



Universidad
de La Laguna

Integración numérica

Numerical integration

Adrián Martín Rodríguez

Trabajo de Fin de Grado

Facultad de Ciencias

Sección de Matemáticas

Universidad de La Laguna

La Laguna, 11 de septiembre de 2015

Ruymán Cruz Barroso, con N.I.F. 78.677.674-X, profesor Ayudante Doctor de Universidad adscrito al Departamento de Análisis Matemático de la Universidad de La Laguna

C E R T I F I C A

Que la presente memoria titulada:

“Integración numérica”

ha sido realizada bajo su dirección por D. **Adrián Martín Rodríguez**, con N.I.F. 54.064.375-T.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos, firman la presente en La Laguna a 11 de septiembre de 2015

Agradecimientos

Quisiera agradecer a mi familia sus múltiples muestras de apoyo, propulsoras de aliento continuo, tenacidad y esfuerzo, a lo largo de esta bonita pero intensa aventura. Asimismo, quisiera trasladar mi más sincero reconocimiento al profesor D. Ruymán Cruz Barroso por su inestimable colaboración en esta Memoria.

Resumen

La *integración numérica* es el estudio de cómo el valor numérico de una integral definida puede ser encontrado mediante una serie de procedimientos o reglas. Aunque hablamos de un problema cuyos orígenes son tan antiguos como los propios de las Matemáticas, esta disciplina sigue teniendo plena vigencia e interés, dado que un gran número de procesos numéricos de la *Matemática Aplicada* tiene entre sus distintas etapas el cálculo de integrales. Uno de los métodos más usuales y de gran sencillez conceptual son las denominadas *fórmulas de cuadratura*, en las que una integral es aproximada por una combinación lineal de valores del integrando. El objetivo de este trabajo es, por un lado, el de analizar y caracterizar algunas fórmulas de cuadratura conocidas en la literatura, como método de aproximación de integrales definidas en intervalos de la recta real, desde un punto de vista teórico, y por otro lado, el de considerar ciertos aspectos computacionales para un cálculo eficiente de estas reglas.

El primer Capítulo de esta Memoria está dedicado a la introducción y al estudio de dos de los conceptos de mayor importancia dentro de la *Matemática Aplicada*, y que están íntimamente relacionados con la Integración Numérica: la *ortogonalidad* y la *interpolación polinómica*. Partiendo de una función peso positiva definida en el intervalo de integración, se establecerá primero cómo construir una familia de polinomios ortogonales con respecto a ésta, para pasar a continuación a estudiar sus propiedades más relevantes: ley de recurrencia a tres términos, propiedades de localización y entrelazamiento de ceros, el núcleo reproductor y matrices de Jacobi asociadas. Algunas funciones peso particulares que dan lugar a familias de polinomios ortogonales muy conocidas en la literatura (polinomios clásicos) serán analizadas de manera particular. Finalmente, se considerará el siguiente problema de interpolación polinómica: dados n nodos distintos de la recta real y n cantidades arbitrarias, determinar el polinomio de grado mínimo que toma en esos nodos los valores dados. Una fórmula explícita para el cómputo de este polinomio será estudiada: la fórmula de interpolación de Lagrange.

El segundo Capítulo está dedicado a la construcción y caracterización de fórmulas de cuadratura, las cuales se tratan de una combinación lineal de valores del integrando, estudiadas desde un punto de vista teórico (obtención de nodos y pesos) y computacional. El elemento clave aquí es el *Teorema de Aproximación de Weierstrass*, que establece que toda función continua definida en un compacto de la recta real puede ser aproximada uniformemente por polinomios. Introduciremos previamente las *fórmulas de Newton-Cotes*, las cuales son unas fórmulas de cuadratura muy particulares, en las que los nodos se toman equiespaciados. También serán estudiadas las *fórmulas de tipo interpolatorio*, en las que partiendo de un conjunto de nodos arbitrarios distintos entre sí y dentro del intervalo de integración, se construye una fórmula de cuadratura considerando como aproximación de una integral definida con respecto a una función peso, la integral del polinomio interpolador basado en esos nodos (supondremos por lo tanto que los monomios son integrables con respecto a la función peso de partida). A continuación consideraremos las denominadas *fórmulas Gaussianas*, en las que una elección apropiada de nodos permitirá incrementar el

dominio de validez de las fórmulas de cuadratura hasta ser maximal (en nuestra situación, parece apropiado considerar el criterio de máxima exactitud polinómica, es decir, que la regla sea exacta en espacios de polinomios con grado máximo posible). Esta elección particular de nodos será precisamente la de tomar los ceros de los polinomios ortogonales con respecto a la función peso de partida. Además, se comprueba en este caso que los pesos de estas fórmulas de cuadratura son positivos, garantizando por tanto la estabilidad y la convergencia en un amplio espacio de funciones. Finalmente consideraremos también las *fórmulas de tipo Gauss*, en las que un extremo del intervalo de integración (Gauss-Radau) o ambos extremos (Gauss-Lobatto) son también elegidos como nodos de la cuadratura. Como en todo proceso numérico, se llevará a cabo un análisis del error, así como un estudio desde un punto de vista computacional, donde el cálculo de este tipo de fórmulas se transforma en un problema de cálculo de autovalores y autovectores de las matrices de Jacobi asociadas.

Esta Memoria contiene finalmente un Anexo donde se han implementado en lenguaje `Matlab` diversos códigos que se han empleado durante el desarrollo de este trabajo para ilustrar numéricamente algunas de las cuestiones abordadas.

Palabras clave: Polinomios Ortogonales, Interpolación Polinómica, Fórmulas de Newton-Cotes, Fórmulas de tipo interpolatorio, Fórmulas de tipo Gauss, Estimaciones del error.

Abstract

Numerical integration is the study of how the numerical value of a definite integral can be found by following a sequence of procedures or rules. Although we speak of a problem whose origins are as old as the own of Mathematics, this discipline continues in full force and interest because a great number of numerical procedures in *Applied Mathematics* have the calculation of integrals in their different stages. One of the most common methods and of high conceptual simplicity are the so called *quadrature formulas*, where an integral is approximated by a linear combination of values of the integrand. The aim of this work is, by one hand, to analyze and to characterize some quadrature formulas already known in the literature, as a method for the approximation of a definite integral over intervals of the real line, from a theoretical point of view, and by other hand, to consider certain computational aspects for the efficient computation of such rules.

