{(p)} S_{n+i}.$$

Este procedimiento tiene algunas ventajas computacionales añadidas. Por otra parte, se puede combinar este proceso con una actualización periódica de los coeficiente $\gamma_{n,i}$, siendo posible, también, una implementación en paralelo si el problema lo requiere. Véase [59] para más detalles.

3.1.3. Polinomios mínimos asociados a una matriz

Para la construcción de uno de los grupos de métodos de extrapolación vectorial, los llamados métodos polinómicos, utilizaremos los conceptos de polinomio mínimo de una matriz y polinomio mínimo de una matriz con respecto a un vector, que recordamos a continuación.

Definición 3.4. Sea $A \in \mathbb{R}^{N \times N}$. Se dice que un polinomio mónico $Q(\lambda)$ es mínimo para A si $Q(A) = 0$ y su grado m es mínimo de entre todos los polinomios que anulan A .

Teorema 3.1. 1. El polinomio $Q(\lambda)$ de A existe y es único.

2. Si $Q_1(\lambda)$ es un polinomio mónico de grado mayor que m y $Q_1(A) = 0$ entonces $Q(\lambda)$ divide a $Q_1(\lambda)$. En particular, $Q(\lambda)$ divide al polinomio característico de A

$$P_A(\lambda) = \det(\lambda I - A),$$

por lo que su grado m es menor o igual que N y sus raíces son autovalores de A .

Demostración:

1. Por el teorema de Cayley-Hamilton, $P_A(A) = 0$; por tanto, existe $Q(\lambda)$ mónico y de grado mínimo $m \leq N$ con $Q(A) = 0$.

Para demostrar la unicidad, supongamos que \hat{Q} es un polinomio mónico de grado m que anula la matriz A y es distinto de $Q(\lambda)$. Entonces, el polinomio

$$S(\lambda) = Q(\lambda) - \hat{Q}(\lambda),$$

tiene grado menor que m y anula la matriz A , lo que no es posible.

2. Si $Q_1(\lambda)$ tiene grado mayor que m y $Q_1(A) = 0$, se tiene que

$$Q_1(\lambda) = C(\lambda)Q(\lambda) + R(\lambda),$$

con $R(\lambda)$ un polinomio de grado menor que m y tal que $R(A) = 0$. Por tanto, $R = 0$ y $Q(\lambda)$ divide a $Q_1(\lambda)$. \square

La propiedad 2 del teorema 3.1 puede utilizarse para la construcción del polinomio mínimo, a partir de los bloques de la forma canónica de Jordan.

Definición 3.5. El polinomio mínimo de $A \in \mathbb{R}^{N \times N}$ con respecto a un vector $v \in \mathbb{R}^N$ se define como el polinomio P de grado mínimo tal que $P(A)v = 0$.

Teorema 3.2. 1. El polinomio P de la definición 3.5 existe y es único.

2. Si P_1 es un polinomio tal que $P_1(A)v = 0$, con grado mayor que el de P , entonces P divide a P_1 . En particular, P divide al polinomio mínimo Q de A .

La propiedad 2 del teorema 3.2 permite construir en la práctica el polinomio P , a partir del polinomio mínimo Q de A y la representación del vector v en la base de Jordan.

3.2. Métodos de extrapolación vectorial

3.2.1. Algoritmos tipo ε vectorial

El primer grupo de métodos vectoriales de aceleración de convergencia tiene como base el algoritmo ε escalar, tratado en el capítulo 2 de la presente memoria. Aquí son destacables dos procedimientos: el algoritmo ε vectorial (VEA) y el algoritmo ε topológico (TEA).

Algoritmo ε vectorial (VEA)

Para generalizar la formulación del algoritmo ε de aceleración de sucesiones escalares es necesario introducir el concepto de inversa de un vector no nulo¹. Wynn, [74, 75], propuso una extensión basada en la inversa de Samuelson

$$y^{-1} := \frac{\bar{y}}{\langle y, y \rangle}, \quad y \in \mathbb{C}^N, \quad y \neq 0, \quad (3.3)$$

donde \bar{y} denota el vector conjugado de y , con $\langle \cdot, \cdot \rangle$ dado por el producto euclídeo en \mathbb{C}^N con norma asociada $\|\cdot\|_2$. La inversa (3.3) de y está relacionada con la pseudoinversa y^+ a través de la fórmula, [59]

$$y^{-1} = (y^+)^T.$$

En lo que sigue, y^{-1} también se denotará como $1/y$. No es difícil comprobar que la definición introducida en (3.3) verifica las propiedades

$$(y^{-1})^{-1} = y, \quad y^T y^{-1} = 1.$$

Usando (3.3), Wynn traslada las fórmulas del algoritmo ε escalar al caso vectorial, generando el algoritmo VEA:

1. $\varepsilon_{-1}^{(n)} = 0, \varepsilon_0^{(n)} = S_n, n = 0, 1, \dots,$

- 2.

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\Delta \varepsilon_k^{(n)}}, \quad n, k = 0, 1, \dots, \quad (3.4)$$

donde $\Delta \varepsilon_k^{(n)} = \varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}$.

Como es de esperar (dado que ése es el objetivo) muchas de las propiedades del algoritmo ε escalar pueden extenderse al caso vectorial. La mayoría tiene que ver con la implementación. Así, tenemos:

¹La aplicación de la transformación de Shanks a la sucesión vectorial $\{S_n\}_{n=0}^{\infty}$ componente a componente no es recomendada por su inestabilidad numérica, cf [59].

(P₁) Los valores (3.4) pueden disponerse en forma de tabla, versión vectorial de la tabla ε , a partir de la molécula computacional de cuatro términos vectoriales

$$\begin{array}{ccc} & \varepsilon_k^{(n)} & \rightarrow \varepsilon_{k+1}^{(n)} \\ & \nearrow & \nearrow \\ \varepsilon_{k-1}^{(n+1)} & \rightarrow \varepsilon_k^{(n+1)} & \end{array} .$$

(P₂) Como en el caso escalar, las aproximaciones al límite (o antilímite) de S_n vienen dadas por los elementos $\varepsilon_{2k}^{(n)}$, cuyo cálculo requiere los $2k + 1$ vectores $\varepsilon_0^{(n+s)} = S_{n+s}$, $s = 0, 1, \dots, 2k$.

(P₃) Según (P₁), calcular $\varepsilon_{2k}^{(n)}$ requiere entonces obtener los valores

$$\varepsilon_r^{(n+j)}, \quad 1 \leq j+r \leq 2k, \quad r \geq 1.$$

Teniendo en cuenta que

$$\varepsilon_1^{(n)} = \frac{1}{\Delta \varepsilon_0^{(n)}}, \quad n \geq 0,$$

entonces el coste computacional, en operaciones vectoriales con N componentes, para calcular $\varepsilon_{2k}^{(n)}$, es $6k^2 + O(k)$. Es necesario almacenar los $2k + 1$ vectores S_n, \dots, S_{n+2k} (en total $(2k + 1)N$ posiciones de memoria) si bien el algoritmo puede ir sobrescribiendo en las mismas variables los vectores $\varepsilon_r^{(m)}$, utilizando un procedimiento similar al del caso escalar.

(P₄) En la formulación del modo cíclico (sección 3.1.2) se tiene en este caso que $l = 2k - 1$.

Algoritmo ε topológico (TEA)

La vectorización del algoritmo ε escalar depende entonces, como se ha visto, de la definición de la inversa de un vector no nulo que se utilice. Una nueva versión del método (el algoritmo ε topológico o TEA) fue propuesto por Brezinski, [12, 13], utilizando un concepto diferente.

Definición 3.6 (Inversa de un vector con respecto a otro). Sean $x, y \in \mathbb{C}^N$, con $\langle x, y \rangle \neq 0$. Se define el inverso del par ordenado $\{x, y\}$ como el par ordenado $\{x^{-1}, y^{-1}\}$, donde

$$y^{-1} = \frac{x}{\langle y, x \rangle}, \quad x^{-1} = \frac{y}{\langle y, x \rangle}. \quad (3.5)$$

Se dice que x^{-1} es el inverso de x con respecto a y y viceversa.

Es un ejercicio comprobar que (3.5) verifica las propiedades

$$(\{x, y\}^{-1})^{-1} = \{x, y\}, \quad \langle x^{-1}, y^{-1} \rangle = \frac{1}{\langle y, x \rangle}.$$

Con esta nueva definición, Brezinski propone dos versiones vectoriales del algoritmo ε (3.4),

3 Extrapolación vectorial

que necesitamos primero reescribir como

$$\begin{aligned}\varepsilon_{2k+1}^{(n)} &= \varepsilon_{2k-1}^{(n+1)} + \frac{1}{\Delta\varepsilon_{2k}^{(n)}}, \\ \varepsilon_{2k+2}^{(n)} &= \varepsilon_{2k}^{(n+1)} + \frac{1}{\Delta\varepsilon_{2k+1}^{(n)}}, \quad n, k = 0, 1, \dots,\end{aligned}\tag{3.6}$$

donde, como antes, el inverso x^{-1} se denota también como $1/x$. Las dos versiones dependen de la interpretación de los inversos en (3.6).

- **TEA1.** Fijamos $y \in \mathbb{R}^N$ arbitrario, verificando $\langle y, \Delta\varepsilon_{2k}^{(n)} \rangle \neq 0$, para cada n y k . Definimos:
 1. $1/\Delta\varepsilon_{2k}^{(n)}$ como el inverso de $\Delta\varepsilon_{2k}^{(n)}$ con respecto a y .
 2. $1/\Delta\varepsilon_{2k+1}^{(n)}$ como el inverso de $\Delta\varepsilon_{2k+1}^{(n)}$ con respecto a y^{-1} , el inverso de y con respecto a $\Delta\varepsilon_{2k}^{(n)}$, es decir: $y^{-1} = \frac{\Delta\varepsilon_{2k}^{(n)}}{\langle y, \Delta\varepsilon_{2k}^{(n)} \rangle}$.

Entonces, usando (3.5), la formulación (3.6) se escribe como

$$\begin{aligned}\varepsilon_{2k+1}^{(n)} &= \varepsilon_{2k-1}^{(n+1)} + \frac{y}{\langle \Delta\varepsilon_{2k}^{(n)}, y \rangle}, \\ \varepsilon_{2k+2}^{(n)} &= \varepsilon_{2k}^{(n+1)} + \frac{\Delta\varepsilon_{2k}^{(n)}}{\langle \Delta\varepsilon_{2k+1}^{(n+1)}, \Delta\varepsilon_{2k}^{(n)} \rangle}, \quad n = 0, 1, \dots,\end{aligned}\tag{3.7}$$

para $k = 0, 1, \dots$

- **TEA2.** Con $y \in \mathbb{R}^N$ arbitrario verificando la condición anterior para TEA1, se definen:
 1. $1/\Delta\varepsilon_{2k}^{(n)}$ como el inverso de $\Delta\varepsilon_{2k}^{(n)}$ con respecto a y .
 2. $1/\Delta\varepsilon_{2k+1}^{(n)}$ como el inverso de $\Delta\varepsilon_{2k+1}^{(n)}$ con respecto a y^{-1} , siendo y^{-1} el inverso de y con respecto a $\Delta\varepsilon_{2k}^{(n+1)}$: $y^{-1} = \frac{\Delta\varepsilon_{2k}^{(n+1)}}{\langle y, \Delta\varepsilon_{2k}^{(n+1)} \rangle}$.

Ello lleva a formular (3.6) como

$$\begin{aligned}\varepsilon_{2k+1}^{(n)} &= \varepsilon_{2k-1}^{(n+1)} + \frac{y}{\langle \Delta\varepsilon_{2k}^{(n)}, y \rangle}, \\ \varepsilon_{2k+2}^{(n)} &= \varepsilon_{2k}^{(n+1)} + \frac{\Delta\varepsilon_{2k}^{(n+1)}}{\langle \Delta\varepsilon_{2k+1}^{(n)}, \Delta\varepsilon_{2k}^{(n+1)} \rangle}, \quad n = 0, 1, \dots,\end{aligned}\tag{3.8}$$

para $k = 0, 1, \dots$

Comparando los algoritmos (3.7)-(3.8) con el algoritmo ε vectorial de Wynn (3.4), se pueden hacer las siguientes observaciones. Por un lado, la aproximación al límite o antilímite de S_n , tanto para TEA1 como para TEA2, viene dada por $\varepsilon_{2k}^{(n)}$, calculable a partir de la tabla ε

vectorial. Por otro, los dos algoritmos topológicos comparten la fórmula para $\varepsilon_{2k+1}^{(n)}$. En el caso de TEA1, la segunda ecuación de (3.7) proporciona una recurrencia de tres términos

$$\varepsilon_{2k+2}^{(n)} = a_{n,k}\varepsilon_{2k}^{(n)} + b_{n,k}\varepsilon_{2k}^{(n+1)},$$

con

$$a_{n,k} = \frac{-1}{\langle \Delta\varepsilon_{2k+1}^{(n)}, \Delta\varepsilon_{2k}^{(n)} \rangle}, \quad b_{n,k} = 1 - a_{n,k}.$$

En el caso de TEA2, se tiene

$$\varepsilon_{2k+2}^{(n)} = c_{n,k}\varepsilon_{2k}^{(n+1)} + d_{n,k}\varepsilon_{2k}^{(n+2)},$$

con

$$d_{n,k} = \frac{-1}{\langle \Delta\varepsilon_{2k+1}^{(n)}, \Delta\varepsilon_{2k}^{(n+1)} \rangle}, \quad c_{n,k} = 1 - d_{n,k}.$$

Otra propiedad relevante es que ambas versiones del método TEA admiten una representación de la forma (3.1). Brezinski, [12], derivó la fórmula a partir del teorema siguiente, debido a Macleod, [43, 29].

Teorema 3.3. *Supongamos que la sucesión $\{S_n\}_{n=0}^{\infty}$ satisface una relación de la forma*

$$\sum_{i=0}^k a_i(S_{n+i} - S) = 0, \quad (3.9)$$

para ciertos a_0, \dots, a_k (con $a_0 \neq 0, a_k \neq 0$) y un vector fijo S . Entonces, para $n \geq 0$,

$$\varepsilon_{2k}^{(n)} = \begin{cases} S & \text{si } \sum_{j=0}^k a_j \neq 0, \\ 0 & \text{si } \sum_{j=0}^k a_j = 0. \end{cases}$$

El razonamiento de Brezinski es como sigue: se parte de una sucesión modelo S_n verificando (3.9), donde se supone que $\sum_{i=0}^k a_i = 1$. De (3.9) se tiene

$$\sum_{j=0}^k a_j \Delta S_{n+j} = 0,$$

lo que lleva al sistema

$$\begin{cases} a_0 + a_1 + \dots + a_k = 1, \\ a_0 \langle y, \Delta S_n \rangle + a_1 \langle y, \Delta S_{n+1} \rangle + \dots + a_k \langle y, \Delta S_{n+k} \rangle = 0, \\ \vdots \\ a_0 \langle y, \Delta S_{n+k-1} \rangle + a_1 \langle y, \Delta S_{n+k} \rangle + \dots + a_k \langle y, \Delta S_{n+2k-1} \rangle = 0. \end{cases} \quad (3.10)$$

Asumiendo que (3.10) tiene solución única, a partir del teorema 3.3 podemos escribir

$$\varepsilon_{2k}^{(n)} = S = \sum_{i=0}^k a_i S_{n+i},$$

3 Extrapolación vectorial

donde las fórmulas de Cramer proporcionan la expresión teórica para los coeficientes

$$a_i = \frac{\Delta_{0i}}{\sum_{j=0}^k \Delta_{0j}}, \quad i = 0, \dots, k, \quad \Delta_{0j} = \begin{vmatrix} 1 & \dots & 1 & \dots & 1 \\ \langle y, \Delta S_n \rangle & \dots & 0 & \dots & \langle y, \Delta S_{n+k} \rangle \\ \vdots & & \vdots & & \vdots \\ \langle y, \Delta S_{n+k-1} \rangle & \dots & 0 & \dots & \langle y, \Delta S_{n+2k-1} \rangle \end{vmatrix}. \quad (3.11)$$

(La existencia de los a_i puede asegurarse a partir de alguna condición suficiente, como por ejemplo, cuando $\det \hat{W} \neq 0$, siendo

$$\hat{W} = \begin{pmatrix} \langle y, w_n \rangle & \langle y, w_{n+1} \rangle & \dots & \langle y, w_{n+k-1} \rangle \\ \vdots & \vdots & & \vdots \\ \langle y, w_{n+k-1} \rangle & \langle y, w_{n+k} \rangle & \dots & \langle y, w_{n+2k-2} \rangle \end{pmatrix},$$

con $w_m = \Delta u_m$, $u_m = \Delta S_m$, [59]. Esto puede también entenderse como un requerimiento añadido para el vector y .)

Utilizando este argumento, Brezinski demostró las siguientes representaciones:

Teorema 3.4 (Representación en determinantes del algoritmo TEA).

1. (TEA1.) Los elementos de (3.7) verifican

$$\varepsilon_{2k}^{(n)} = e_k(S_n), \quad \varepsilon_{2k+1}^{(n)} = (e_k(\Delta S_n))^{-1} = \frac{y}{\langle e_k(\Delta S_n), y \rangle},$$

donde

$$e_k(S_n) = \frac{F(S_n, S_{n+1}, \dots, S_{n+k})}{F(1, 1, \dots, 1)},$$

siendo

$$F(v_0, \dots, v_k) = \begin{vmatrix} v_0 & v_1 & \dots & v_k \\ \langle y, \Delta S_n \rangle & \langle y, \Delta S_{n+1} \rangle & \dots & \langle y, \Delta S_{n+k} \rangle \\ \vdots & \vdots & & \vdots \\ \langle y, \Delta S_{n+k-1} \rangle & \langle y, \Delta S_{n+k} \rangle & \dots & \langle y, \Delta S_{n+2k-1} \rangle \end{vmatrix}. \quad (3.12)$$

En (3.12), si v_0, \dots, v_k son escalares, F es el determinante correspondiente; si v_0, \dots, v_k son vectores, (3.12) es un vector definido a partir de su desarrollo con respecto a la primera fila, es decir

$$F(v_0, \dots, v_k) = \sum_{i=0}^k a_i v_i,$$

con los a_i dados por (3.11).

2. (TEA2.) Los elementos (3.8) verifican

$$\tilde{\varepsilon}_{2k}^{(n)} = \tilde{e}_k(S_n), \quad \tilde{\varepsilon}_{2k+1}^{(n)} = (\tilde{e}_k(\Delta S_n))^{-1} = \frac{y}{\langle y, \tilde{e}_k(\Delta S_n) \rangle},$$

donde

$$\tilde{e}_k(S_n) = \frac{F(S_{n+k}, S_{n+k+1}, \dots, S_{n+2k})}{F(1, 1, \dots, 1)},$$

y F dado por (3.12)².

Podemos asimismo analizar varios aspectos de la implementación del TEA. No es difícil observar que tanto el coste operativo como de almacenamiento es similar a los del VEA. Hay, sin embargo, una implementación más económica para el TEA (llamada ETEA), que calcula $\varepsilon_{2k}^{(n)}$ sin usar la computación de $\varepsilon_r^{(n+j)}$, $1 \leq j+r \leq 2k$ y que se basa en las representaciones del teorema 3.4. El procedimiento se utilizará también en los métodos polinómicos, pues en ambas situaciones se comparte la misma idea de formular la extrapolación en la forma (3.1). Para el caso del TEA, la implementación es como sigue, [59]: del teorema 3.4 se tiene

$$\varepsilon_{2k}^{(n)} = \sum_{i=0}^k a_i S_{n+i} \quad (\text{TEA1}), \quad \varepsilon_{2k}^{(n)} = \sum_{i=0}^k a_i S_{n+k+i} \quad (\text{TEA2}),$$

con los a_i verificando (3.11). Los pasos de la implementación son:

1. Elegimos enteros $n \geq 0$, $k \geq 1$, vectores S_0, y .
2. Calculamos S_1, S_2, \dots, S_n .
3. Para $i = 0, 1, \dots, 2k-1$, calculamos S_{n+i+1} , $\Delta S_{n+i} = S_{n+i+1} - S_{n+i}$ y $\alpha_i = \langle y, \Delta S_{n+i} \rangle$.

Una vez calculados estos valores, podemos ahorrar en el coste de almacenamiento del siguiente modo:

- En el caso TEA1, guardamos $S_n, \Delta S_{n+i}$, $i = 0, \dots, k-1$, descartando el resto de los $S_{n+j}, \Delta S_{n+j}$.
- En el caso TEA2, guardamos S_{n+k} y ΔS_{n+k+i} , $i = 0, \dots, k-1$, descartando el resto de los $S_{n+j}, \Delta S_{n+j}$.

4. Fijando $\delta_{i,j} = \alpha_{i+j}$, se resuelve el sistema (3.10), a partir de escribir

$$a_i = \frac{c_i}{\sum_{j=0}^k c_j}, \quad i = 0, \dots, k,$$

donde $c_k = 1$ y c_0, \dots, c_{k-1} es la solución de

$$\sum_{j=0}^{k-1} \delta_{i,j} c_j = -\delta_{i,k}, \quad i = 0, \dots, k-1. \quad (3.13)$$

(Se asume que $\sum_{j=0}^k c_j \neq 0$.)

5. Con los a_i disponibles, calculamos ξ_0, \dots, ξ_{k-1} dados por

$$\xi_j = \sum_{i=j+1}^k a_i = 1 - \sum_{i=0}^j a_i, \quad (3.14)$$

es decir

$$\xi_{k-1} = a_k, \quad \xi_j = \xi_{j+1} + a_{j+1}, \quad j = k-2, \dots, 0,$$

²Al parecer, una de las motivaciones de Brezinski para la derivación del TEA procedía de la falta de representación determinantal del VEA. Recientemente, [59], Sidi obtuvo una formulación para el algoritmo ε vectorial, parecida a la dada en el teorema 3.4 para el TEA, utilizando aproximantes de Padé vectoriales.

3 Extrapolación vectorial

(con $\xi_0 = 1 - a_0$).

6. Reescribimos

$$\varepsilon_{2k}^{(n)} = \sum_{i=0}^k a_i S_{n+i} = S_n + \sum_{j=0}^{k-1} \xi_j \Delta S_{n+j},$$

en el caso de TEA1 y

$$\varepsilon_{2k}^{(n)} = \sum_{i=0}^k a_i S_{n+k+i} = S_{n+k} + \sum_{j=0}^{k-1} \xi_j \Delta S_{n+k+j},$$

en el caso de TEA2.

La economización proporcionada por los pasos 1-6 se refleja en $5k$ operaciones vectoriales y un sistema lineal (3.13) que puede resolverse con $O(k^2)$ operaciones escalares. Asimismo, necesitamos $N(k+1)$ posiciones de memoria (para $S_n, \Delta S_{n+i}, 0 \leq i \leq k-1$, en el caso de TEA1, y para $S_{n+k}, \Delta S_{n+k+i}, 0 \leq i \leq k-1$, en el caso de TEA2) en todo el proceso de cálculo de $\varepsilon_{2k}^{(n)}$. Otras opciones para reducir el esfuerzo computacional en el algoritmo topológico pueden consultarse, por ejemplo, en [17].

3.2.2. Métodos polinómicos

La construcción del segundo grupo de métodos de extrapolación vectorial suele motivarse a partir de la búsqueda de la solución de un sistema lineal

$$S = TS + d, \quad (3.15)$$

donde $T \in \mathbb{R}^{N \times N}$, $d \in \mathbb{R}^N$, con $I - T$ invertible. En este caso, para un vector inicial S_0 , la sucesión

$$S_{m+1} = TS_m + d, \quad m = 0, 1, \dots \quad (3.16)$$

converge hacia S si y sólo si el radio espectral de T es menor que uno. La primera observación es que S puede alcanzarse exactamente a partir de la sucesión (3.16).

Teorema 3.5. *Sea*

$$p_n(z) = \sum_{i=0}^{k_n} c_i z^i, \quad c_{k_n} = 1, \quad (3.17)$$

el polinomio mínimo de T con respecto a $e_n = S_n - S$. Entonces $\sum_{i=0}^{k_n} c_i \neq 0$ y además

$$S = \frac{\sum_{i=0}^{k_n} c_i S_{n+i}}{\sum_{i=0}^{k_n} c_i}. \quad (3.18)$$

Demostración:

Como $I - T$ es invertible, $p_n(1) = \sum_{i=0}^{k_n} c_i \neq 0$. Por otro lado, a partir de (3.15) y (3.16), y como p_n es el polinomio mínimo de T con respecto a e_n , se tiene

$$0 = p_n(T)e_n = \sum_{i=0}^{k_n} c_i T^i e_n = \sum_{i=0}^{k_n} c_i e_{n+i} = \sum_{i=0}^{k_n} c_i S_{n+i} - S \sum_{i=0}^{k_n} c_i$$

verificando así (3.17). □

El teorema 3.5 afirma que para calcular S a partir de (3.16), sería necesario obtener el polinomio mínimo de T con respecto a e_n , lo que en principio no es posible al no conocerse e_n . El siguiente resultado da una alternativa.

Teorema 3.6. *El polinomio mínimo de T con respecto a e_n coincide con el polinomio mínimo de T con respecto a $u_n = \Delta S_n$.*

Demostración

Denotemos por $s_n(z)$ el polinomio mínimo de T con respecto a u_n . Observemos que

$$(T - I)e_n = u_n.$$

Entonces $p_n(T)u_n = 0$ y, por tanto, $s_n(z)$ divide a $p_n(z)$. De forma similar,

$$0 = s_n(T)u_n = (T - I)s_n(T)e_n,$$

y como $(I - T)$ es invertible, entonces $s_n(T)e_n = 0$ y, por tanto, $p_n(z)$ divide a $s_n(z)$. \square

De este modo, el polinomio $p_n(z)$ definido en (3.17) puede determinarse a partir de la condición

$$p_n(T)u_n = 0, \tag{3.19}$$

donde los vectores u_n sí pueden calcularse.

El sistema (3.19) puede escribirse

$$U_{k_n} c = 0, \tag{3.20}$$

o bien

$$U_{k_n-1} c' = -u_{n+k_n}, \tag{3.21}$$

donde

$$U_j = [u_n | u_{n+1} | \dots | u_{n+j}] \tag{3.22}$$

es la matriz $N \times (j + 1)$ cuyas columnas son los vectores $u_n, u_{n+1}, \dots, u_{n+j}$, $c = (c_0, \dots, c_{k_n})^T$ y $c' = (c_0, \dots, c_{k_n-1})^T$. Por la existencia y unicidad del polinomio mínimo, cf. teorema 3.2, el sistema (3.21) (que hace uso de la propiedad $c_{k_n} = 1$) es consistente, aunque sobredeterminado en general. Para su resolución, son necesarios los $k_n + 2$ vectores S_n, \dots, S_{n+k_n+1} . Alternativamente, la expresión (3.18) puede también escribirse

$$S = \sum_{i=0}^{k_n} \gamma_i S_{n+i}, \quad \gamma_i = \frac{c_i}{\sum_{j=0}^{k_n} c_j}, \tag{3.23}$$

de manera que $\sum_{i=0}^{k_n} \gamma_i = 1$. Usando (3.20) y (3.23) resulta que $\gamma = (\gamma_0, \dots, \gamma_{k_n})^T$ satisface

$$U_{k_n} \gamma = 0, \quad \sum_{i=0}^{k_n} \gamma_i = 1,$$

que es un sistema de $N + 1$ ecuaciones para las $k_n + 1$ incógnitas γ_i , de solución única. Observemos que (3.23) tiene sentido ya sea S el límite de S_n (si T tiene radio espectral $\rho(T) < 1$) o el antilímite (si $\rho(T) > 1$).

El procedimiento anterior para calcular S tiene varios inconvenientes que lo hacen inviable en la práctica. El primero es que el polinomio mínimo de T con respecto a u_n (y, por tanto,

su grado k_n) depende de n . Por otra parte, este grado k_n puede ser tan grande como N , lo que puede llevar a un coste computacional excesivo.

Sin embargo, podemos utilizar los argumentos anteriores para obtener aproximaciones a S , sustituyendo el polinomio mínimo de T con respecto a u_n por otro, de grado fijo k y mucho menor, para reducir el coste operativo. Observemos que si $k_n \leq N$ es el grado del polinomio mínimo de T con respecto a u_n , es claro, por la definición, que el sistema de vectores $\{u_n, u_{n+1}, \dots, u_{n+k}\}$ es libre si $0 \leq k \leq k_n - 1$, pero $\{u_n, u_{n+1}, \dots, u_{n+k_n}\}$ no lo es. Esto implica que las matrices U_j dadas por (3.22) para $j = 0, \dots, k_n - 1$, tienen rango completo $j + 1$, pero el rango de U_{k_n} es k_n . De este modo, la resolución de sistemas del tipo (3.21)

$$U_{k-1}c' = -u_{n+k} \quad (3.24)$$

con k mucho menor que k_n , debe entenderse en el sentido de mínimos cuadrados; igualmente, la expresión

$$S_{n,k} = \sum_{i=0}^k \gamma_i S_{n+i}, \quad \gamma_i = \frac{c_i}{\sum_{j=0}^k c_j}, \quad (3.25)$$

será una aproximación a S , definiendo así un método de extrapolación, llamado método MPE (Minimal Polynomial Extrapolation). La construcción sigue entonces los pasos, [59, 60]:

1. Elegimos k pequeño y n , calculando los términos $S_n, S_{n+1}, \dots, S_{n+k+1}$ de la sucesión original.
2. Calculamos los vectores correspondientes u_n, \dots, u_{n+k} y formamos la matriz $U_{k-1} \in \mathbb{R}^{N \times k}$.
3. Resolvemos (3.24) en el sentido de mínimos cuadrados; es decir, calculamos

$$\min_{c_0, \dots, c_{k-1}} \left\| \sum_{i=0}^{k-1} c_i u_{n+i} + u_{n+k} \right\|, \quad (3.26)$$

siendo $\|\cdot\|$ la norma euclídea en \mathbb{R}^N .