The first Chapter of this work is devoted to the introduction and the study of two of the most important concepts in *Applied Mathematics*, closely interrelated with Numerical Integration: *orthogonality* and *polynomial interpolation*. Starting from a (positive) weight function defined on the interval of integration, we will establish first how to construct a family of orthogonal polynomials with respect to it, and next we will study the most relevant properties: the three-term recurrence relation, location and interlacing properties of their zeros, the reproducing kernel and the associated Jacobi matrices. Some particular elections of weight functions that give rise to particular families of orthogonal polynomials very well known in the literature (classical polynomials) will be analyzed in a more particular way. Finally, we will consider the following problem of polynomial interpolation: given n distinct nodes of the real axis and n values, to determine the polynomial of minimum degree that takes these vales at these nodes. An explicit formula for the computation of this polynomial will be studied: the Lagrange interpolation formula.

The second Chapter is dedicated to the construction and characterization of quadrature formulas, which are just a linear combination of values of the integrand, studied from a theoretical point of view (obtention of nodes and weights) and computationally. The key fact here is the *Weierstrass Approximation Theorem*, that states that every continuous function on a compact of the real line can be uniformly approximated by polynomials. We will introduce first the *Newton-Cotes's formula*, which are a very particular set of quadrature formulas, where the nodes are equally distributed. *Interpolatory-type* rules will be also studied. Here, starting from an arbitrary set of different nodes inside the interval of integration, a quadrature formula is constructed by considering as the approximation of a definite integral with respect to a weight function, the integral of the interpolant polynomial based on these nodes (so, we will assume that the monomials are integrable with respect to the weight function). Next, we will consider the so called *Gaussian formulas*, where an appropriate election of the nodes will increase the domain of exactness to be maximal (in our situation, it seems appropriate to consider the highest polynomial accuracy criterion, that is, that the rule is exact in polynomial spaces of highest possible degree). This particular election of the nodes will be precisely to take the zeros of orthogonal poly-

nomials with respect to the starting weight function. Moreover, it can be proved in this case that the weights of these quadrature rules are positive, so we are ensuring stability and convergence for a large function space. Finally, we will consider also the *Gauss-type* quadrature formulas, where an endpoint of the interval of integration (Gauss-Radau) or both endpoints (Gauss-Lobatto) are also taken as nodes of the quadrature. Like in every numerical process, an analysis of the error will be carried out, along with a study from a computational point of view, where the obtention of these kind of formulas is transformed to an eigenvalue and eigenvector finding problem of the associated Jacobi matrices.

This work also contains an Asside at the end where several codes have been implemented in `Matlab` language to be used along all the text to illustrate numerically some of the addressed issues.

Keywords: Orthogonal Polynomials, Polynomial Interpolation, Newton-Cotes rules, Interpolatory-type quadrature formulas, Gauss-type quadrature formulas, Error estimates.

Índice general

1. Polinomios ortogonales e interpolación polinómica	3
1.1. Polinomios ortogonales	3
1.1.1. Introducción. Construcción de polinomios ortogonales	3
1.1.2. Ceros de polinomios ortogonales	6
1.1.3. Ley de recurrencia	7
1.1.4. Fórmula de Christoffel-Darboux. Fórmula confluyente	8
1.1.5. Matrices de Jacobi	10
1.1.6. Entrelazamiento de ceros	11
1.1.7. Familias particulares de polinomios ortogonales	12
1.2. Interpolación polinómica	16
1.3. Ejemplos numéricos	19
1.3.1. Polinomios de Chebyshev de primera especie	19
1.3.2. Polinomios de Chebyshev de segunda especie	19
1.3.3. Polinomios de Legendre	20
1.3.4. Resultados numéricos y análisis de resultados	21
2. Integración numérica	23
2.1. Fórmulas de Newton-Cotes	24
2.2. Fórmulas de cuadratura de tipo interpolatorio	31
2.3. Fórmulas de cuadratura Gaussianas	34
2.3.1. Fórmulas de cuadratura con máximo grado de precisión	34
2.3.2. Fórmulas de cuadratura con nodos preasignados	40
2.3.3. Resultados numéricos y análisis de resultados	41
3. Conclusiones y futuras líneas de investigación	46
A. Programación en Matlab	49
A.1. Polinomio interpolador de Lagrange	49
A.2. Problema de autovalores, autovectores y pesos para los polinomios de Chebyshev de primera especie	50
A.2.1. Método 1	50
A.2.2. Método 2	51

A.3. Problema de autovalores, autovectores y pesos para los polinomios de Chebyshev de segunda especie	52
A.4. Problema de autovalores, autovectores y pesos para los polinomios de Legendre	53
A.5. Integración numérica: fórmulas de Newton-Cotes	53
Bibliografía	56

Índice de símbolos más frecuentes

\mathbb{P}	Espacio vectorial real de polinomios con coeficientes reales
\mathbb{P}_n	Subespacio vectorial de polinomios con grado menor o igual que n
$\deg(p(x))$	Grado del polinomio $p(x)$
a_n	Coficiente director del n -ésimo polinomio ortogonal
A_n	Coficiente director del n -ésimo polinomio ortonormal
$\{p_k\}_{k=0}^{\infty}$	Familia de polinomios ortogonales
$\{P_k\}_{k=0}^{\infty}$	Familia de polinomios ortonormales
$\{Q_k\}_{k=0}^{\infty}$	Familia de polinomios ortogonales mónicos
$\{p_k^{(\alpha,\beta)}\}_{k=0}^{\infty}$	Familia de polinomios ortogonales de Jacobi
$\{p_k^{(0,0)}\}_{k=0}^{\infty}$	Familia de polinomios ortogonales de Legendre
$\{t_k\}_{k=0}^{\infty}$	Familia de polinomios ortogonales de Chebyshev de primera especie
$\{u_k\}_{k=0}^{\infty}$	Familia de polinomios ortogonales de Chebyshev de segunda especie
$\{h_k\}_{k=0}^{\infty}$	Familia de polinomios ortogonales de Hermite
$\{l_k\}_{k=0}^{\infty}$	Familia de polinomios ortogonales de Laguerre
$\{P_k^{(\alpha,\beta)}\}_{k=0}^{\infty}$	Familia de polinomios ortonormales de Jacobi
$\{P_k^{(0,0)}\}_{k=0}^{\infty}$	Familia de polinomios ortonormales de Legendre
$\{T_k\}_{k=0}^{\infty}$	Familia de polinomios ortonormales de Chebyshev de primera especie
$\{U_k\}_{k=0}^{\infty}$	Familia de polinomios ortonormales de Chebyshev de segunda especie
$\{H_k\}_{k=0}^{\infty}$	Familia de polinomios ortonormales de Hermite
$\{L_k\}_{k=0}^{\infty}$	Familia de polinomios ortonormales de Laguerre
\mathbb{N}	Conjunto de los números naturales
\mathbb{Z}	Conjunto de los números enteros
\mathbb{Q}	Conjunto de los números racionales
\mathbb{R}	Conjunto de los números reales
$\langle f, g \rangle_{\sigma}$	Producto interior de las funciones f y g con respecto a la función peso σ
$\delta_{k,s}$	Función delta de Kronecker
$\mathbb{K}_n(\xi, w)$	n -ésimo núcleo reproductor
c_k	Momento de orden k
$\beta_{n,k}$	Coficientes de Fourier
Δ_n	Matriz de Hankel de orden $n + 1$
$w(x)$	Polinomio nodal
η_n, ζ_n	Coficientes de la matriz de Jacobi
J_n	Matriz de Jacobi de orden n
$x_{j,n}$	Cero j -ésimo del polinomio P_n (autovalor de J_n)
$\tilde{v}_{j,n}$	Autovector normalizado asociado al autovalor $x_{j,n}$ de J_n
$\lambda_{j,n}$	Peso j -ésimo de la fórmula Gaussiana de n puntos