4. Fijamos $c_k = 1$ y calculamos $S_{n,k}^{MPE} = S_{n,k}$ usando (3.25).

El segundo método polinómico es una variante del método MPE conocido como método RRE (Reduced Rank Extrapolation). Está basado en sustituir (3.26) por un problema de minimización similar, cf. (3.20) y (3.21):

1. Elegimos k y n , calculando $S_n, S_{n+1}, \dots, S_{n+k+1}$.
2. Formamos la matriz $U_k = [u_n | u_{n+1} | \dots | u_{n+k}] \in \mathbb{R}^{N \times (k+1)}$.
3. Resolvemos el sistema sobredeterminado

$$U_k \gamma = 0,$$

en el sentido de mínimos cuadrados y con la restricción $\sum_{i=0}^k \gamma_i = 1$. Esto significa resolver el problema de minimización con una restricción

$$\min_{\gamma_0, \dots, \gamma_k} \left\| \sum_{i=0}^k \gamma_i u_{n+i} \right\|, \quad \text{s.a.} \quad \sum_{i=0}^k \gamma_i = 1. \quad (3.27)$$

4. Calculamos $S_{n,k}^{RRE} = S_{n,k}$ usando la fórmula (3.25) con los γ_i calculados en el paso 3.

El problema (3.27) puede sustituirse por un problema de minimización equivalente pero sin restricciones, lo que es más conveniente en términos computacionales. Para ello, observemos que, si $u_m = \Delta S_m$ y $w_m = \Delta u_m$, entonces se tiene

$$S_{n+i} = S_n + \sum_{j=0}^{i-1} u_{n+j}, \quad u_{n+i} = u_n + \sum_{j=0}^{i-1} w_{n+j}. \quad (3.28)$$

Utilizando la primera relación de (3.28) y la restricción $\sum_{i=0}^k \gamma_i = 1$, no es difícil ver que

$$\sum_{i=0}^k \gamma_i S_{n+i} = S_n + \sum_{j=0}^{k-1} \xi_j u_{n+j}, \quad (3.29)$$

donde

$$\xi_j = \sum_{i=j+1}^k \gamma_i = 1 - \sum_{i=0}^j \gamma_i, \quad j = 0, \dots, k-1, \quad (3.30)$$

(cf. (3.14)). Igualmente, utilizando la segunda relación de (3.28)

$$\sum_{i=0}^k \gamma_i u_{n+i} = u_n + \sum_{j=0}^{k-1} \xi_j w_{n+j}. \quad (3.31)$$

Podemos entonces, usando (3.31), sustituir (3.27) por

$$\min_{\xi_0, \dots, \xi_{k-1}} \left\| u_n + \sum_{j=0}^{k-1} \xi_j w_{n+j} \right\| = \min_{\xi} \| W_{k-1} \xi + u_n \|, \quad (3.32)$$

con $\xi = (\xi_0, \dots, \xi_{k-1})^T$, $W_{k-1} = [w_n \dots w_{n+k-1}]$. Una vez determinado ξ , podemos avanzar en el método RRE calculando $S_{n,k}$, via (3.29), con los coeficientes ξ_j directamente, o a partir de (3.25) resolviendo el sistema (3.30) para los γ_i :

$$\gamma_0 = 1 - \xi_0, \quad \gamma_i = \xi_{i-1} - \xi_i, \quad i = 1, \dots, k-1, \quad \gamma_k = \xi_{k-1}.$$

La equivalencia de los procedimientos fue demostrada en [60].

Es importante observar que la introducción de un parámetro k , pequeño y arbitrario, para la construcción, hace que los métodos MPE y RRE sean aplicables para la extrapolación de cualquier sucesión $\{S_n\}_{n=0}^{\infty}$ con independencia de cómo se genera ésta. En el caso lineal (3.16), es destacable, [55, 59], la equivalencia matemática de los métodos MPE, RRE y TEA1 con tres métodos de Krylov (Arnoldi, GMRES y Lanczos, respectivamente). La literatura sobre los métodos de Krylov es abundante, véase, por ejemplo [51, 69]. Asimismo, el método MPE admite otras dos variantes, los llamados métodos MMPE (Modified MPE) y SVD-MPE. En el primero se requiere que el vector $U_{k-1}c' + u_{n+k}$, a minimizar en (3.26), sea ortogonal a un subespacio fijo de dimensión k (lo que permite una resolución alternativa para el cálculo de c'); en el segundo, se minimiza $\|U_k c\|$ para $c = (c_0, \dots, c_k)^T$ con $\|c\| = 1$. La solución c resulta ser el vector singular por la derecha correspondiente al menor valor singular de U_k . Véase [59] y referencias para más detalles.

En analogía con lo descrito para los algoritmos vectoriales de tipo ε , podemos analizar las siguientes propiedades de los métodos polinómicos. En lo que sigue, $S_{n,k}$ denotará $S_{n,k}^{MPE}$ ó $S_{n,k}^{RRE}$, distinguiéndose cuando sea necesario.

1. Núcleo de la transformación (cf. teorema 3.3). Supongamos que la sucesión $\{S_n\}_{n=0}^{\infty}$ satisface una relación de la forma

$$\sum_{i=0}^k a_i (S_{m+i} - S) = 0, \quad m = 0, 1, \dots, \quad (3.33)$$

para cierto vector S , coeficientes a_i con $a_0, a_k \neq 0$, $\sum_{i=0}^k a_i \neq 0$ y siendo k mínimo con esta propiedad. Entonces, $S_{n,k} = S$ para todo n , [59]. (Observemos que en el caso particular de la sucesión (3.16) se satisface la relación anterior con k el grado del polinomio mínimo y el resultado anterior fue ya demostrado para este caso en el teorema 3.5. La condición (3.33) puede entonces considerarse una generalización de dicho teorema.)

2. Representación con determinantes. Los métodos polinómicos admiten también una representación en forma determinantal. Para su derivación se utiliza el siguiente lema previo, [59].

Lema 3.1. Sean $u_{i,j}, \gamma_j$ escalares verificando

$$\sum_{j=0}^k u_{i,j} \gamma_j = 0, \quad i = 0, \dots, k-1, \quad \sum_{j=0}^k \gamma_j = 1. \quad (3.34)$$

Entonces, ya sean v_0, \dots, v_k escalares o vectores,

$$\sum_{i=0}^k \gamma_i v_i = \frac{D(v_0, \dots, v_k)}{D(1, \dots, 1)},$$

donde

$$D(v_0, \dots, v_k) = \begin{vmatrix} v_0 & v_1 & \dots & v_k \\ u_{0,0} & u_{0,1} & \dots & u_{0,k} \\ \vdots & \vdots & \dots & \vdots \\ u_{k-1,0} & u_{k-1,1} & \dots & u_{k-1,k} \end{vmatrix}.$$

Si v_0, \dots, v_k son vectores, la notación del determinante se entiende como la combinación vectorial obtenida al desarrollar por la primera fila, cf. teorema 3.4.

Teorema 3.7. Para los métodos MPE y RRE se tienen las representaciones

$$S_{n,k} = \frac{D(S_n, S_{n+1}, \dots, S_{n+k})}{D(1, 1, \dots, 1)},$$

donde $D(v_0, \dots, v_k)$ es el determinante definido en el lema 3.1 con

$$u_{i,j} = \begin{cases} \langle u_{n+i}, u_{n+j} \rangle & \text{(MPE)} \\ \langle w_{n+i}, u_{n+j} \rangle & \text{(RRE)} \end{cases} \quad (3.35)$$

y siendo $u_m = \Delta S_m, w_m = \Delta u_m$.

Demostración:

Buscamos aplicar el lema 3.1 con los valores u_{ij} especificados en (3.35). Observemos que, por construcción, los coeficientes γ_j satisfacen ya, en ambos casos, la segunda condición de (3.34), por lo que bastará con demostrar la primera.

En el caso del método MPE, tenemos que los coeficientes c_j verifican las ecuaciones normales correspondientes al problema de minimización

$$\sum_{j=0}^{k-1} \underbrace{\langle u_{n+i}, u_{n+j} \rangle}_{u_{ij}} c_j = - \underbrace{\langle u_{n+i}, u_{n+k} \rangle}_{u_{ik}}, \quad i = 0, \dots, k-1.$$

Tomando $c_k = 1$ y dividiendo por $\sum_{j=0}^k c_j$, las ecuaciones anteriores pasan a tener la forma de (3.34). Esto demuestra el resultado para el método MPE. En el caso del método RRE, las ecuaciones normales del problema de minimización pueden escribirse, cf. (3.32),

$$\sum_{j=0}^{k-1} \langle w_{n+i}, w_{n+j} \rangle \xi_j = - \langle w_{n+i}, u_n \rangle, \quad i = 0, \dots, k-1,$$

o, equivalentemente

$$\begin{aligned} 0 &= \sum_{j=0}^{k-1} \langle w_{n+i}, w_{n+j} \rangle \xi_j + \langle w_{n+i}, u_n \rangle = \langle w_{n+i}, u_n + \sum_{j=0}^{k-1} w_{n+j} \xi_j \rangle = \\ &= \langle w_{n+i}, \sum_{j=0}^k \gamma_j u_{n+j} \rangle = \sum_{j=0}^k \underbrace{\langle w_{n+i}, u_{n+j} \rangle}_{u_{ij}} \gamma_j. \quad \square \end{aligned}$$

3. Implementación.

Puede darse una implementación eficiente y unificada para ambos métodos MPE y RRE, utilizando la resolución de los correspondientes problemas de minimización mediante la factorización QR. En este sentido, hacemos uso de los siguientes resultados, véanse e.g. [28, 59].

Teorema 3.8. Sea $A \in \mathbb{R}^{m \times s}$ con $m \geq s = \text{rango}(A)$, $A = QR$ la factorización QR de A . Entonces:

(a) El problema de minimización

$$\min_{x \in \mathbb{R}^s} \|Ax - b\|, \quad b \in \mathbb{R}^m,$$

tiene solución única \hat{x} , dada por la solución del sistema

$$Rx = Q^T b.$$

(b) Si $g \in \mathbb{R}^s$, el problema de minimización

$$\min_{x \in \mathbb{R}^s} \|Ax\|, \quad \text{s.a. } g^T x = 1$$

tiene solución única, dada por

$$\hat{x} = \lambda(R^T R)^{-1}g, \quad \lambda = \frac{1}{\langle g, (R^T R)^{-1}g \rangle}.$$

Además

$$\|Ax\| = \sqrt{\lambda}.$$

Los pasos de la implementación son los siguientes:

- (i) Elegimos $n \geq 0, k \geq 1, S_0$ y generamos S_1, \dots, S_n . A partir de este último, calculamos $S_{n+1}, \dots, S_{n+k+1}$.
- (ii) Construimos las diferencias $u_{n+j} = \Delta S_{n+j}, j = 0, \dots, k$ y formamos la matriz $U_k = [u_n | u_{n+1} | \dots | u_{n+k}]$.

Realizamos la factorización QR de U_k por recursividad: si $U_0 = u_n$, es claro que $U_0 = Q_0 R_0$ con $Q_0 = \frac{u_n}{\|u_n\|}$ y $R_0 = \|u_n\|$. Si denotamos por

$$U_{j-1} = [u_n | \dots | u_{n+j-1}],$$

y suponemos conocida la factorización QR de U_{j-1}

$$U_{j-1} = Q_{j-1} R_{j-1},$$

entonces no es difícil ver que la factorización QR de $U_j = Q_j R_j$ se obtiene como

$$Q_j = [Q_{j-1} | q_j], \quad R_j = \begin{pmatrix} R_{j-1} & r_j \\ 0 & r_{j,j} \end{pmatrix},$$

donde

$$r_j = Q_{j-1}^T u_{n+j}, \quad r_{j,j} = \|u_{n+j} - Q_{j-1} r_j\|, \quad q_j = \frac{1}{r_{j,j}} (u_{n+j} - Q_{j-1} r_j). \quad (3.36)$$

El proceso se realiza sin coste operativo adicional. Al final del mismo, Q_k se almacena en U_k .

- (iii) Cálculo de $\gamma = (\gamma_0, \dots, \gamma_k)^T$. En el caso del método MPE, el sistema (3.24), utilizando el paso (ii) y el teorema 3.8.(a), se escribe

$$R_{k-1} c' = -Q_{k-1}^T u_{n+k} = -r_k, \quad (3.37)$$

(donde r_k procede de la primera fórmula de (3.36) con $j = k$). Con c' disponible, como solución del sistema triangular superior (3.37), fijamos $c_k = 1$ y calculamos

$$\gamma_i = \frac{c_i}{(\sum_{j=0}^k c_j)}, \quad i = 0, \dots, k.$$

En el caso del método RRE, utilizando (3.27) y el teorema 3.8.(b), se tiene que

$\gamma = \lambda h$, donde h es solución de

$$R_k^T R_k h = e = (1, 1, \dots, 1)^T, \quad (3.38)$$

y $\lambda = \frac{1}{e^T h}$. (Observamos que el sistema (3.38) puede resolverse a partir de dos sistemas triangulares, con matrices R_k^T y R_k , respectivamente.)

(iv) Conocido γ , calculamos $\xi = (\xi_0, \dots, \xi_{k-1})^T$ utilizando (3.30).

(v) Calculamos $S_{n,k}$ para ambos métodos, a partir de (3.29),

$$S_{n,k} = S_n + \sum_{j=0}^{k-1} \xi_j u_{n+j},$$

que, una vez computados los u_{n+j} , no requiere el almacenamiento de S_{n+i} , $i = 1, \dots, k+1$.

Para completar la implementación, indicamos cómo introducir el modo cíclico y una estimación de los errores para controlar el proceso.

En relación con el segundo punto, es habitual utilizar el valor relativo $\|r(S_{n,k})\|/\|r(S_0)\|$, siendo $r(x)$ el residuo, en x , de la fórmula para generar la sucesión S_m . Notemos que en el caso lineal (3.16)

$$\begin{aligned} r(S_{n,k}) &= TS_{n,k} + d - S_{n,k} = \sum_{i=0}^k \gamma_i (TS_{n+i} + d) - \sum_{i=0}^k \gamma_i S_{n+i} \\ &= \sum_{i=0}^k \gamma_i (S_{n+i+1} - S_{n+i}) = \sum_{i=0}^k \gamma_i u_{n+i} = U_k \gamma. \end{aligned}$$

Por otro lado, como $u_m = (T - I)e_m$, entonces

$$r(S_{n,k}) = \sum_{i=0}^k \gamma_i (T - I)e_{n+i} = (T - I) \sum_{i=0}^k \gamma_i e_{n+i} = (T - I)(S_{n,k} - S).$$

Por tanto

$$\|S_{n,k} - S\| \leq \|(T - I)^{-1}\| \|r(S_{n,k})\| = \|(T - I)^{-1}\| \|U_k \gamma\|.$$

De este modo, es razonable utilizar $\|U_k \gamma\|$ como estimación del error de extrapolación, tanto para el caso lineal como si la generación de S_m es no lineal. En esta última situación, la justificación viene dada a partir de los argumentos anteriores y usando linealización.

La principal ventaja de elegir $\|U_k \gamma\|$ para estimar los errores es que puede calcularse sin coste adicional, [59]:

Teorema 3.9. Con la notación de los pasos (i)-(v) de la implementación, se tiene que

$$\|U_k \gamma\| = \begin{cases} |\gamma_k| r_{k,k} & (MPE) \\ \sqrt{\lambda} & (RRE) \end{cases}.$$

La estimación del error debe incorporarse en el modo cíclico de implementación, que

en su versión completa tendría la forma siguiente:

(M1) Pasos (i)-(v) con $n = 0$ para calcular $S_{0,k}$ y cómputo de $\|U_k \gamma\|$ según el teorema 3.9.

(M2) Mientras $\|U_k \gamma\|$ sea mayor que una tolerancia prefijada, hacemos $S_0 = S_{0,k}$ y volvemos al paso (M1).

Observemos que, en la formulación general del modo cíclico, descrita en la sección 3.1, $l = k$ en el caso de los métodos polinómicos (es decir, necesitamos almacenar $k + 1$ términos S_1, \dots, S_{k+1}) frente a los métodos vectoriales de tipo ε , para los que $l = 2k - 1$.

Por otra parte, tal y como se menciona en [59], si bien los métodos MPE y RRE son diferentes, pueden establecerse algunos resultados que justifican un comportamiento general común en los experimentos, tanto en sentido positivo de aceleración de convergencia, como negativo, cuando ambos procesos no funcionan.

3.3. Convergencia y estabilidad de los métodos de extrapolación vectorial

Dedicamos esta sección a presentar algunos resultados de convergencia y estabilidad para los dos tipos de métodos de extrapolación vectorial analizados en la sección 3.2. Las demostraciones de muchos de los teoremas adoptan argumentos del caso escalar. Por sencillez, nuestra exposición aquí será descriptiva, resaltando la naturaleza de los resultados y su aplicación en la práctica. El lector puede consultar las demostraciones en las referencias citadas a lo largo de la sección.

Los principales resultados suelen ser de dos tipos, para procesos lineales:

- Convergencia de $\{S_{n,k}\}_{n=0}^{\infty}$ con k fijo.
- Comportamiento del error $S_{n,k} - S$, con n y k fijos.

Los del segundo tipo se aplican asimismo al caso no lineal, vía linealización, cuando el método se implementa en modo cíclico.

3.3.1. Métodos vectoriales de tipo ε .

Los resultados de convergencia para los algoritmos vectoriales de tipo ε están basados en el teorema 3.3 de caracterización del núcleo, que asegura la exactitud de la extrapolación $\{\varepsilon_{2k}^{(n)}\}_{n=0}^{\infty}$ cuando la sucesión está generada por un proceso lineal (3.16) y siendo k el grado del polinomio mínimo de T con respecto a $u_0 = \Delta S_0$. Por otro lado, Brezinski, [12, 13], utiliza la generalización de la transformación de Shanks al caso vectorial para obtener resultados de convergencia de $\{\varepsilon_{2k}^{(n)}\}_{n=0}^{\infty}$, con k fijo, similares a los del algoritmo general GEA escalar (cf. sección 2.2). Otros resultados de convergencia, que hacen uso de aproximantes de Padé vectoriales, pueden consultarse en [29, 30].

En relación con el comportamiento de $\varepsilon_{2k}^{(n)}$ con k y n fijos, los métodos alcanzan convergencia cuadrática en el siguiente sentido:

3.3 Convergencia y estabilidad de los métodos de extrapolación vectorial

Teorema 3.10. Supongamos que S es un punto fijo de una aplicación $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$, diferenciable en un entorno de S y con $I - f'(S)$ invertible. Supongamos que $\{S_n\}_{n=0}^{\infty}$ se genera por iteración de punto fijo con f :

$$S_{m+1} = f(S_m), \quad m = 0, 1, \dots,$$

y sea $\{X_n\}_{n=0}^{\infty}$ la sucesión obtenida al aplicar el método VEA o el método TEA en modo cíclico, con $l = 2k_n - 1$, siendo k_n el grado del polinomio mínimo de $f'(S)$ con respecto a $S_n - S$. Entonces, existe un entorno V de S tal que para todo $X_0 \in V$ se tiene que

$$\|X_{n+1} - S\| = \mathcal{O}(\|X_n - S\|^2), \quad n = 0, 1, \dots$$

Nótese que el resultado incluye el caso lineal (3.16), si $f(x) = Tx + d$. Las correspondientes demostraciones pueden consultarse en [27, 39, 13, 60].

3.3.2. Métodos polinómicos.

En el caso de los métodos MPE y RRE, el principal resultado de convergencia de $\{S_{n,k}\}_{n=0}^{\infty}$ con k fijo se refiere a sucesiones de la forma

$$S_m = S + \sum_{i=1}^p \lambda_i^m v_i, \quad p \leq N, \quad (3.39)$$

donde los vectores v_i son linealmente independientes, los escalares λ_i son distintos entre sí, con $\lambda_i \neq 0, 1$ y

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_p|. \quad (3.40)$$

Observemos que para sucesiones verificando (3.16), se cumple (3.39) cuando la matriz T es diagonalizable y $I - T$ invertible. En ese caso, los escalares λ_i son algunos o todos los autovalores no nulos de T y los v_i son autovectores asociados.

Teorema 3.11 ([59]). Supongamos que $\{S_n\}_{n=0}^{\infty}$ satisface (3.39)-(3.40) y que existe k tal que $|\lambda_k| > |\lambda_{k+1}|$. Entonces:

(i) $S_{n,k}^{\text{RRE}}$ existe para todo n y $S_{n,k}^{\text{MPE}}$ existe para n suficientemente grande.

(ii) Para ambos métodos, se tiene

$$S_{n,k} - S = (C_{n,k} + \mathcal{O}(1))\lambda_{k+1}^n = \mathcal{O}(\lambda_{k+1}^n), \quad n \rightarrow \infty$$

con los vectores $C_{n,k}$ iguales para los dos métodos y tales que

$$\sup_n \|C_{n,k}\| < \infty.$$

(iii) Si $S_{n,k} = \sum_{i=0}^k \gamma_i S_{n+i}$, $\gamma_i = \gamma_i^{(n,k)}$, entonces existe $\lim_{n \rightarrow \infty} \gamma_i^{(n,k)}$ y además

$$\lim_{n \rightarrow \infty} \sum_{i=0}^k \gamma_i^{(n,k)} \lambda^i = \prod_{i=1}^k \frac{\lambda - \lambda_i}{1 - \lambda_i}.$$

3 Extrapolación vectorial

En consecuencia, $\Gamma_{n,k} = \sum_{i=0}^k |\gamma_i^{(n,k)}|$ satisface

$$\lim_{n \rightarrow \infty} \Gamma_{n,k} = \sum_{i=0}^k \lim_{n \rightarrow \infty} |\gamma_i^{(n,k)}| \leq \prod_{i=1}^k \frac{\lambda + |\lambda_i|}{|1 - \lambda_i|},$$

y ambos métodos son estables.

(iv) Para $k = p$ se tiene que

$$S_{n,p} = S; \quad \sum_{i=0}^p \gamma_i^{(n,p)} \lambda^i = \prod_{i=1}^p \frac{\lambda - \lambda_i}{1 - \lambda_i}.$$

Comentarios al teorema 3.11

1. La demostración del teorema 3.11 puede consultarse en [59]. La prueba hace uso de la representación con determinantes de los métodos y adapta al caso vectorial muchos argumentos y técnicas usadas para la transformación de Shanks, véase sección 2.1.
2. El resultado de existencia para $S_{n,k}^{RRE}$ de (i), así como la aceleración de convergencia dado por (ii) son también válidos para el método TEA y para otras variantes del método MPE. Por otro lado, el hecho de que los vectores $C_{n,k}$ en (ii) sean los mismos para ambos métodos implica que los errores $S_{n,k} - S$ son asintóticamente iguales, en el sentido de que

$$S_{n,k}^{MPE} - S \sim S_{n,k}^{RRE} - S,$$

componente a componente.

3. Por (3.39), es $|\lambda_{k+1}| < |\lambda_k|$. Entonces si $|\lambda_1| < 1$, (es decir, si S_m converge a S) se tiene por (ii) que $S_{n,k}$ converge a S cuando $n \rightarrow \infty$ más rápidamente que S_m . Dicha convergencia se mantiene en el caso de que $|\lambda_{k+1}| < 1 \leq |\lambda_1|$, es decir, aunque S_m no converja hacia S . En particular, la aceleración de la convergencia dada por $S_{n,k}$ relativa a S_n , se comporta como

$$\frac{S_{n,k} - S}{S_n - S} = \mathcal{O}\left(\left|\frac{\lambda_{k+1}}{\lambda_1}\right|^n\right).$$

4. Los coeficientes $\gamma_i^{(n,k)}$ en (iii) contienen información relevante sobre los λ_i , que no proporciona, en general, la sucesión S_m . Para ello, se tiene en cuenta la identidad, [59],

$$Q_{n,k}(\lambda) = \sum_{i=0}^k \gamma_i^{(n,k)} \lambda^i = \prod_{i=1}^k \frac{\lambda - \lambda_i}{1 - \lambda_i} + \mathcal{O}\left(\left|\frac{\lambda_{k+1}}{\lambda_1}\right|^n\right), \quad n \rightarrow \infty.$$

5. Algunos resultados del teorema 3.11 pueden extenderse para analizar $S_{n,k}$ con $|\lambda_k| = |\lambda_{k+1}|$. Se refieren a una sucesión S_m generada como en (3.16), con T diagonalizable, $I - T$ no singular. Específicamente, se cumple la existencia incondicional de $S_{n,k}^{RRE}$, mientras que la de $S_{n,k}^{MPE}$ está condicionada a que la parte hermítica de $E = a(I - T)$ (es decir, $E_H = \frac{1}{2}(E + E^*)$, siendo E^* la traspuesta conjugada de E) para algún $a \in \mathbb{C}$ con $|a| = 1$, sea definida positiva. En caso de existencia, para ambos métodos se cumple

$$S_{n,k} - S = \mathcal{O}(\lambda_{k+1}^n), \quad n \rightarrow \infty.$$

La implicación general es que si

$$|\lambda_t| > |\lambda_{t+1}| = \dots = |\lambda_{t+r}| > |\lambda_{t+r+1}|,$$

entonces

$$S_{n,k} - S = \mathcal{O}(\lambda_{t+i}^n), \quad n \rightarrow \infty,$$

para $k = t, t+1, \dots, t+r-1$. Así, por ejemplo, si $|\lambda_1| = |\lambda_2| = \dots = |\lambda_r| > |\lambda_{r+1}|$,

$$S_{n,k} - S = \mathcal{O}(\lambda_1^n), \quad k = 0, \dots, r-1, \quad S_{n,r} - S = \mathcal{O}(\lambda_{r+1}^n),$$

mientras que si $|\lambda_1| = |\lambda_2| = \dots = |\lambda_p|$, entonces $S_{n,k} - S = \mathcal{O}(\lambda_1^n)$ para todo k y no hay aceleración de convergencia.

6. Los resultados del teorema 3.11 y del comentario anterior pueden generalizarse al caso no diagonalizable, véase [59] y referencias allí mencionadas.

En relación con el análisis del error para $S_{n,k}$ con n y k fijos, el siguiente resultado, referido al caso lineal (3.16), es la clave para justificar el comportamiento de los métodos en procesos no lineales y aplicados en modo cíclico, [58, 62].

Teorema 3.12. *Supongamos que, en las condiciones de (3.16), $I - T$ es no singular, T es diagonalizable con*

$$T = V\Lambda V^{-1}, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_N),$$

y V no singular. Sea $r(x)$ el residuo en x

$$r(x) = d - (I - T)x, \quad x \in \mathbb{R}^N,$$

\mathbb{P}_k el conjunto de polinomios de grado menor o igual que k y

$$\tilde{\mathbb{P}}_k = \{p \in \mathbb{P}_k / p(1) = 1\}.$$

Entonces

$$\|r(S_{n,k}^{RRE})\| \leq \min_{p \in \tilde{\mathbb{P}}_k} \|T^n p(T)r(S_0)\| \leq K_2(V)\Omega_{n,k}\|r(S_0)\|, \quad (3.41)$$

donde

- $K_2(V) = \|V\|\|V^{-1}\|$ es el número de condición de V en norma euclídea.
- $\Omega_{n,k} := \min_{p \in \tilde{\mathbb{P}}_k} \max_i |\lambda_i^n p(\lambda_i)|$.

Comentarios al teorema 3.12

1. Se puede obtener un resultado similar para el método MPE siempre que se asegure la existencia de $S_{n,k}^{MPE}$, [61].
2. La cantidad $\Omega_{n,k}$ puede acotarse en función del espectro de T . Las cotas pueden expresarse en términos de n y k , proporcionando, a través de (3.41), el control de los residuos, [59].

Como se ha mencionado anteriormente, el teorema 3.12 es la base para el siguiente resultado teórico, cf. teorema 3.10.

Teorema 3.13. Supongamos que S es punto fijo de una aplicación $f : D \subset \mathbb{C}^N \rightarrow \mathbb{C}^N$, con D abierto y convexo, tal que

- f es diferenciable en un entorno de $S \in D$.
- $I - f'(S)$ es no singular.
- Existe $L > 0$ tal que si $x, y \in D$

$$\|f'(x) - f'(y)\| \leq L\|x - y\|.$$

(A la izquierda de la desigualdad se utiliza la norma matricial.)

Sea $\{X_n\}_{n=0}^\infty$ la sucesión obtenida al aplicar el método MPE o el método RRE en modo cíclico, a partir de la sucesión

$$S_{m+1} = f(S_m), \quad m = 0, 1, \dots,$$

con $l = k_n$ el grado del polinomio mínimo de $f'(S)$ con respecto a $X_n - S$ en el ciclo $n + 1$. Se define

$$f^0(x) = x, \quad f^i(x) = f(f^{i-1}(x)), \quad i = 1, 2, \dots,$$

para cada $n = 0, 1, \dots$, la matriz $N \times k_n$ por columnas

$$H_n(x) = \left[\frac{f^1(x) - f^0(x)}{\|f^1(x) - f^0(x)\|} \mid \frac{f^2(x) - f^1(x)}{\|f^2(x) - f^1(x)\|} \mid \dots \mid \frac{f^{k_n}(x) - f^{k_n-1}(x)}{\|f^{k_n}(x) - f^{k_n-1}(x)\|} \right],$$

y, por último,

$$\alpha_n(x) = (\det(H_n^*(x)H_n(x)))^{\frac{1}{2}},$$

donde $*$ denota traspuesta conjugada. Supongamos que existen $\alpha > 0$ y un entero $M > 0$ tal que

$$\alpha_n(X_n) \geq \alpha, \quad n \geq M.$$

Entonces, para X_0 suficientemente cerca de S , se tiene que

$$\|X_{n+1} - S\| = \mathcal{O}(\|X_n - S\|^2), \quad n = 0, 1, \dots$$

La demostración puede consultarse en [34]; véase también [35, 59]. El resultado, puramente teórico, está limitado en la práctica por el desconocimiento de $f'(S)$. Una discusión en términos de experimentación numérica con ejemplos de los teoremas 3.10 y 3.13 puede consultarse en [63, 35].

3.4. Aplicaciones de la extrapolación vectorial

Ilustramos aquí los contenidos de las secciones anteriores aplicando extrapolación a la resolución numérica de dos problemas.

3.4.1. Aceleración en procesos de Markov

En el caso del primer ejemplo, es necesario recordar algunas definiciones.

Definición 3.7. Sea $A = (a_{ij})_{i,j=1}^N \in \mathbb{R}^{N \times N}$.

- Se llama grafo de A , y se denota por $G(A)$, al grafo dirigido de N nodos Q_1, \dots, Q_N , con un enlace directo entre Q_i y Q_j si y sólo si $a_{ij} \neq 0$.
- Se dice que $G(A)$ es fuertemente conexo si para cada par de nodos (Q_i, Q_k) hay un camino de enlaces directos que van de Q_i a Q_k .
- Se dice que A es irreducible si $G(A)$ es fuertemente conexo.

Para matrices A no negativas ($a_{ij} \geq 0$), la propiedad de ser irreducible proporciona información sobre su espectro, [44]:

Teorema 3.14 (Perron-Frobenius). *Si A es no negativa ($A \geq 0$) e irreducible, entonces:*

- El radio espectral de A , $\rho(A)$, es un autovalor de A , simple y positivo.*
- Existe un autovector $x > 0$ tal que $Ax = \rho(A)x$. Los únicos autovectores no negativos de A (independientes del autovalor) son los múltiplos positivos del único autovector p definido por*

$$Ap = \rho(A)p, \quad p > 0, \quad \|p\|_1 = 1,$$

llamado vector de Perron de A por la derecha y donde $\|\cdot\|_1$ denota la norma uno en \mathbb{R}^N .