Motivación y Objetivos

La *Integración numérica* se enmarca dentro de la *Teoría de Aproximación*, un área de la *Matemática Aplicada* que ha experimentado un crecimiento notable en los últimos años; fundamentalmente desde la aparición de los ordenadores, debido a su importancia en la resolución aproximada de problemas aplicados en diferentes áreas científicas. Más concretamente, es un área de la Matemática con vinculación directa al desarrollo social y tecnológico.

Dada la importancia de este tema por sus numerosas aplicaciones, y el hecho de que éste se haya estudiado, por razones obvias de tiempo, solamente en una parte de un tema de la asignatura *Métodos Numéricos II* del actual Grado en Matemáticas de la Universidad de La Laguna, se motiva este *Trabajo de Fin de Grado* con el fin de abordarlo de una manera mucho más profunda, rigurosa y experimental.

El objetivo de este trabajo es la aproximación numérica del valor de la integral de una función con respecto a una función peso definida sobre un intervalo de la recta real. Para este propósito se ha llevado a cabo previamente un estudio pormenorizado de los polinomios ortogonales y de la interpolación polinómica, que ha permitido concluir con la construcción y caracterización de las llamadas “fórmulas de cuadratura de tipo interpolatorio” y “de tipo Gauss” (Gauss, Gauss-Radau y Gauss-Lobatto). Esta Memoria está distribuida de tal forma que recoge un conjunto de conocimientos cuidadosamente estudiados y auto-contenidos en el transcurso de su desarrollo.

El cálculo de la integral definida de una función continua sobre un intervalo acotado del eje real puede ser difícil de calcular. Por ello se recurre a utilizar el Teorema de Aproximación de Weierstrass, que permite aproximar uniformemente la función continua del integrando definida en ese compacto de \mathbb{R} por un polinomio. Así, si el integrando se factoriza como el producto de una función “suave” y de otra que contenga las posibles singularidades próximas al intervalo de integración, se podrá aproximar la integral pedida por la integral de un polinomio siempre que admitamos como hipótesis de partida que las integrales de los monomios (momentos) sean conocidas y fácilmente computables.

Este trabajo comienza con un estudio detallado sobre la construcción de familias de polinomios ortogonales con respecto a una función peso (positiva) definida en un intervalo del eje real. También se han considerado las propiedades más relevantes: ley de recurrencia, localización y entrelazamiento de ceros, fórmula de Christoffel-Darboux para el núcleo reproductor y la introducción de las matrices de Jacobi asociadas. Asimismo, se ha particularizado esta Teoría con diferentes familias de polinomios ortogonales clásicas:

Jacobi, Legendre, Laguerre, Chebyshev, Hermite,... cuyas propiedades vienen estudiadas e implementadas mediante procedimientos de cálculo en lenguaje Matlab. También jugará un papel crucial previo a la construcción de fórmulas de cuadratura la interpolación polinómica. Para ello estudiaremos una fórmula explícita para el polinomio interpolador conocida como fórmula de Lagrange.

En cuanto a las aproximaciones de las integrales, profundizaremos en las fórmulas de cuadratura, las cuales se tratan de una combinación lineal de valores del integrando. Consideraremos previamente las fórmulas de Newton-Cotes, las cuales son unos métodos de integración numérica que tienen nodos equiespaciados; se tratan por tanto de toda una familia general de métodos según el grado del polinomio de interpolación que se considere. También serán analizadas las fórmulas de tipo interpolatorio en las que partiendo de un conjunto de nodos arbitrarios distintos entre sí y dentro del intervalo de integración se construye una fórmula de cuadratura considerando como aproximación de una integral definida con respecto a una función peso, la integral del polinomio interpolador basado en esos nodos. Veremos también el papel fundamental que juegan los polinomios ortogonales en esta Teoría, esto es, la elección particular de nodos en fórmulas de cuadratura tomando los ceros de los correspondientes polinomios ortogonales propician que se obtengan las fórmulas Gaussianas que son aquellas que poseen el máximo dominio de validez. Dicho en otras palabras, una elección particular de nodos elegidos en el intervalo de integración determina una fórmula de cuadratura que es exacta en un subespacio de polinomios con dimensión máxima alcanzable. Finalmente, en esta línea se introducirán los dos casos clásicos relacionados con la preasignación de nodos en una fórmula de cuadratura: las fórmulas de Gauss-Radau, donde uno de los extremos del intervalo de integración se toma como nodo de la fórmula, y las de Gauss-Lobatto, donde se fijarán ambos extremos.

La Teoría anterior es aplicada y extendida en multitud de problemas de diversos ámbitos, tales como Matemáticas, Física e Ingeniería. Son bien conocidos los trabajos relacionados con las fórmulas de cuadratura sobre la circunferencia unidad, llamadas fórmulas Szegő, así como la preasignación de nodos. Asimismo, caben resaltar las cuestiones relacionadas con la ortogonalidad matricial (familias de polinomios ortogonales matriciales que satisfacen ecuaciones diferenciales de segundo orden) y la caracterización de las fórmulas de cuadratura con funciones ortogonales racionales. Otras líneas de investigación se centran en las múltiples aplicaciones en la modelización de los sistemas cuánticos relativistas (ecuación de Dirac,...) como no relativistas (ecuación de Schrödinger) y problemas de equilibrio electrostático (con potencial logarítmico), entre otras muchas cuestiones.

Capítulo 1

Polinomios ortogonales e interpolación polinómica

El objetivo de este trabajo es la aproximación de la integral definida de una función con respecto a una función peso definida sobre un intervalo de la recta real. El elemento clave de esta Teoría será el Teorema de Aproximación de Weierstrass, que establece que toda función continua en un compacto puede ser aproximada uniformemente por polinomios.

1.1. Polinomios ortogonales

En esta sección trataremos nociones básicas acerca del concepto de ortogonalidad.