Cuando la matriz es positiva ($a_{ij} > 0, \forall i, j$), el teorema de Perron para matrices positivas asegura que, para el resto de autovalores $\lambda \neq \rho(A)$, se tiene que $|\lambda| < \rho(A)$. En el caso de que A sea no negativa, el carácter irreducible no es suficiente para asegurar esta propiedad.

Definición 3.8. Se dice que una matriz A no negativa e irreducible es primitiva si el único autovalor de A en la circunferencia espectral

$$C(A) = \{z = (x, y) \in \mathbb{R}^2 / |z| = \rho(A)\},$$

es el propio radio espectral $\rho(A)$.

No es difícil ver que esta propiedad equivale a la convergencia de la recurrencia

$$x(k+1) = Ax(k),$$

hacia el autoespacio asociado al radio espectral, más concretamente, [44]:

Proposición 3.1. *Una matriz A , no negativa e irreducible, es primitiva si y sólo si existe $\lim_{k \rightarrow \infty} \left(\frac{A}{\rho(A)} \right)^k$, con*

$$\lim_{k \rightarrow \infty} \left(\frac{A}{\rho(A)} \right)^k = \frac{pq^T}{q^T p} > 0,$$

siendo $q > 0$ el vector de Perron de A^T por la derecha (llamado también vector de Perron de A por la izquierda).

Definición 3.9. Se dice que una matriz $P = (p_{ij})_{i,j=1}^N$ es estocástica si $P \geq 0$ y para cada $j = 1, \dots, N$

$$\sum_{i=1}^N p_{ij} = 1.$$

Pasamos a utilizar los conceptos anteriores en el contexto de las cadenas finitas de Markov. Recordemos, (véase, e.g. [65, 66]) que una cadena de Markov discreta y finita es un proceso estocástico discreto en el que las variables aleatorias X_n , $n = 0, 1, \dots$, toman valores en un conjunto finito de estados $\{s_1, \dots, s_N\}$ y cuya función de distribución de probabilidad condicionada satisface la llamada propiedad de Markov

$$\text{prob}(X_{n+1} = s_{i_{n+1}}/X_n = s_{i_n}, \dots, X_0 = s_{i_0}) = \text{prob}(X_{n+1} = s_{i_{n+1}}/X_n = s_{i_n}), \quad (3.42)$$

para todo n y los posibles estados $s_{i_0}, \dots, s_{i_{n+1}}$. Se trata de un proceso sin memoria, en el sentido de que el estado del sistema en tiempo $n + 1$ depende sólo de dónde se encuentre en el instante n .

La transición entre estados del instante n al instante $n + 1$ viene dada por la llamada matriz de transición

$$P(n) = (p_{ij}(n))_{i,j=1}^N, \quad p_{ij}(n) = \text{prob}(X_{n+1} = s_j/X_n = s_i), \quad (3.43)$$

que es una matriz estocástica. La cadena de Markov se dice que es estacionaria si $p_{ij}(n) = p_{ij}$ para todo $i, j = 1, \dots, N$ y todo n ; en ese caso escribimos $P(n) = P$.

La evolución de una cadena de Markov finita de N estados se obtiene a partir de la matriz (3.43) y de los valores de distribución de probabilidades en cada instante k

$$p(k) = (p_1(k), \dots, p_N(k))^T, \quad p_j(k) = \text{prob}(X_k = s_j),$$

esto es, $p_j(k)$ es la probabilidad de que en el instante k el sistema se encuentre en el estado j . Usando las propiedades de la probabilidad condicionada y la propiedad de Markov (3.42), no es difícil ver que

$$p(k+1) = Pp(k), \quad k = 0, 1, \dots, \quad (3.44)$$

asumiendo que el proceso es estacionario. Es decir, $p(k) = P^k p(0)$, siendo $p(0)$ la distribución inicial de probabilidades. De esta forma, el elemento (i, j) de P^k muestra la probabilidad de cambiar del estado i al estado j en k pasos.

Algunas propiedades de la matriz P , deducibles a partir de su carácter de matriz estocástica y del teorema de Perron-Frobenius, proporcionan información sobre la evolución del proceso de Markov dado por (3.44), [44].

Teorema 3.15. *Sea P la matriz estocástica que define la recurrencia (3.44). Entonces $\rho(P) = 1$. Además:*

(a) *Si P es primitiva*

$$\lim_{k \rightarrow \infty} p(k) = \pi,$$

siendo π el vector de Perron de P por la derecha.

(b) *Si P es irreducible pero no primitiva*

$$\lim_{k \rightarrow \infty} \left(\frac{p(0) + \dots + p(k-1)}{k} \right) = \pi.$$

Según el teorema anterior, el carácter de matriz primitiva asegura la convergencia del proceso (3.44), si bien ésta dependerá de factores como la magnitud del resto del espectro de P y su cercanía al radio espectral. Es en este contexto en el que los métodos de aceleración de convergencia pueden ser útiles y aplicables en cualquier modelo que esté gobernado por

procesos del tipo (3.44), sea cual sea el método numérico utilizado para la implementación, [6, 65].

Existe en la literatura gran variedad de modelos disponibles para ilustrar el uso de la extrapolación en (3.44), [5, 64, 46, 36]. Presentaremos aquí uno de los más conocidos, como es el cálculo del PageRank.

El PageRank (PR) es un procedimiento de base matemática que asocia a cada página existente en la red de internet un número que cuantifica su relevancia y fue originariamente propuesto por Brin y Page, [19]. Para la selección de páginas relacionadas con la petición de un usuario de internet, los motores de búsqueda en la red constan básicamente de dos etapas:

- Un procesamiento del texto indicado por el usuario, para encontrar aquellos documentos relacionados, en algún sentido semántico, con la búsqueda, [7].
- Un criterio para ordenar las páginas seleccionadas, [37].

El modelo original de Brin y Page, [19], se centra en el segundo punto, basando el criterio de ordenación en la búsqueda de un número (el PageRank) que cuantifique la relevancia de una página. Las reglas principales son las siguientes:

1. Se supone que el usuario navega por la red de dos posibles formas: siguiendo los enlaces de la página en la que se encuentre, con cierta probabilidad d , o eligiendo una página al azar (con probabilidad $1 - d$).
2. Los vínculos entrantes de una página procedente de páginas relevantes deberían tener más peso que aquellos procedentes de páginas poco importantes.
3. La relevancia de un enlace entrante debería escalarse en función del número de enlaces salientes de la página de la que procede.

Bajo estas reglas, la formulación del modelo es como sigue (cf. [38, 52] para más detalles). Supongamos que se navega en un universo de N páginas T_1, \dots, T_N , cuyas relaciones se pueden representar en forma de grafo con una matriz de conectividad asociada $G = (g_{ij})_{i,j=1}^N$, donde

$$g_{ij} = \begin{cases} 1, & \text{si existe enlace de } T_j \text{ a } T_i, \\ 0, & \text{en otro caso.} \end{cases}$$

Si desde cualquier página puede alcanzarse cualquier otra (es decir, si todas las páginas tienen al menos un enlace saliente) se puede definir la matriz $M = (m_{ij})_{i,j=1}^\infty$, donde

$$m_{ij} = \begin{cases} \frac{1}{C_j}, & \text{si existe enlace de } T_j \text{ a } T_i, \\ 0, & \text{en otro caso,} \end{cases}$$

siendo C_j el número de enlaces que salen de la página j , es decir

$$C_j = \sum_{i=1}^N g_{ij}, \quad j = 1, \dots, N. \quad (3.45)$$

Se puede comprobar fácilmente que M es estocástica. Por otra parte, es natural asumir que cada página no tiene enlace a sí misma, de manera que la diagonal de M es nula. Sin embargo,

la definición de esta matriz plantea dos problemas que hay que resolver para la formulación del PageRank:

- (P1) Podemos tener páginas sin enlaces salientes ('colgadas'). M contiene entonces alguna columna nula y perdería su carácter de matriz estocástica.
- (P2) Aunque no tengamos páginas colgadas, quizá no podamos ir de cualquier página T_j a cualquier otra T_i mediante un camino de enlaces. Esto es, necesitaríamos que M fuese irreducible.

Para resolver (P1), se asume que cada página colgada va a apuntar a cada página de la red (incluida ella misma) reemplazando la columna de ceros en M por el vector $\frac{1}{N}e_N$, siendo $e_N = (1, 1, \dots, 1)^T$. Se obtiene de esta forma la matriz estocástica

$$\tilde{M} = M + \frac{1}{N}e_N b^T, \quad (3.46)$$

donde los elementos del vector columna $b = (b_1, \dots, b_N)^T$ valen uno o cero según que la página con el mismo subíndice sea colgada o no. Esto corresponde a asumir una distribución uniforme para la transición a otras páginas desde páginas sin enlaces salientes.

Para resolver (P2), se construye la matriz $E = (e_N e_N^T)/N$ y se genera

$$P = d\tilde{M} + (1-d)E. \quad (3.47)$$

Así, en la formulación (3.47) se incorporan las características principales del modelo: el término $d\tilde{M}$ cuantifica la navegación siguiendo los enlaces de las páginas (e incorporando la solución a las páginas colgadas), mientras que $(1-d)E$ representa la navegación al azar y que, dada la forma de la matriz E , sigue una distribución uniforme.

Con todo ello, no es difícil comprobar que la matriz P , definida en (3.47), es positiva, estocástica e irreducible. El teorema de Perron, [44], asegura entonces que la recurrencia correspondiente (3.44) converge, para cualquier distribución inicial $p(0)$, al vector de Perron $\pi > 0$, de manera que

$$\pi = P\pi. \quad (3.48)$$

En este contexto, $\pi = (\pi_1, \dots, \pi_N)^T$ se llama vector PageRank (o vector PR) y contiene los valores π_i de dicho índice PR para la página T_i , $i = 1, \dots, N$. Por componentes, (3.48) se escribe como

$$\pi_i = \frac{(1-d)}{N} + d \sum_{j=1}^{n_i} \frac{\pi_j}{C_j}, \quad 1 \leq i \leq N,$$

donde n_i denota el número de enlaces entrantes de la página T_i y C_i el número de enlaces salientes, cf. (3.45).

En general, (3.48) no puede resolverse exactamente. Un procedimiento numérico elemental para su resolución aproximada es el método de la potencia, cf. (3.44), cuya convergencia viene asegurada por el teorema 3.15. Para el caso de la matriz (3.47), puede hacerse una implementación más eficiente. Observemos, en primer lugar, que si $p(0) = \frac{e_n}{N}$, no es difícil comprobar, por inducción sobre k , que (3.44) puede escribirse en la forma más conveniente

$$p(k+1) = d\tilde{M}p(k) + \frac{(1-d)}{N}e_N, \quad k = 0, 1, \dots, \quad (3.49)$$

con \tilde{M} dada por (3.46). Así, la programación tendría la estructura siguiente:

- Los datos de entrada vienen dados por la matriz de conectividad G (que incluyen el valor N) y el parámetro de probabilidad d .
- A partir de $p(0) = \frac{e_N}{N}$, se implementa el método de la potencia según (3.49) (lo que, en particular requiere formar \tilde{M}) y utilizando un criterio de parada basado en fijar una tolerancia para el error en norma euclídea del residuo asociado a (3.49).
- El dato de salida es el valor PR dado por la última aproximación a (3.48).

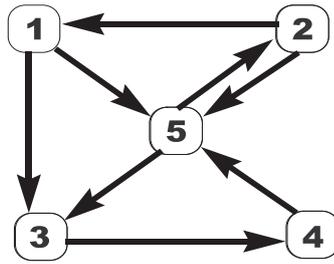


Figura 3.1: Grafo asociado a una red de $N = 5$ páginas.

$d = 0.6$	$d = 0.85$	$d = 0.95$
0.13025	0.10036	0.088908
0.16751	0.16555	0.166123
0.20659	0.20820	0.208354
0.20395	0.20697	0.207936
0.29170	0.31893	0.328679

Tabla 3.1: Valores del PR de las páginas de la red de la figura 3.1 y para diferentes valores de d .

Llevamos a cabo la validación del algoritmo con el ejemplo de una red de $N = 5$ páginas, cuyo grafo asociado viene dado por la figura 3.1. Fijando una tolerancia para el error de 10^{-15} , analizamos primero la influencia del parámetro d sobre la convergencia. En la tabla 3.1 se muestran los valores aproximados del vector PR obtenidos con tres valores para d . Las aproximaciones son diferentes (estamos modificando las condiciones del modelo al variar d) pero no el orden de importancia de las páginas (de mayor a menor PR, éste es 5 – 3 – 4 – 2 – 1 en las tres columnas).

El parámetro d influye también en la velocidad de convergencia. La figura 3.2 muestra, en escala semilogarítmica, el error en la iteración en función del número de iteraciones. Se observa una convergencia más lenta cuanto mayor es el parámetro d .

Por otro lado, para un valor fijo de d , un segundo factor que influye en la convergencia es el tamaño N de la matriz, es decir, el número de páginas relacionadas que, dada la estructura hyperlink de la red, debe ser grande. Generando la matriz G de conectividad de tamaño 20, 40, 100 correspondiente a subredes de la Universidad de Valladolid y fijando el valor de

3 Extrapolación vectorial

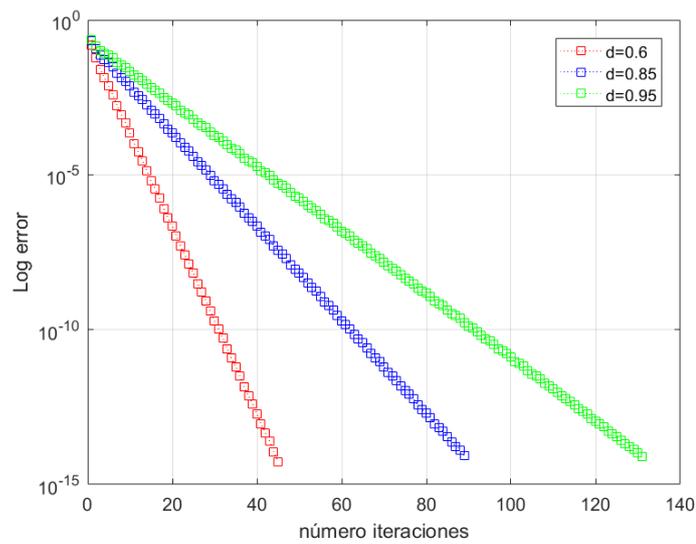


Figura 3.2: Logaritmo del error vs número de iteraciones.

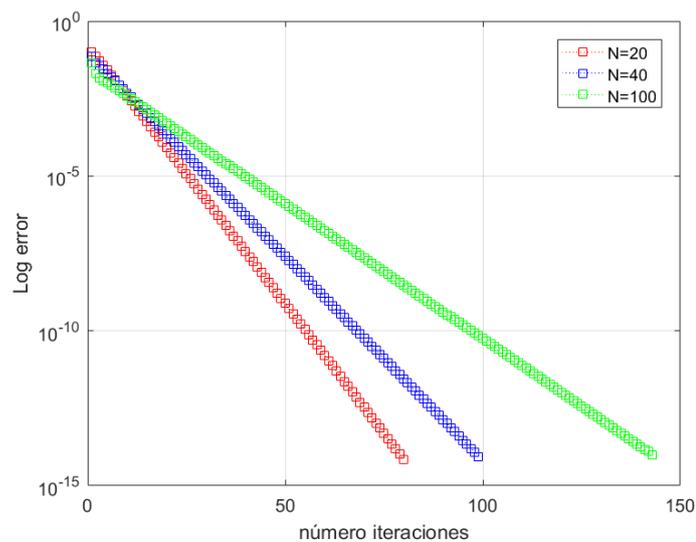


Figura 3.3: Logaritmo del error vs. número de iteraciones para matrices G de tamaño $N = 20, 40, 100$, con $d = 0.85$.

$d = 0.85$, en la figura 3.3 podemos ver el correspondiente comportamiento del residuo frente al número de iteraciones. Como era de esperar, el algoritmo es más lento a medida que aumenta el número de páginas de la red. La figura 3.3 muestra esta pérdida de eficiencia en un doble sentido. En primer lugar, a medida que el tamaño de la red crece, se requieren más iteraciones (y por tanto más tiempo) para alcanzar un determinado valor del residuo. También, para un número fijo de iteraciones, el error alcanzado en el residuo es mayor al

aumentar N , y con ello se pierde precisión al calcular los valores PageRank.

Para acelerar la convergencia del proceso y así aumentar la velocidad de cálculo de los índices PageRank, se puede aplicar la extrapolación vectorial al método de la potencia, [38, 57, 16]. Ello queda ilustrado en la figura 3.4, que compara el logaritmo del error en el residuo frente al número de iteraciones, para el experimento de la figura 3.3 con $N = 100$ proporcionado por el método de la potencia sin extrapolación y con extrapolaciones dadas por los métodos MPE y RRE para $k = 3$. En la figura se observa la mejora en la convergencia, tanto en el número de iteraciones para alcanzar un error en el residuo determinado como en el error obtenido para un número fijo de iteraciones.

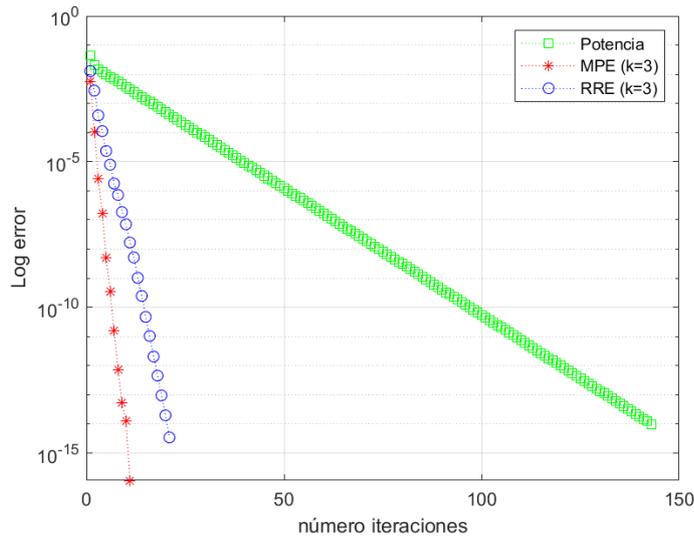


Figura 3.4: Logaritmo del error vs número de iteraciones, comparación MPE, RRE y método de la potencia. Sistema (3.44) con $N = 100$, $d = 0.85$.

3.4.2. Aproximación a soluciones especiales de ecuaciones en derivadas parciales.

Una ilustración más detallada de los mecanismos de aceleración de convergencia vectorial requiere tratar un caso no lineal. En este sentido, la segunda aplicación de los métodos de extrapolación vectorial que se menciona en esta memoria tiene que ver con la aproximación numérica de soluciones especiales de algunas ecuaciones en derivadas parciales (EDP) de evolución. Se trata de soluciones de tipo estacionario, ondas viajeras, etc. y cuya búsqueda, en un contexto más o menos abstracto, convierten la EDP de evolución en un sistema diferencial para la solución v , de la forma

$$\mathcal{L}v - \mathcal{N}(v) = 0, \quad (3.50)$$

con una parte lineal representada por un operador diferencial \mathcal{L} invertible y términos no lineales agrupados en un operador \mathcal{N} , definidos ambos en algún espacio normado. Como

3 Extrapolación vectorial

ejemplo, consideramos el sistema de EDPs

$$\begin{aligned}\eta_t + (u + A(\eta, u) + au_{xx} - b\eta_{xt})_x &= 0, \\ u_t + (\eta + B(\eta, u) + c\eta_{xx} - du_{xt})_x &= 0, \quad -\infty < x < \infty, \quad t > 0,\end{aligned}\tag{3.51}$$

que aparece como modelo en la teoría matemática de ondas de agua, [9, 10, 24, 25]. En (3.51), $a, b, c, d \in \mathbb{R}$ son constantes, mientras que A y B son funciones cuadráticas de las incógnitas η y u ,

$$\begin{aligned}A(\eta, u) &= \alpha_{11}\eta^2 + \alpha_{12}\eta u + \alpha_{22}u^2, \\ B(\eta, u) &= \beta_{11}\eta^2 + \beta_{12}\eta u + \beta_{22}u^2,\end{aligned}\tag{3.52}$$

con coeficientes reales $\alpha_{ij}, \beta_{ij}, i, j = 1, 2$. Desde el punto de vista del modelo que origina la derivación de (3.51), (3.52), tiene interés la búsqueda de soluciones de tipo onda viajera: son soluciones de la forma

$$\eta(x, t) = \eta(x - c_s t), \quad u(x, t) = u(x - c_s t),\tag{3.53}$$

que representan perfiles de forma permanente y que viajan a velocidad constante $c_s \in \mathbb{R}$ o, equivalentemente, soluciones estacionarias en las variables (y, t) con $y = x - c_s t$. De este modo, los perfiles $\eta(y), u(y)$ deben ser solución del sistema diferencial

$$\begin{aligned}-c_s \eta + u + A(\eta, u) + au'' + bc_s \eta'' &= c_1, \\ -c_s u + \eta + B(\eta, u) + c\eta'' + dc_s u'' &= c_2,\end{aligned}\tag{3.54}$$

con c_1, c_2 constantes. Fijando $c_1 = c_2 = 0$ (lo que puede justificarse en términos del modelo a estudio), entonces (3.54) tiene la forma (3.50) con $v = (\eta, u)^T$ y los operadores

$$\mathcal{L} = \begin{pmatrix} c_s(1 - b\partial_{yy}) & -(1 + a\partial_{yy}) \\ -(1 + c\partial_{yy}) & c_s(1 - d\partial_{yy}) \end{pmatrix}, \quad \mathcal{N}(\eta, u) = \begin{pmatrix} A(\eta, u) \\ B(\eta, u) \end{pmatrix}.\tag{3.55}$$

La resolución de (3.50) debe ser, en general, aproximada. Un procedimiento típico para ello consiste en utilizar métodos iterativos. Si bien el desarrollo de la teoría puede llevarse a cabo en el contexto abstracto del espacio normado en el que el sistema esté definido, suele ser más práctico discutir la aproximación a partir de alguna discretización del sistema diferencial (3.50) que nos lleve a un sistema algebraico de la forma

$$F_h(v_h) = L_h(v_h) - N_h(v_h) = 0,\tag{3.56}$$

con una estructura similar, donde L_h es una matriz invertible y N_h un vector de componentes no lineales, para la aproximación representada por un vector $v_h \in \mathbb{R}^N$, con h denotando los correspondientes parámetros de la discretización. La estrategia pasa entonces por la resolución iterativa del sistema algebraico (3.56) y la necesidad de introducir técnicas de extrapolación vectoriales.

La resolución de (3.56) por iteración ofrece, como es natural, muchas alternativas: desde procedimientos clásicos (métodos de tiro, algoritmos clásicos de punto fijo, técnicas de tipo Newton, etc) a otros más sofisticados y *ad hoc* (e.g [76]), dado que el éxito en la iteración depende de las propiedades del sistema (3.56). En este sentido, hay muchos modelos que contienen términos no lineales que son funciones homogéneas de grado $p > 1$, es decir

$$N_h(\lambda z) = \lambda^p N_h(z), \quad z \in \mathbb{R}^N, \quad \lambda \neq 0.\tag{3.57}$$

Si N es diferenciable, esto implica que

$$N'_h(z)z = pN_h(z).$$

Supongamos entonces que aplicamos el algoritmo clásico de punto fijo a (3.56)

$$L_h v_h^{[v+1]} = N_h(v_h^{[v]}), \quad v = 0, 1, \dots, \quad (3.58)$$

donde $v_h^{[v]}$ representa el iterante vectorial v -ésimo, aproximación a v_h . Observemos entonces que, de (3.57), se tiene

$$(L_h)^{-1} N'_h(v_h) = p (L_h)^{-1} N_h(v_h) = p v_h,$$

de manera que $\lambda = p > 1$ es un autovalor de la matriz de iteración $S = (L_h)^{-1} N'_h(v_h)$, lo que implica que (3.58) no es convergente en general.

Tal es la situación de nuestro modelo ejemplo (3.54), (3.55). Supongamos que \mathcal{L} en (3.55) es invertible, y que discretizamos (3.54) en la forma (3.56) con

$$L_h = \begin{pmatrix} c_s(I - bD_h^2) & -(I + aD_h^2) \\ -(I + cD_h^2) & c_s(I - dD_h^2) \end{pmatrix}, \quad N_h(\eta_h, u_h) = \begin{pmatrix} A(\eta_h, u_h) \\ B(\eta_h, u_h) \end{pmatrix}, \quad (3.59)$$

donde $\eta_h, u_h \in \mathbb{R}^N$ son aproximaciones a η y u , respectivamente, L_h es una matriz $2N \times 2N$ definida por bloques $N \times N$, $I = I_N$ es la matriz identidad de tamaño N , $D_h \in \mathbb{R}^{N \times N}$ representa una aproximación al operador ∂_y , mientras que

$$\begin{aligned} A(\eta_h, u_h) &= \alpha_{11}\eta_h \cdot \eta_h + \alpha_{12}\eta_h \cdot u_h + \alpha_{22}u_h \cdot u_h, \\ B(\eta_h, u_h) &= \beta_{11}\eta_h \cdot \eta_h + \beta_{12}\eta_h \cdot u_h + \beta_{22}u_h \cdot u_h, \end{aligned} \quad (3.60)$$

son vectores en \mathbb{R}^N donde, si $z = (z_1, \dots, z_N)^T$, $w = (w_1, \dots, w_N)^T$, entonces $z \cdot w$ es el producto de Hadamard

$$z \cdot w = (z_1 w_1, \dots, z_N w_N)^T. \quad (3.61)$$

Observamos así que el término N_h en (3.59) hereda el carácter homogéneo de grado $p = 2$ de \mathcal{N} en (3.55).

Nos planteamos entonces la introducción de métodos de extrapolación vectorial para convertir el proceso divergente en (3.58) con N_h verificando (3.57), en una sucesión convergente, ilustrándolo con la generación numérica de aproximaciones a varios tipos de ondas viajeras (3.53).

Aproximación espectral de colocación de Fourier

Para ello, pasamos primero a describir el método utilizado en los experimentos numéricos para aproximar las soluciones de (3.54) con $c_1 = c_2 = 0$. Fijemos $l > 0$ y consideremos el sistema

$$\begin{aligned} -c_s \eta + u + A(\eta, u) + au'' + bc_s \eta'' &= 0, \\ -c_s u + \eta + B(\eta, u) + c\eta'' + dc_s u'' &= 0, \end{aligned} \quad x \in (-l, l), \quad (3.62)$$

con condiciones periódicas, donde A y B vienen dadas por (3.52). El sistema (3.62) se tomará como aproximación a (3.54) utilizando l como parámetro. Las diferentes soluciones, descritas más abajo, pueden requerir un valor de l mayor o menor para que la aproximación a (3.54) dada por (3.62) no afecte a la precisión de los cálculos.

3 Extrapolación vectorial

Sea $N \geq 1$ entero par y los nodos de colocación

$$y_j = -l + jh, \quad j = 0, \dots, N-1, \quad h = \frac{2l}{N}. \quad (3.63)$$

Consideramos el conjunto de polinomios trigonométricos

$$S_N = \text{span} \left\{ e^{ik\frac{\pi}{l}(y+l)}, \quad k \in \mathbb{Z}, \quad -\frac{N}{2} \leq k \leq \frac{N}{2} - 1 \right\},$$

y definimos la aproximación espectral de colocación de Fourier de la solución (η, u) de (3.62) (asumiendo que existe) como el par $(\eta^N, u^N) \in S_N \times S_N$ tal que se cumple

$$\begin{aligned} -c_s \eta^N + u^N + A(\eta^N, u^N) + a(u^N)'' + bc_s(\eta^N)'' &= 0, \\ -c_s u^N + \eta^N + B(\eta^N, u^N) + c(\eta^N)'' + dc_s(u^N)'' &= 0, \end{aligned} \quad (3.64)$$

en los nodos de colocación (3.63). La aproximación (η^N, u^N) viene normalmente representada por los valores nodales

$$\eta_h = (\eta_0, \dots, \eta_{N-1})^T, \quad u_h = (u_0, \dots, u_{N-1})^T \quad (3.65)$$

con $\eta_j = \eta^N(y_j)$, $u_j = u^N(y_j)$, $j = 0, \dots, N-1$, como aproximaciones a los valores de la sucesión $\eta(y_j)$, $u(y_j)$ en los nodos de colocación (3.63). Esto viene justificado por las representaciones

$$\eta^N(y) = \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{\eta}_p e^{\frac{\pi}{l}ip(y+l)}, \quad u^N(y) = \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_p e^{\frac{\pi}{l}ip(y+l)}, \quad (3.66)$$

donde

$$\hat{\eta}_p = \frac{1}{N} \sum_{j=0}^{N-1} \eta_j e^{-ipj\frac{2\pi}{N}}, \quad \hat{u}_p = \frac{1}{N} \sum_{j=0}^{N-1} u_j e^{-ipj\frac{2\pi}{N}}, \quad -\frac{N}{2} \leq p \leq \frac{N}{2} - 1, \quad (3.67)$$

son los coeficientes de Fourier discretos de η^N y u^N respectivamente, calculables mediante la transformada discreta de Fourier (DFT), [49], de los vectores η_h , u_h dados por (3.65).