1.1.1. Introducción. Construcción de polinomios ortogonales

Por el Teorema de Aproximación de Weierstrass, la integral de una función se aproximará por la integral de un polinomio. Usualmente, este integrando se factoriza como el producto de una función “suave” f y una “función peso” σ que contendrá las posibles singularidades próximas al intervalo de integración $[a, b]$. La función peso σ debe verificar:

- $\sigma > 0$ e integrable $[a, b]$,
- $x^n \cdot \sigma(x)$ es integrable en $[a, b]$.

Sin pérdida de generalidad podemos asumir que trabajamos en el intervalo $[-1, 1]$. En caso contrario, aplicaríamos primero la transformación lineal

$$T: \begin{array}{ll} [a, b] & \longrightarrow [-1, 1] \\ x & \longmapsto t = -1 + 2\frac{x-a}{b-a}. \end{array}$$

Consecuentemente se puede definir el operador integral asociado a la función peso σ como

$$I_\sigma(f) := \int_{-1}^1 f(x)\sigma(x)dx.$$

Para nuestro propósito será necesario introducir el concepto de ortogonalidad.

Definición 1.1.1. *El producto interior de las funciones f y g inducido por la función peso σ se define como*

$$\langle f, g \rangle_\sigma := \int_{-1}^1 f(x)g(x)\sigma(x)dx, \quad (1.1)$$

el cual verifica:

- $\langle f, f \rangle_\sigma \geq 0$ y $\langle f, f \rangle_\sigma = 0 \Leftrightarrow f \equiv 0$,
- $\langle f, g \rangle_\sigma = \langle g, f \rangle_\sigma$,
- $\langle \lambda f + \mu h, g \rangle_\sigma = \lambda \langle f, g \rangle_\sigma + \mu \langle h, g \rangle_\sigma, \quad \forall \lambda, \mu \in \mathbb{R}$.

Nótese que en el producto interior (1.1) no se toma conjugación dado que estamos considerando medidas con soporte en intervalos de la recta real y funciones real evaluadas. Precisemos que todo producto interior induce una norma, en este trabajo se utilizará la norma euclídea $\|f\|_{2,\sigma}^2 = \langle f, f \rangle_\sigma$.

Definición 1.1.2. *Las funciones f y g son ortogonales en $[-1, 1]$ si verifican:*

- *el producto fg es σ -integrable,*
- $\int_{-1}^1 f(x)g(x)\sigma(x)dx = 0$.

Definición 1.1.3. *Una función f ortogonal en $[-1, 1]$ con respecto a σ se dice que es ortonormal si su norma es uno, es decir, $1 = \|f\|_{2,\sigma}^2 = \langle f, f \rangle_\sigma = \int_{-1}^1 f^2(x)\sigma(x)dx$.*

Al conjunto ordenado $\{1, x, x^2, \dots, x^n\}$ con $n \in \mathbb{N}$ se le puede aplicar el proceso de ortogonalización de Gram-Schmidt, permitiendo construir una familia de polinomios ortogonales $\{p_k\}_{k=0}^n$ asociados a la función peso σ en el intervalo $[-1, 1]$, verificando:

- $\deg(p_k) = k$;
- $\langle p_k, p_s \rangle_\sigma = \delta_{k,s} = \begin{cases} 0 & \text{si } k \neq s, \\ 1 & \text{si } k = s, \end{cases}$ (función delta de Kronecker).

Si el proceso de ortogonalización se realiza para todo $n \in \mathbb{N}$ se obtiene una familia de polinomios ortogonales $\{p_k\}_{k=0}^\infty$. Obsérvese que esta familia es esencialmente única salvo factor multiplicativo λ_n , esto es, como $p_n \perp_\sigma \{1, x, x^2, \dots, x^{n-1}\}$ se tiene que $\langle \lambda_n p_n, x^s \rangle_\sigma = \lambda_n \langle p_n, x^s \rangle_\sigma = 0, \quad \forall s = 0, 1, \dots, n-1, \quad \lambda_n \neq 0$. La constante λ_n anterior puede elegirse adecuadamente para obtener dos familias particulares de polinomios ortogonales de especial relevancia:

- familia mónica: si los polinomios son todos mónicos, esto es, el coeficiente que multiplica al monomio de mayor grado vale uno. Esta familia particular será denotada por $\{Q_k\}_{k=0}^{\infty}$;
- familia ortonormal: para todo $k \geq 1$ se cumple que $\|P_k\|_{2,\sigma} = \sqrt{\langle P_k, P_k \rangle_{\sigma}} = 1$. Esta familia se denotará por $\{P_k\}_{k=0}^{\infty}$.

Teniendo presente que nuestro objetivo será el de evaluar la integral con respecto a σ de un polinomio, introducimos las siguientes definiciones:

Definición 1.1.4. Para cada $k \in \mathbb{N}$ se define el momento k -ésimo como:

$$c_k := \int_{-1}^1 x^k \sigma(x) dx. \quad (1.2)$$

En esta Memoria se asumirá, como es usual, que los momentos c_k de la función peso son finitos y pueden computarse fácilmente. También podemos asumir, sin pérdida de generalidad, que $c_0 = 1$, es decir, que σ es una función peso de probabilidad.

Definición 1.1.5. A partir de la sucesión de momentos se define la matriz de Hankel como:

$$\Delta_n := \begin{pmatrix} c_0 & c_1 & \cdots & c_n \\ c_1 & c_2 & \cdots & c_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n+1} & \cdots & c_{2n} \end{pmatrix}. \quad (1.3)$$

Veamos a continuación que las matrices de Hankel son no singulares.

Proposición 1.1.6. Para todo $n \geq 1$ se cumple que $\det \Delta_n \neq 0$, siendo Δ_n la matriz de Hankel definida en (1.3).

Demostración. Construyamos el sistema homogéneo de $(n+1)$ -ecuaciones con $(n+1)$ -incógnitas a_0, a_1, \dots, a_n :

$$\begin{cases} a_0 c_0 + a_1 c_1 + \cdots + a_n c_n = 0, \\ a_0 c_1 + a_1 c_2 + \cdots + a_n c_{n+1} = 0, \\ \vdots \\ a_0 c_n + a_1 c_{n+1} + \cdots + a_n c_{2n} = 0. \end{cases} \quad (1.4)$$

Supóngase que $\det \Delta_n = 0$, luego, el sistema homogéneo admite solución no trivial. Si en el sistema (1.4) sustituimos la expresión de los momentos (1.2) obtenemos:

$$\begin{cases} \int_{-1}^1 \sigma(x) [a_0 + a_1 x + \cdots + a_n x^n] dx = 0, \\ \int_{-1}^1 \sigma(x) x [a_0 + a_1 x + \cdots + a_n x^n] dx = 0, \\ \vdots \\ \int_{-1}^1 \sigma(x) x^n [a_0 + a_1 x + \cdots + a_n x^n] dx = 0. \end{cases} \quad (1.5)$$