El correspondiente sistema (3.60) para los valores nodales se deduce de (3.63) utilizando los representantes (3.66), (3.67). En este caso, D_h es el operador de diferenciación pseudoespectral, [68, 54, 21]; para $f \in S_N$ dado por

$$f(y) = \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_p e^{\frac{\pi}{l}ip(y+l)}, \quad \hat{f}_p = \frac{1}{N} \sum_{j=0}^{N-1} f(y_j) e^{-ipj\frac{2\pi}{N}}, \quad -\frac{N}{2} \leq p \leq \frac{N}{2} - 1,$$

entonces $D_h f \in S_N$ viene representado por los valores nodales

$$(D_h f)_j = \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} \left(ip \frac{\pi}{l} \right) \hat{f}_p e^{ipj\frac{2\pi}{N}}, \quad 0 \leq j \leq N-1. \quad (3.68)$$

Así, la diagonalización del operador D_h en el espacio de Fourier suele utilizarse en la implementación de (3.64) para los coeficientes de Fourier discretos, recuperando los valores

nodales (3.65) con la DFT inversa. El correspondiente sistema tendría la forma

$$\begin{aligned} -c_s \hat{\eta}_p + \hat{u}_p + A(\eta_h, u_h) \hat{p} + a \left(ip \frac{\pi}{l} \right)^2 \hat{u}_p + bc_s \left(ip \frac{\pi}{l} \right)^2 \hat{\eta}_p &= 0, \\ -c_s \hat{u}_p + \hat{\eta}_p + B(\eta_h, u_h) \hat{p} + c \left(ip \frac{\pi}{l} \right)^2 \hat{\eta}_p + dc_s \left(ip \frac{\pi}{l} \right)^2 \hat{u}_p &= 0, \end{aligned} \quad \frac{-N}{2} \leq p \leq \frac{N}{2} - 1, \quad (3.69)$$

donde $A(\eta_h, u_h) \hat{p}$, $B(\eta_h, u_h) \hat{p}$ denotan los p -ésimos coeficientes de Fourier discretos de (3.60). Su cálculo combina una DFT inversa para pasar a η_h , u_h , a partir de sus coeficientes de Fourier discretos, productos componente a componente según (3.61) y una aplicación de la DFT para volver al espacio de Fourier, véase [68, 54, 21] para los detalles. El procedimiento iterativo (3.58) se implementa sobre el sistema (3.69) para cada p , en la forma

$$\begin{pmatrix} c_s \left(1 - b \left(p \frac{\pi}{l} \right)^2 \right) & - \left(1 - a \left(p \frac{\pi}{l} \right)^2 \right) \\ - \left(1 - c \left(p \frac{\pi}{l} \right)^2 \right) & c_s \left(1 + d \left(p \frac{\pi}{l} \right)^2 \right) \end{pmatrix} \begin{pmatrix} \hat{\eta}_p^{[v+1]} \\ \hat{u}_p^{[v+1]} \end{pmatrix} = \begin{pmatrix} A(\eta_h^{[v]}, u_h^{[v]}) \hat{p} \\ B(\eta_h^{[v]}, u_h^{[v]}) \hat{p} \end{pmatrix}, \quad \frac{-N}{2} \leq p \leq \frac{N}{2} - 1, \quad (3.70)$$

donde $\hat{\eta}_p^{[v]}$, $\hat{u}_p^{[v]}$ denotan los p -ésimos coeficientes de Fourier discretos del iterante $(\eta_h^{[v]}, u_h^{[v]})$ $v = 0, 1, \dots$

Sobre la iteración (3.70), inicialmente divergente, se aplica la extrapolación vectorial. Para los ejemplos mostrados a continuación, se han utilizado cuatro métodos de extrapolación (VEA, TEA1, MPE y RRE), implementados en modo cíclico completo y diferentes longitudes de extrapolación ($k+1$ en el caso de métodos polinómicos y $2k$ en el caso de los algoritmos vectoriales de tipo ε). El control de la iteración está basado en el residuo

$$RES(v) = L_h \begin{pmatrix} \eta_h^{[v]} \\ u_h^{[v]} \end{pmatrix} - N_h(u_h^{[v]}), \quad v = 0, 1, \dots, \quad (3.71)$$

donde $\eta_h^{[v]}$, $u_h^{[v]} \in \mathbb{R}^N$ representan, respectivamente, el iterante en el paso v que aproxima η_h y u_h . El modo cíclico se ejecuta mientras la norma euclídea del residuo (3.71) sea mayor que una tolerancia, que en los experimentos está fijada en 5×10^{-12} .

El sistema (3.51) admite diferentes tipos de soluciones de la forma (3.53), en función del rango elegido de los parámetros a, b, c y d (c.f. [24] y referencias en él). Ilustramos los experimentos realizados con tres casos, donde por sencillez se han fijado $c_s = 1.3$ (la velocidad de la onda) y los parámetros de (3.52)

$$\begin{aligned} \alpha_{11} &= 0, & \alpha_{12} &= 0.46, & \alpha_{22} &= 0, \\ \beta_{11} &= 0.23, & \beta_{12} &= 0, & \beta_{22} &= 0.73, \end{aligned}$$

así como $c_1 = c_2 = 0$ en (3.54).

Con estos datos, se ha implementado la extrapolación dada por los cuatro métodos a partir de un iterante inicial en forma de secante hiperbólica. Los cuatro métodos alcanzan convergencia en todos los casos expuestos. La precisión de los perfiles numéricos obtenidos se ha evaluado de dos formas:

3 Extrapolación vectorial

- Mostrando la convergencia de la norma euclídea del residuo (3.71) frente al número de iteraciones.
- Disponiendo los perfiles numéricos obtenidos como datos iniciales de un esquema numérico totalmente discreto para aproximar al sistema (3.51) y observando la evolución de la solución numérica obtenida. En todos los casos, la aproximación se comporta como una onda de forma permanente y viajando a velocidad constante, simulando la forma (3.53).

Por sencillez en la exposición, mostramos solamente los resultados correspondientes al comportamiento del residuo.

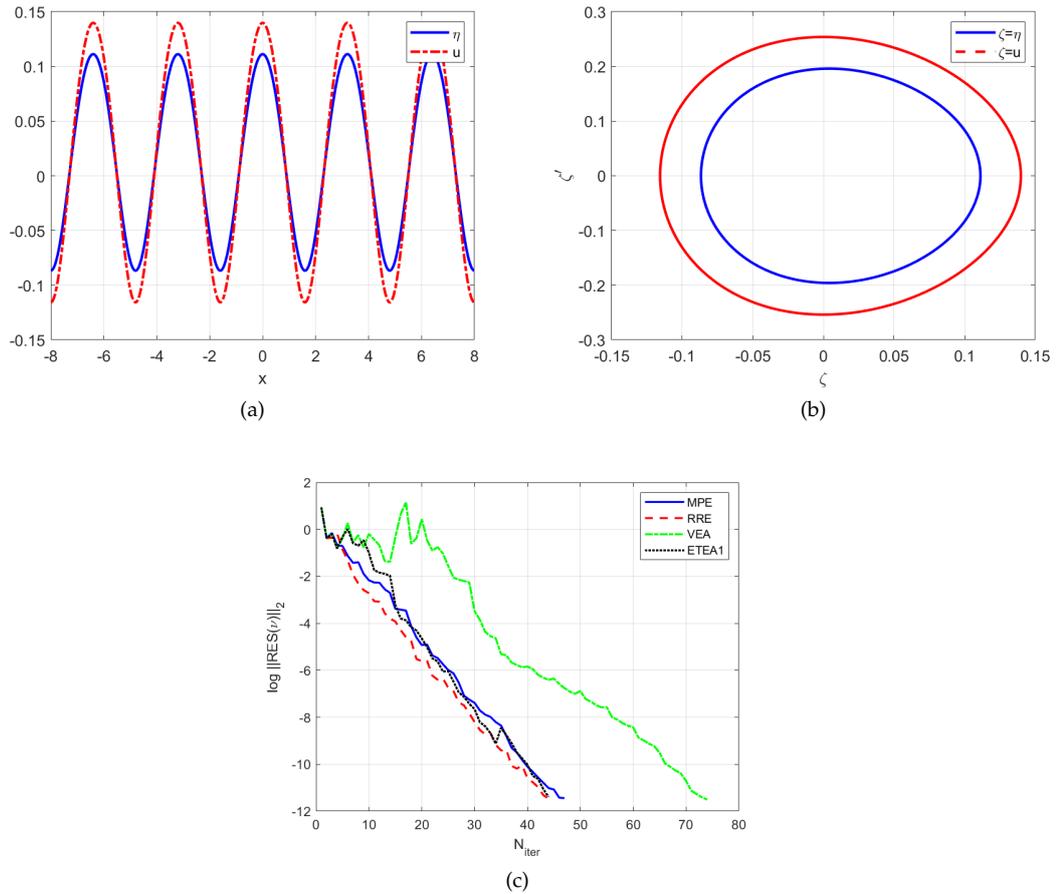


Figura 3.5: (a) Onda viajera periódica solución de (3.54) con $a = c = -1/6$, $b = 1/6$, $d = 0$, $\alpha_{11} = 0$, $\alpha_{12} = 0.46$, $\alpha_{22} = 0$, $\beta_{11} = 0.23$, $\beta_{12} = 0$, $\beta_{22} = 0.73$, $c_s = 1.3$; (b) mapa de fase; (c) logaritmo del error en el residuo (3.71) vs. número de iteraciones para los cuatro métodos con $k = 2$.

Ondas viajeras periódicas. Tomando $a = c = -1/6$, $b = 1/6$, $d = 0$, el sistema (3.54) admite soluciones de la forma (3.53) con perfiles η y u que son funciones periódicas. Tomando $l = 8$, la figura 3.5 (a) muestra las aproximaciones a dichos perfiles dados por el método MPE con

$k = 2$. El carácter periódico de las soluciones es observable en la figura 3.5 (b), que muestra el correspondiente mapa de fase aproximado para ambos perfiles. Para su implementación se han computado los valores $(\eta_j, \eta'_j), (u_j, u'_j), j = 0, \dots, N - 1$, donde $\eta'_j = (D_h \eta_h)_j, u'_j = (D_h u_h)_j$, utilizando (3.68).

Hemos comparado la aceleración proporcionada por los cuatro métodos fijando $k = 2$ y estudiando la norma euclídea del residuo (3.71) como función del número de iteraciones. El resultado se muestra en la figura 3.5c, donde se observa una mayor eficiencia de los métodos polinómicos y el método TEA (implementado en modo económico) frente al algoritmo VEA. Para otros valores de k , se observa un mayor rendimiento de los métodos polinómicos frente a los de tipo ε , véanse las conclusiones al final de los experimentos numéricos.

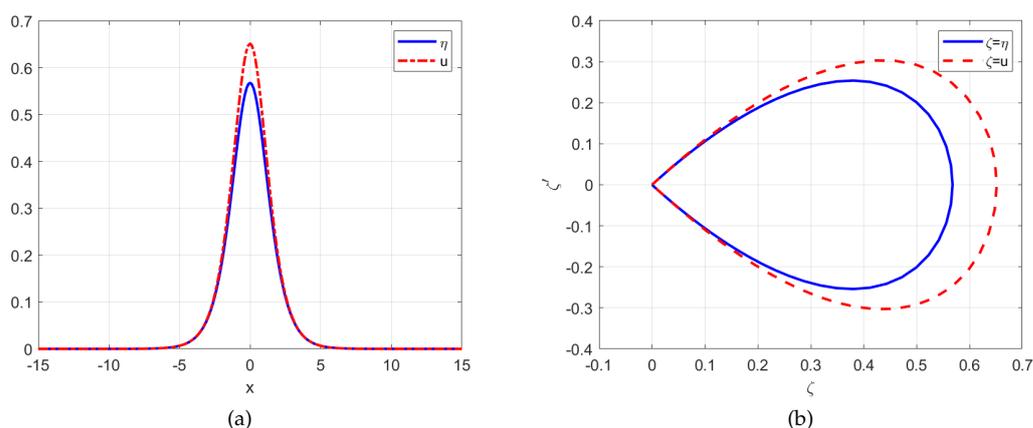


Figura 3.6: (a) Onda solitaria clásica solución de (3.54) con $a = c = 0, b = d = 1/6, \alpha_{11} = 0, \alpha_{12} = 0.46, \alpha_{22} = 0, \beta_{11} = 0.23, \beta_{12} = 0, \beta_{22} = 0.73, c_s = 1.3$; (b) mapa de fase.

Ondas solitarias clásicas. En el caso $a = c = 0, b = d = 1/6$, el sistema (3.54) admite soluciones de tipo (3.53) que son ondas solitarias viajando a velocidad constante c_0 y con perfiles que decrecen exponencialmente hacia cero en el infinito. Para una correcta simulación se requiere tomar un valor de l mayor ($l = 256$ en los experimentos) y hacer uso de la propiedad de que estas ondas pueden verse como límite de ondas viajeras periódicas, [8, 22]. La aproximación obtenida con el método VEA con $k = 2$ se muestra en la figura 3.6 (a). El mapa de fase correspondiente, figura 3.6 (b), ilustra el comportamiento homoclínico hacia el origen de la órbita asociada a cada perfil.

En éste y en el siguiente experimento hemos estudiado la influencia de la longitud de la extrapolación en el rendimiento numérico. Recordemos que, en la práctica, la elección de k no tiene reglas establecidas, más allá de la conveniencia de que sea pequeño para no aumentar el coste computacional. En la figura 3.8 (a) se muestra el logaritmo del error en el residuo frente al número de iteraciones para la generación numérica de la onda solitaria de la figura 3.6 (a) con el método MPE y tres valores de k . La conclusión es que el aumento de k mejora el rendimiento, alcanzándose, por ejemplo, un error fijo con menos iteraciones. El paso de $k = 2$ a $k = 3$ es significativo, en el sentido de que la mejora ya no es tan clara como al pasar de $k = 1$ a $k = 2$. Ello sugiere que valores mayores de k no aumentan el rendimiento lo suficiente como para compensar el coste computacional añadido.

Ondas solitarias generalizadas. En el caso $a = c = -1/6, b = 0, d = 1/6$, el sistema (3.54)

3 Extrapolación vectorial

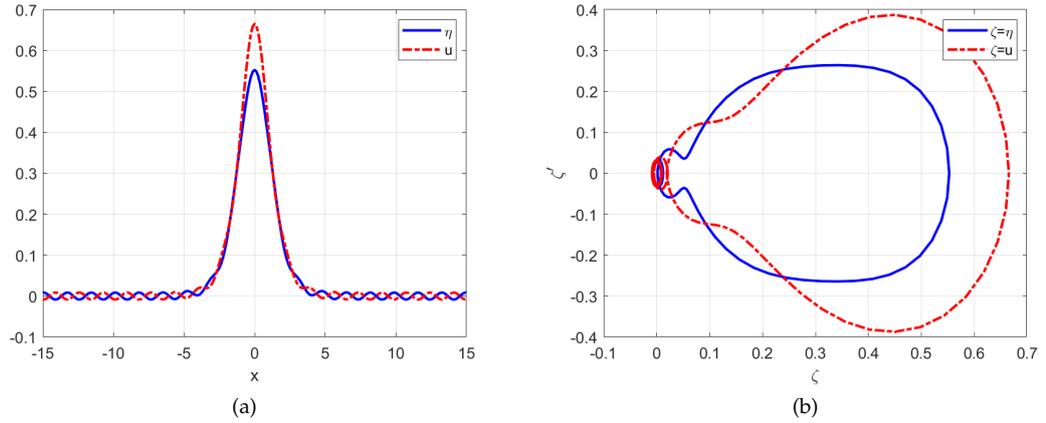


Figura 3.7: (a) Onda solitaria generalizada solución de (3.54) con $a = c = -1/6$, $b = 0$, $d = 1/6$, $\alpha_{11} = 0$, $\alpha_{12} = 0.46$, $\alpha_{22} = 0$, $\beta_{11} = 0.23$, $\beta_{12} = 0$, $\beta_{22} = 0.73$, $c_s = 1.3$; (b) mapa de fase.

admite un nuevo tipo de solución de la forma (3.53) llamada onda solitaria generalizada. Se trata de un perfil de tipo onda solitaria pero, a diferencia de las ondas solitarias clásicas, en este caso los perfiles $\eta(y)$, $u(y)$ se aproximan a una función periódica cuando $y \rightarrow \pm\infty$. Ello se refleja en la generación de oscilaciones de cierto periodo donde antes las ondas solitarias clásicas tendían a cero. Esto puede observarse en la figura 3.7 (a), que muestra la aproximación obtenida con el método VEA y $k = 2$. En el mapa de fase correspondiente (figura 3.7 (b)), se

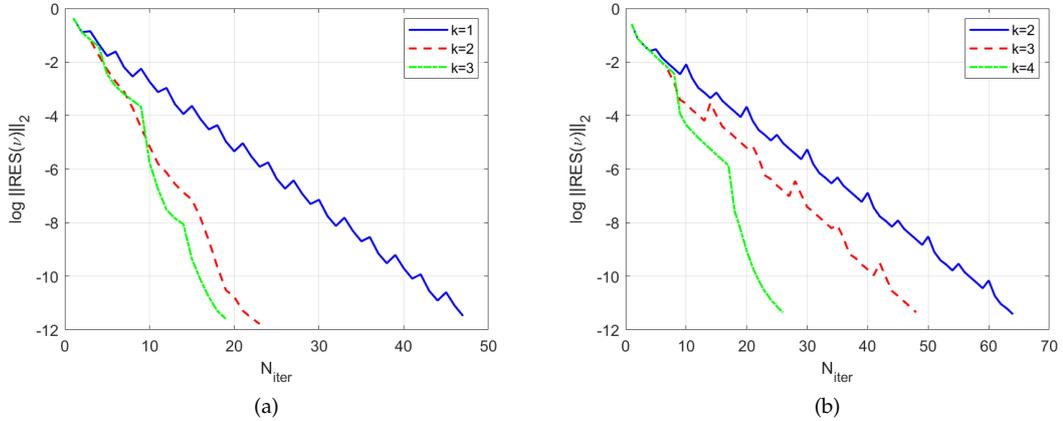


Figura 3.8: (a) Onda solitaria clásica, figura 3.6: logaritmo del error en el residuo (3.71) vs. número de iteraciones para el método MPE con $k = 2, 3, 4$; (b) onda solitaria generalizada, figura 3.7: logaritmo del error en el residuo (3.71) vs. número de iteraciones para el método VEA con $k = 2, 3, 4$.

puede observar el comportamiento asintótico homoclínico hacia una órbita periódica. Tales ondas solitarias generalizadas son observables en la naturaleza, [3], y su estudio en términos de sistemas dinámicos puede consultarse en, por ejemplo, [41]. El comportamiento de los

métodos con respecto a la longitud de extrapolación es similar a la del experimento anterior. Ello queda ilustrado en la figura 3.8 (b), que muestra el logaritmo del error en el residuo (3.71) como función del número de iteraciones obtenido con el método VEA y tres valores de k . En este caso, la mejora observada frente al coste computacional requiere tomar un valor de k un poco más alto que el del experimento con la onda solitaria clásica.