Multiplicando las ecuaciones del sistema (1.5) respectivamente por a_0, a_1, \dots, a_n y sumándo-
las se obtiene $\int_{-1}^1 \sigma(x)[a_0 + a_1x + \dots + a_nx^n]^2 dx = 0$. Como hemos asumido que $\sigma(x) \geq 0$,
concluimos que $a_0 + a_1x + \dots + a_nx^n = 0$, es decir, $a_i = 0$, para todo $i = 0, \dots, n$, lo cual
es una contradicción. Por lo tanto, se concluye que $\det \Delta_n \neq 0$. □

El resultado anterior establece que toda función peso σ en $[-1, 1]$ tiene asociada una
sucesión de momentos, para las cuales las matrices de Hankel son no singulares. Esto
permite construir familias de polinomios ortogonales. Nótese además que de (1.4) se sigue
que el sistema

$$\Delta_{n-1} \cdot (a_0 \ a_1 \ a_2 \ \dots \ a_{n-2} \ a_{n-1})^T = -a_n \cdot (c_n \ c_{n+1} \ c_{n+2} \ \dots \ c_{2n-2} \ c_{2n-1})^T.$$

admite solución única, probando la existencia de la familia de polinomios ortogonales
para la función peso σ en términos de una constante multiplicativa a_n que permite, en
particular, elegir familias mónicas u ortonormales.

Después de haber estudiado cómo construir familias de polinomios ortogonales con respecto
a una función peso σ , veamos algunas propiedades que cumplen estas familias.

1.1.2. Ceros de polinomios ortogonales

Los ceros de familias de polinomios ortogonales cumplen propiedades de especial relevancia;
fundamentalmente se tiene el siguiente:

Teorema 1.1.7. *El n -ésimo polinomio ortogonal p_n con respecto a la función peso σ en $[-1, 1]$ tiene sus ceros reales, distintos y situados en el intervalo abierto $(-1, 1)$.*

Demostración. Sea p_n el n -ésimo polinomio ortogonal con respecto a σ . Por el Teorema
Fundamental del Álgebra sabemos que este polinomio tiene exactamente n raíces comple-
jas. Construyamos un polinomio r_m formado por las m raíces de multiplicidad impar de
 p_n que estén en el intervalo $(-1, 1)$ $r_m(x) = (x - \psi_1)(x - \psi_2) \cdots (x - \psi_m)$. Por lo tanto, es
suficiente probar que el número de raíces m no es menor que n . Supongamos que $m < n$.
El polinomio r_m cambia de signo en $(-1, 1)$ tantas veces como p_n . Así se tiene que el po-
linomio producto $p_n r_m$ no cambia de signo en $(-1, 1)$ y nótese que $\deg(p_n \cdot r_m) = n + m$.
Como $\sigma \geq 0$ se sigue que $\int_{-1}^1 p_n(x)r_m(x)\sigma(x)dx \neq 0$, pero como $m < n$, esto contradice la
hipótesis de ortogonalidad. Se sigue por lo tanto que $m = n$. □

En el Teorema 1.1.7 se ha expuesto que los polinomios p_n tienen sus ceros simples, reales y
en el intervalo $(-1, 1)$. Otra propiedad sobre entrelazamiento de ceros será estudiada más
adelante.

1.1.3. Ley de recurrencia

Otra notable propiedad que cumplen las familias de polinomios ortogonales es que éstas cumplen una ley de recurrencia a tres términos que, entre otras cosas, permite su computación recursiva a partir de la familia de momentos.

Teorema 1.1.8. *Sea $\{P_k\}_{k=0}^{\infty}$ el sistema de polinomios ortonormales para σ con $P_n(x) = a_n x^n + b_n x^{n-1} + \dots$ ($a_n \neq 0$). Entonces, tres polinomios consecutivos de este sistema satisfacen la relación de recursividad siguiente:*

$$xP_n(x) = \frac{a_n}{a_{n+1}}P_{n+1}(x) + \left(\frac{b_n}{a_n} - \frac{b_{n+1}}{a_{n+1}}\right)P_n(x) + \frac{a_{n-1}}{a_n}P_{n-1}(x), \quad n \geq 0 \quad (1.6)$$

con $a_{-1} = 0, P_{-1} \equiv 0$ y $P_0 \equiv 1$.

Demostración. Se considera la familia ortonormal $\{P_k\}_{k=0}^{n+1}$ con respecto al producto interior $\langle \cdot, \cdot \rangle_{\sigma}$. Como $P_n \in \mathbb{P}_n \setminus \mathbb{P}_{n-1}$ y la familia $\{P_k\}_{k=0}^{n+1}$ es base de \mathbb{P}_{n+1} se tiene que $xP_n(x) = \sum_{k=0}^{n+1} \beta_{n,k} P_k(x)$, donde $\beta_{n,k}$ son los coeficientes de Fourier. A continuación se detalla el cálculo de los coeficientes de Fourier, teniendo presente por la hipótesis de ortonormalidad que $\langle P_i, P_k \rangle_{\sigma} = \delta_{i,k}$ (delta de Kronecker). Sea $k \in \{0, 1, \dots, n+1\}$ fijo. Entonces:

$$\begin{aligned} \langle xP_n, P_k \rangle_{\sigma} &= \beta_0 \langle P_0, P_k \rangle_{\sigma} + \dots + \beta_{k-1} \langle P_{k-1}, P_k \rangle_{\sigma} + \beta_k \langle P_k, P_k \rangle_{\sigma} \\ &\quad + \beta_{k+1} \langle P_{k+1}, P_k \rangle_{\sigma} + \dots + \beta_{n+1} \langle P_{n+1}, P_k \rangle_{\sigma} \\ &= \beta_k. \end{aligned} \quad (1.7)$$

Como $\beta_{n,k} = \beta_k = \langle xP_n, P_k \rangle_{\sigma} = \int_a^b xP_n(x)P_k(x)\sigma(x)dx = \beta_{k,n}$, se sigue que si $k+1 < n$, entonces por las condiciones de ortogonalidad tendremos que $\beta_k = 0$. Más concretamente, $\beta_k = 0, \forall k < n-1$. Por lo tanto tenemos que

$$xP_n(x) = \beta_{n-1}P_{n-1}(x) + \beta_n P_n(x) + \beta_{n+1}P_{n+1}(x)$$

con coeficientes:

$$\begin{aligned} \beta_{n-1} &= \int_a^b xP_n(x)P_{n-1}(x)\sigma(x)dx, \\ \beta_n &= \int_a^b xP_n^2(x)\sigma(x)dx, \quad \beta_{n+1} = \int_a^b xP_n(x)P_{n+1}(x)\sigma(x)dx. \end{aligned}$$