Las principales conclusiones de los tres grupos de experimentos son las siguientes:

- Los cuatro métodos de extrapolación alcanzan la convergencia. En general, se observa un mejor rendimiento de los métodos polinómicos frente a los de tipo ε , en un doble sentido:
 1. Un menor error en el residuo para un número de iteraciones fijo.
 2. Los métodos polinómicos alcanzan un error fijo con menos iteraciones.
- Los dos métodos polinómicos proporcionan un rendimiento muy similar, en consonancia con su equivalencia en el sentido especificado en [59] para el caso lineal. Por otro lado, el método TEA (en su versión económica) se muestra más competitivo frente a VEA.
- La experimentación sugiere tomar una longitud de extrapolación con k entre dos y cuatro. Valores superiores no mejoran en exceso el rendimiento y aumentan el coste computacional.

Bibliografía

Las referencias se listan por orden alfabético. Aquellas referencias con más de un autor están ordenadas de acuerdo con el primer autor.

- [1] A. C. Aitken, *On Bernoulli's numerical solution of algebraic equations*, Proceedings of the Royal Society of Edinburgh **46** (1926), 289–305.
- [2] ———, *On Interpolation by Iteration of Proportional Parts, without the Use of Differences*, Proceedings of the Edinburgh Mathematical Society **3** (1932), no. 1, 56–76.
- [3] T. R. Akylas and R. H. J. Grimshaw, *Solitary internal waves with oscillatory tails*, Journal of Fluid Mechanics **242** (1992), 279–298.
- [4] T. M. Apostol, *An Elementary View of Euler's Summation Formula*, The American Mathematical Monthly **106** (1999), no. 5, 409–418.
- [5] W. S. Barge and W. J. Stewart, *Autonomous solution methods for large Markov chains*, to appear.
- [6] R. Barrett, Berry M, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, Ch. Romine, and H. Van der Vorst, *Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods*, SIAM, Philadelphia, USA, 1994.
- [7] M. W. Berry, Z. Drmac, and E. R. Jessup, *Matrices, vector spaces, and information retrieval*, SIAM Review **41** (1999), no. 2, 335–362.
- [8] J. L. Bona, *Convergence of periodic wavetrains in the limit of large wavelength*, Applied Scientific Research **37** (1981), no. 1-2, 21–30.
- [9] J. L. Bona, M. Chen, and J.-C. Saut, *Boussinesq equations and other systems for small-amplitude long waves in nonlinear dispersive media. I: Derivation and linear theory*, Journal of Nonlinear Science **12** (2002), no. 4, 283–318.
- [10] ———, *Boussinesq equations and other systems for small-amplitude long waves in nonlinear dispersive media: II. The nonlinear theory*, Nonlinearity **17** (2004), no. 3, 925–952.
- [11] C. Brezinski, *Accélération de suites à convergence logarithmique*, Comptes Rendus de L' Académie des Sciences, Series A **273** (1971), 727–730.
- [12] ———, *Généralisations de la transformation de Shanks, de la table de Padé et de l' ϵ -algorithme*, Calcolo **12** (1975), no. 4, 317–360.
- [13] ———, *Accélération de la convergence en analyse numérique*, Lecture Notes in Mathematics, no. 584, Springer-Verlag, Berlin, Germany, 1977.
- [14] ———, *A General Extrapolation Algorithm*, Numerische Mathematik **35** (1980), no. 2, 175–187.
- [15] C. Brezinski and M. Redivo-Zaglia, *Extrapolation Methods: Theory and Practice*, North-Holland, Amsterdam, Holland, 1991.
- [16] ———, *The PageRank vector: Properties, computation, approximation, and acceleration*, SIAM Journal on Matrix Analysis and Applications **28** (2006), no. 2, 551–575.
- [17] ———, *The Simplified Topological ϵ -Algorithms for Accelerating Sequences in a Vector Space*, SIAM Journal on Scientific Computing **36** (2014), no. 5, A2227–A2247.
- [18] ———, *Extrapolation and Rational Approximation: The Works of the Main Contributors*, Springer Nature Switzerland AG, Cham, Switzerland, 2020.

Bibliografía

- [19] S. Brin and L. Page, *The anatomy of a large-scale hypertextual Web search engine*, Computer Networks and ISDN Systems **30** (1998), no. 1-7, 107–117.
- [20] R. Bulirsch and J. Stoer, *Fehlerabschätzungen und Extrapolation mit rationalen Funktionen bei Verfahren vom Richardson-Typus*, Numerische Mathematik **6** (1964), no. 1, 413–427.
- [21] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods. Fundamentals in Single Domains*, Springer, Berlin Heidelberg, Germany, 2006.
- [22] H. Chen, *Long-period limit of nonlinear dispersive waves: the BBM-equation*, Differential Integral Equations **19** (2006), no. 4, 463–480.
- [23] N. G. de Bruijn, *Asymptotic Methods in Analysis*, North Holland, Amsterdam, Holland, 1970.
- [24] A. Durán, D. Dutykh, and D. Mitsotakis, *On the multi-symplectic structure of Boussinesq-type systems. I: Derivation and mathematical properties*, Physica D: Nonlinear Phenomena **388** (2019), 10–21.
- [25] ———, *On the multi-symplectic structure of Boussinesq-type systems. II: Geometric discretization*, Physica D: Nonlinear Phenomena **397** (2019), 1–16.
- [26] T. Fessler, W. F. Ford, and D. A. Smith, *HURRY: An Acceleration Algorithm for Scalar Sequences and Series*, ACM Transactions on Mathematical Software **9** (1983), no. 3, 346–354.
- [27] E. Gekeler, *On the solution of systems of equations by the epsilon algorithm*, Mathematics of Computation **26** (1972), no. 118, 427–436.
- [28] G. H. Golub and Ch. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, USA, 2013.
- [29] P. R. Graves-Morris, *Vector valued rational interpolants I*, Numerische Mathematik **42** (1983), no. 3, 331–348.
- [30] ———, *A review of Padé methods for the acceleration of convergence of a sequence of vectors*, Applied Numerical Mathematics **15** (1994), no. 2, 153–174.
- [31] E. Hairer, S. P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I*, 2nd ed., Springer, Berlin-Heidelberg, Germany, 2000.
- [32] T. Håvie, *Remarks on a unified theory for classical and generalized interpolation and extrapolation*, BIT Numerical Mathematics **21** (1981), no. 4, 465–474.
- [33] P. Hillion, *Méthode d’Aitken itérée pour les suites oscillantes d’approximations*, Comptes Rendus de L’Académie des Sciences, Series A (1975), no. 280, 1701–1704.
- [34] K. Jbilou and H. Sadok, *Some results about vector extrapolation methods and related fixed-point iterations*, Journal of Computational and Applied Mathematics **36** (1991), no. 3, 385–398.
- [35] ———, *Vector extrapolation methods. applications and numerical comparison*, Journal of Computational and Applied Mathematics **122** (2000), no. 1-2, 149–165.
- [36] M. Kaiser, R. L. Jack, and J. Zimmer, *Acceleration of convergence to equilibrium in Markov chains by breaking detailed balance*, Journal of Statistical Physics **168** (2017), no. 2, 259–287.
- [37] A. N. Langville and C. D. Meyer, *A survey of eigenvector methods for Web information retrieval*, SIAM Review **47** (2005), no. 1, 135–161.
- [38] ———, *Google’s Pagerank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, USA, 2012.
- [39] H. Le Ferrand, *The quadratic convergence of the topological epsilon algorithm for systems of nonlinear equations*, Numerical Algorithms **3** (1992), no. 1, 273–283.
- [40] D. Levin, *Development of non-linear transformations for improving convergence of sequences*, International Journal of Computer Mathematics **3** (1972), no. 1-4, 371–388.
- [41] E. Lombardi, *Oscillatory Integrals and Phenomena Beyond all Algebraic Orders*, Springer, Berlin-Heidelberg, Germany, 2000.

- [42] J. H. Mathews and K. D. Fink, *Métodos Numéricos con MATLAB*, 3 ed., Prentice-Hall, Madrid, Spain, 2000.
- [43] J. B. McLeod, *A note on the ε -algorithm*, *Computing* **7** (1971), no. 1-2, 17–24.
- [44] C. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, USA, 2001.
- [45] E. N. Neville, *Iterative interpolation*, *Journal of the Indian Mathematical Society* **20** (1934), 87–120.
- [46] E. P. Petrov and N. L. Kharina, *Markov processes in Image Processing*, *International Conference Informantion Technologies in Business and Industry*, IOP Conference Series: Journal of Physics **1015** (2018), 032059.
- [47] L. F. Richardson, *The Approximate Arithmetical Solution by Finite Differences of Physical Problems involving Differential Equations, with an Application to the Stresses in a Masonry Dam*, *Philosophical Transactions of the Royal Society of London. Series A* **210** (1910), 307–357.
- [48] ———, *The Deferred Approach to the Limit, Part I. Single Lattice.*, *Philosophical Transactions of the Royal Society of London. Series A* **226** (1927), 299–349.
- [49] L. F. Richardson and J. A. Gaunt, *The Deferred Approach to the Limit, Part II. Interpenetrating Lattices*, *Philosophical Transactions of the Royal Society of London. Series A* **226** (1927), 349–361.
- [50] W. Romberg, *Vereinfachte numerische integration*, *Det Kongelige Norske Videnskabers Selskab Fordhandling* **28** (1955), no. 7, 30–36.
- [51] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, USA, 2003.
- [52] E. Salinelli and F. Tomarelli, *Discrete Dynamical Models*, Springer, Cham, Switzerland, 2014.
- [53] D. Shanks, *Non-linear transformations of divergent and slowly convergent sequences*, *Journal of Mathematics and Physics* **34** (1955), no. 1-4, 1–42.
- [54] J. Shen, T. Tang, and L. L. Wang, *Spectral Methods. Algorithms, Analysis and Applications*, vol. 41, Springer, Berlin- Heidelberg, Germany, 2011.
- [55] A. Sidi, *Extrapolation vs. projection methods for linear systems of equations*, *Journal of Computational and Applied Mathematics* **22** (1988), 71–88.
- [56] ———, *Practical Extrapolation Methods: Theory and Applications*, Cambridge University Press, Cambridge, UK, 2003.
- [57] ———, *Vector extrapolation methods with applications to solution of large systems of equations and to PageRank computations*, *Computers & Mathematics with Applications* **56** (2008), no. 1, 1–24.
- [58] ———, *Methods for acceleration of convergence (extrapolation) of vector sequences*, *Wiley Encyclopedia of Computer Science and Engineering*, B. W. Wah (ed), John Wiley & Sons, 2009, pp. 1828–1846.
- [59] ———, *Vector Extrapolation Methods with Applications*, SIAM, Philadelphia, USA, 2017.
- [60] A. Sidi, W. F. Ford, and D. A. Smith, *Acceleration of convergence of vector sequences*, *SIAM Journal of Numerical Analysis* **23** (1986), no. 1, 178–196.
- [61] A. Sidi and Y. Shapira, *Upper bounds for convergence rates of vector extrapolation methods on linear systems with initial iterations*, *Tech. Report 701*, Computer Science Department, Technion-Israel Institute of Technology, 1991.
- [62] ———, *Upper bounds for convergence rates of acceleration methods with initial iterations*, *Numerical Algorithms* **18** (1998), no. 2, 113–132.
- [63] D. A. Smith and W. F. Ford, *Numerical Comparison of Nonlinear Convergence Accelerators*, *Mathematics of Computation* **38** (1982), no. 158, 481–499.
- [64] W. J. Stewart, *MARCA_Models: A collection of Markov chain models*, https://people.engr.ncsu.edu/billy/MARCA_Models.html.
- [65] ———, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, USA, 1994.

Bibliografía

- [66] ———, *Probability, Markov Chains, Queues, and Simulation*, Princeton University Press, Princeton, USA, 2009.
- [67] J. J. Sylvester, *On the relation between the minor determinants of linearly equivalent quadratic functions*, Philosophical Magazine **1** (1851), 295–305, Citado en: A. G. Akritas, E. G. Akritas, G. I. Malaschonok, Various proofs of Sylvester's (determinant) identity, Math. Comput. Simul., **42** (1996) 585–593.
- [68] Ll. Trefethen, *Spectral Methods in MATLAB*, SIAM, Philadelphia, USA, 2000.
- [69] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monographs on Applied and Computational Mathematics, no. 13, Cambridge University Press, Cambridge, UK, 2003.
- [70] E. J. Weniger, *Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series*, Computer Physics Reports **10** (1989), no. 5–6, 189 – 371.
- [71] J. Wimp, *Sequence Transformations and their Applications*, Academic Press, New York, EE.UU, 1981.
- [72] P. Wynn, *On a Device for Computing the $e_m(s_n)$ Transformation*, Mathematics of Computation **10** (1956), no. 54, 91–96.
- [73] ———, *On a Procrustean technique for the numerical transformation of slowly convergent sequences and series*, Mathematical Proceedings of the Cambridge Philosophical Society **52** (1956), no. 4, 663–671.
- [74] ———, *Acceleration Techniques for Iterated Vector and Matrix Problems*, Mathematics of Computation **16** (1962), no. 79, 301–322.
- [75] ———, *General Purpose Vector epsilon Algorithm Algol Procedures*, Numerische Mathematik **6** (1964), no. 1, 22–36.
- [76] J. Yang, *Nonlinear Waves in Integrable and Nonintegrable Systems*, SIAM, Philadelphia, USA, 2010.