Comparando algunos monomios en (1.1.3) se sigue lo siguiente:

$$\begin{aligned} \text{comparación coeficiente } x^{n+1} &: a_n = \beta_{n+1}a_{n+1} &\Rightarrow \beta_{n+1} &= \frac{a_n}{a_{n+1}}, \\ \text{comparación coeficiente } x^n &: b_n = \beta_n a_n + \beta_{n+1}b_{n+1} &\Rightarrow \beta_n &= \frac{b_n}{a_n} - \frac{b_{n+1}}{a_{n+1}}. \end{aligned}$$

Finalmente, de la expresión del coeficiente de Fourier de β_{n+1} y haciendo un cambio de índice se deduce que $\beta_{n-1} = \frac{a_{n-1}}{a_n}$. Sustituyendo las expresiones obtenidas en (1.1.3) se deduce la relación (1.6). \square

Corolario 1.1.9. *La fórmula de recurrencia para los polinomios ortogonales mónicos admite la forma:*

$$Q_{n+1}(x) = (x - \beta_n)Q_n(x) - \left(\frac{a_{n-1}}{a_n}\right)^2 Q_{n-1}(x), \quad (1.8)$$

donde $\beta_n = \frac{b_n}{a_n} - \frac{b_{n+1}}{a_{n+1}}$, bajo las condiciones iniciales $Q_{-1} \equiv 0$ y $Q_0 \equiv 1$ y la notación del Teorema 1.1.8.

Demostración. Si $\{P_k\}_{k=0}^{\infty}$ es la familia ortonormal y $\{Q_k\}_{k=0}^{\infty}$ la correspondiente familia mónica, entonces $P_n = a_n Q_n$, para todo $n \geq 0$. Sustituyendo en (1.6) se sigue (1.8). \square

Finalizamos esta sección comentando que existe un resultado recíproco a la relación de recurrencia a tres términos conocido como Teorema de Favard. A continuación, lo enunciamos y su demostración puede verse en [1, Teorema 4.4].

Teorema 1.1.10. *Sean $p_{-1} \equiv 0$, $p_0 \equiv 1$ y $\{a_n\}, \{b_n\}$ dos sucesiones arbitrarias y acotadas de números reales con $a_n > 0$. Sea $\{P_k\}_{k=0}^{\infty}$ la familia de polinomios generada por estas dos sucesiones según la ley de recurrencia (1.6). Entonces existe una medida σ tal que esa familia de polinomios es la familia de polinomios ortonormales asociada a σ .*

1.1.4. Fórmula de Christoffel-Darboux. Fórmula confluyente

Comenzamos esta sección con la siguiente definición general:

Definición 1.1.11. *Sea \mathbb{H} un espacio de Hilbert de funciones definidas en \mathbb{X} con producto interior $\langle \cdot, \cdot \rangle$. Entonces, decimos que $\mathbb{K}_w(z) = \mathbb{K}(w, z)$ es un núcleo reproductor si:*

- $\mathbb{K}_w(z) \in \mathbb{H}$, para todo $w \in \mathbb{X}$,
- $\langle f, \mathbb{K}_w \rangle = f(w)$, para todo $w \in \mathbb{X}$ y $f \in \mathbb{H}$.

Se puede comprobar que si \mathbb{H} es un espacio de Hilbert separable y $\{P_k\}_{k \in \Upsilon}$ es una base ortonormal, entonces el único núcleo reproductor viene dado por:

$$\mathbb{K}_n(z, w) = \sum_{k \in \Upsilon} P_k(z)P_k(w).$$

Considérese $\mathbb{H} = \mathbb{P}_n$, $\mathbb{X} = \mathbb{R}$, $\langle \cdot, \cdot \rangle = \langle \cdot, \cdot \rangle_{\sigma}$ (el producto interior inducido por σ), con $\{P_k\}_{k=0}^{\infty}$ base ortonormal y veamos que en efecto se cumple la propiedad reproductora. Si $f_n \in \mathbb{P}_n$, entonces $f_n(x) = \sum_{k=0}^n \beta_k P_k$, donde los coeficientes de Fourier vienen dados por $\beta_k = \int_{-1}^1 f_n(x)P_k(x)\sigma(x)dx$. Tenemos por tanto que :

$$\begin{aligned} \langle f_n(x), \mathbb{K}_n(x, w) \rangle_{\sigma} &= \int_{-1}^1 f_n(x) \sum_{k=0}^n P_k(w)P_k(x)\sigma(x)dx \\ &= \sum_{k=0}^n P_k(w) \int_{-1}^1 f_n(x)P_k(x)\sigma(x)dx \\ &= \sum_{k=0}^n \beta_k P_k(w) \\ &= f_n(w). \end{aligned}$$

El siguiente resultado es una expresión para el n -ésimo núcleo reproductor en términos de los polinomios ortonormales n y $(n + 1)$ -ésimos.

Teorema 1.1.12. Fórmula de Christoffel-Darboux. *El n -ésimo núcleo reproductor admite la expresión:*

$$(x - t) \cdot \mathbb{K}_n(x, t) = \frac{a_n}{a_{n+1}} \left(P_{n+1}(x)P_n(t) - P_n(x)P_{n+1}(t) \right).$$

Demostración. Recordemos la fórmula de recurrencia (1.6) para la familia ortonormal:

$$xP_n(x) = \frac{a_n}{a_{n+1}}P_{n+1}(x) + \left(\frac{b_n}{a_n} - \frac{b_{n+1}}{a_{n+1}} \right) P_n(x) + \frac{a_{n-1}}{a_n}P_{n-1}(x).$$

Si multiplicamos esta expresión por $P_n(t)$ se obtiene (I) e intercambiando las variables x y t , se obtiene (II):

$$\begin{aligned} xP_n(x)P_n(t) &= \frac{a_n}{a_{n+1}}P_{n+1}(x)P_n(t) + \left(\frac{b_n}{a_n} - \frac{b_{n+1}}{a_{n+1}} \right) P_n(x)P_n(t) + \frac{a_{n-1}}{a_n}P_{n-1}(x)P_n(t), & (I) \\ tP_n(t)P_n(x) &= \frac{a_n}{a_{n+1}}P_{n+1}(t)P_n(x) + \left(\frac{b_n}{a_n} - \frac{b_{n+1}}{a_{n+1}} \right) P_n(t)P_n(x) + \frac{a_{n-1}}{a_n}P_{n-1}(t)P_n(x). & (II) \end{aligned}$$

Restándole a (I) la expresión (II) se deduce:

$$(x-t)P_n(x)P_n(t) = \frac{a_n}{a_{n+1}}[P_{n+1}(x)P_n(t) - P_n(x)P_{n+1}(t)] - \frac{a_{n-1}}{a_n}[P_{n-1}(x)P_n(t) - P_n(x)P_{n-1}(t)].$$

Reiterando el proceso para los valores $n - 1, n - 2, \dots, 0$, y sumando los resultados se concluye (1.1.1). \square

Como aplicación directa de la fórmula de Christoffel-Darboux concluimos esta sección deduciendo la siguiente:

Proposición 1.1.13. Fórmula confluyente. *Si $\{P_k\}_{k=0}^{\infty}$ denota la familia de polinomios ortonormales, entonces se cumple*

$$\sum_{k=0}^n P_k^2(x) = \frac{a_n}{a_{n+1}} \left(P'_{n+1}(x)P_n(x) - P'_n(x)P_{n+1}(x) \right).$$

Demostración. Directamente, tomando límite cuando t tiende a x en (1.1.12) y aplicando la regla de L'Hôpital:

$$\begin{aligned} \sum_{k=0}^n P_k^2(x) &= \lim_{t \rightarrow x} \frac{\left((a_n/a_{n+1})[P_{n+1}(x)P_n(t) - P_n(x)P_{n+1}(t)] \right)}{x - t} \\ &= \frac{a_n}{a_{n+1}} \left(P'_{n+1}(x)P_n(x) - P'_n(x)P_{n+1}(x) \right). \end{aligned}$$

\square

1.1.5. Matrices de Jacobi

En esta sección veremos cómo el problema de computar ceros de polinomios ortogonales es equivalente a un problema de cálculo de autovalores para ciertas matrices asociadas a la ley de recurrencia a tres términos, conocidas como *matrices de Jacobi*. Desde el punto de vista de la estabilidad numérica, es preferible computar ceros de polinomios mediante esta vía alternativa en lugar de hacerlo directamente como raíces de funciones polinómicas. Además, como las matrices de Jacobi son tridiagonales, es bien conocido que existen numerosos métodos numéricos del Álgebra Lineal Numérica para el cálculo rápido y eficiente de sus autovalores.

Comenzamos con la ley de recurrencia de polinomios ortonormales a tres términos en la forma:

$$\begin{cases} xP_n(x) &= \zeta_{n+1}P_{n+1}(x) + \eta_n P_n(x) + \zeta_n P_{n-1}(x), & n \geq 0, \\ P_{-1}(x) &\equiv 0, & P_0(x) \equiv 1. \end{cases} \quad (1.9)$$

Nótese que en las condiciones iniciales estamos considerando, sin pérdida de generalidad, una función peso de probabilidad, esto es, el primer momento es unitario. En el caso de que la función peso no lo sea, bastaría con hacer un reesclamiento. Asimismo, aclarar que no debe confundirse la relación de recurrencia (1.6) con la anterior (1.9), si bien ambas son equivalentes.

Desarrollando el sistema para todo $n \geq 0$ se obtiene:

$$\begin{cases} xP_0(x) &= \zeta_1 P_1(x) + \eta_0 P_0(x) + 0, \\ xP_1(x) &= \zeta_2 P_2(x) + \eta_1 P_1(x) + \zeta_1 P_0(x), \\ xP_2(x) &= \zeta_3 P_3(x) + \eta_2 P_2(x) + \zeta_2 P_1(x), \\ &\vdots \\ xP_{n-1}(x) &= \zeta_n P_n(x) + \eta_{n-1} P_{n-1}(x) + \zeta_{n-1} P_{n-2}(x), \end{cases} \quad (1.10)$$

que en forma matricial queda expresado de la forma:

$$x \cdot \begin{pmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ \vdots \\ P_{n-2}(x) \\ P_{n-1}(x) \end{pmatrix} = \begin{pmatrix} \eta_0 & \zeta_1 & 0 & 0 & \cdots & 0 & 0 \\ \zeta_1 & \eta_1 & \zeta_2 & 0 & \cdots & 0 & 0 \\ 0 & \zeta_2 & \eta_2 & \zeta_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \eta_{n-2} & \zeta_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & \zeta_{n-1} & \eta_{n-1} \end{pmatrix} \cdot \begin{pmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ \vdots \\ P_{n-2}(x) \\ P_{n-1}(x) \end{pmatrix} + \zeta_n P_n(x) \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (1.11)$$

Denotando por $x = x_{j,n}$ al cero j -ésimo de P_n con $j = 1, \dots, n$, esto es, $P_n(x_{j,n}) = 0$, y evaluándolo en la expresión anterior se llega a

$$J_n \cdot \begin{pmatrix} P_0(x_{j,n}) \\ P_1(x_{j,n}) \\ P_2(x_{j,n}) \\ \vdots \\ P_{n-2}(x_{j,n}) \\ P_{n-1}(x_{j,n}) \end{pmatrix} = x_{j,n} \cdot \begin{pmatrix} P_0(x_{j,n}) \\ P_1(x_{j,n}) \\ P_2(x_{j,n}) \\ \vdots \\ P_{n-2}(x_{j,n}) \\ P_{n-1}(x_{j,n}) \end{pmatrix},$$

siendo J_n la matriz tridiagonal y simétrica de Jacobi definida por

$$J_n := \begin{pmatrix} \eta_0 & \zeta_1 & 0 & 0 & \cdots & 0 & 0 \\ \zeta_1 & \eta_1 & \zeta_2 & 0 & \cdots & 0 & 0 \\ 0 & \zeta_2 & \eta_2 & \zeta_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \eta_{n-2} & \zeta_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & \zeta_{n-1} & \eta_{n-1} \end{pmatrix}. \quad (1.12)$$

Vemos por tanto que $x = x_{j,n}$, para todo $j = 1, \dots, n$, es un autovalor de J_n con autovector asociado:

$$v_{j,n} = (P_0(x_{j,n}) \ P_1(x_{j,n}) \ P_2(x_{j,n}) \ \cdots \ P_{n-2}(x_{j,n}) \ P_{n-1}(x_{j,n}))^T. \quad (1.13)$$

Nótese que como J_n es real y simétrica se sigue que los autovalores son reales. Además, es directo comprobar de la fórmula confluyente que estos autovalores son simples: si x_0 es un cero de multiplicidad mayor que 1, entonces de la fórmula confluyente (1.11) tenemos $\sum_{k=0}^{n-1} (P_k(x_0))^2 = 0$, lo cual es imposible ya que $P_0(x_0) \equiv 1$.

El procedimiento aportado anteriormente puede resumirse en el siguiente

Teorema 1.1.14. *Los n autovalores de la matriz de Jacobi J_n definida en (1.12) asociada a σ son los n ceros de $P_n(x)$.*

1.1.6. Entrelazamiento de ceros

En este apartado se aportará una nueva propiedad sobre los ceros de polinomios ortogonales.

Teorema 1.1.15. *Si denotamos por $x_{i,n}$ ($i = 1, \dots, n$) los n ceros del polinomio ortogonal p_n , entonces se cumple $x_{j,n} < x_{j,n-1} < x_{j+1,n}$. En otras palabras, los ceros de dos polinomios ortogonales de grados consecutivos se entrelazan.*

Demostración. Si $p_n(x) = a_n x^n + \dots$, $a_n > 0$, utilizando la fórmula confluyente evaluada en $x_{j,n}$ se tiene que

$$\begin{aligned} \sum_{k=0}^n p_k^2(x_{j,n}) &= a_n [p_n'(x_{j,n})p_{n-1}(x_{j,n}) - p_{n-1}'(x_{j,n})p_n(x_{j,n})] \\ &= a_n [p_n'(x_{j,n})p_{n-1}(x_{j,n})] > 0. \end{aligned}$$

Esto implica que $\text{sign}(p_{n-1}(x_{j,n})) = (-1)^{n-j}$, siendo $\text{sign}(x)$ la función signo. Como el polinomio P_{n-1} cambia de signo al pasar de $x_{j,n}$ a $x_{j+1,n}$, se verifica que posee al menos un cero entre dos ceros de P_n , para todo $j = 1, \dots, n-1$ y como es de grado $n-1$, por el Teorema Fundamental del Álgebra tiene a lo sumo $(n-1)$ ceros reales, concluyendo que P_{n-1} tiene exactamente un cero entre dos ceros consecutivos de P_n .

Además, no es posible que P_n y P_{n-1} tengan un cero común: si x_0 fuera un cero común, como consecuencia de la fórmula confluyente se tendría

$$\sum_{k=0}^{k=n} (P_k(x_0))^2 = a_n [P_n'(x_0)P_{n-1}(x_0) - P_{n-1}'(x_0)P_n(x_0)] = 0,$$

lo cual es imposible dado que $P_0(x_0) \neq 0$. □

Nótese que en la demostración anterior se han utilizado implícitamente los Teoremas de Bolzano y Rolle.

1.1.7. Familias particulares de polinomios ortogonales

Dedicamos esta sección a mencionar brevemente algunas familias de polinomios ortogonales correspondientes a funciones peso particulares, ampliamente conocidas en la literatura, y para las cuales se han estudiado numerosas propiedades.

Polinomios de Jacobi

- Función peso: $\sigma(x) = (1-x)^\alpha(1+x)^\beta$, $\alpha, \beta > -1$, $x \in [-1, 1]$.
- Fórmula de Rodrigues:

$$\begin{aligned} p_n^{(\alpha, \beta)}(x) &= \frac{(-1)^n}{2^n n!} (1-x)^{-\alpha} (1+x)^{-\beta} \frac{d^n}{dx^n} [(1-x)^{\alpha+n} (1+x)^{\beta+n}] \\ &= \frac{\Gamma(\alpha+\beta+2n+1)}{2^n n! \Gamma(\alpha+\beta+n+1)} x^n + \dots, \end{aligned}$$

siendo $\Gamma(x)$ la función Gamma.

- Ley de recurrencia:

$$\begin{aligned} &2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)p_{n+1}^{(\alpha, \beta)}(x) \\ &= (2n+\alpha+\beta+1)[(\alpha^2-\beta^2) + (2n+\alpha+\beta+2)(2n+\alpha+\beta)x]p_n^{(\alpha, \beta)}(x) \\ &\quad - 2(n+\alpha)(n+\beta)(2n+\alpha+\beta+2)p_{n-1}^{(\alpha, \beta)}(x). \end{aligned}$$

- Condiciones de ortogonalidad:

$$\begin{aligned} I_{n,m} &= \int_{-1}^1 (1-x)^\alpha (1+x)^\beta p_n^{(\alpha, \beta)}(x) p_m^{(\alpha, \beta)}(x) dx \\ &= \begin{cases} 0 & \text{si } m \neq n, \\ \frac{2^{\alpha+\beta+1} \Gamma(\alpha+n+1) \Gamma(\beta+n+1)}{(\alpha+\beta+2n+1)(n!) \Gamma(\alpha+\beta+n+1)} & \text{si } m = n. \end{cases} \end{aligned}$$

- Coeficiente director de los polinomios ortogonales: $a_n = \frac{\Gamma(\alpha+\beta+2n+1)}{2^n n! \Gamma(\alpha+\beta+n+1)}$.
- Condiciones de ortonormalidad: $P_n^{(\alpha, \beta)}(x) = \frac{1}{\sqrt{I_{n,n}}} p_n^{(\alpha, \beta)}(x)$.
- Coeficiente director de los polinomios ortonormales: $A_n = \frac{1}{\sqrt{I_{n,n}}} a_n$.

Polinomios de Legendre

- Función peso $\sigma(x) \equiv 1$, $x \in [-1, 1]$. Nótese que es un caso particular de la función peso de Jacobi con $\alpha = \beta = 0$.
- Fórmula de Rodrigues:

$$\begin{aligned} p_n^{(0,0)}(x) &= \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n = \frac{(2n)!}{2^n (n!)^2} x^n - \frac{(2n-2)!}{2^n (n-1)!(n-2)!} x^{n-2} + \dots \\ &= \frac{1}{2^n} \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^m \binom{n}{m} \binom{2n-2m}{n} x^{n-2m}. \end{aligned}$$

siendo $\lfloor \cdot \rfloor$ la función parte entera.

- Ley de recurrencia:

$$(n+1)p_{n+1}(x) = (2n+1)xp_n(x) - np_{n-1}(x), \quad p_0(x) \equiv 1, \quad p_1(x) = x. \quad (1.14)$$

- Condiciones de ortogonalidad:

$$I_{n,m} = \int_{-1}^1 p_n^{(0,0)}(x)p_m^{(0,0)}(x)dx = \begin{cases} 0 & \text{si } m \neq n, \\ \frac{2}{2n+1} & \text{si } m = n. \end{cases}$$

- Coeficiente director de los polinomios ortogonales: $a_n = \frac{(2n)!}{2^n (n!)^2}$.
- Condiciones de ortonormalidad: $P_n^{(0,0)}(x) = \sqrt{\frac{2n+1}{2}} p_n^{(0,0)}(x)$.
- Coeficiente director de los polinomios ortonormales: $A_n = \sqrt{\frac{2n+1}{2}} a_n$.

Polinomios de Chebyshev de primera especie

- Función peso: $\sigma(x) = \frac{1}{\sqrt{1-x^2}}$, $x \in [-1, 1]$. Nótese que es un caso particular de la función peso de Jacobi con $\alpha = \beta = -\frac{1}{2}$.
- Polinomio ortogonal: $t_n(x) = \cos(n \arccos(x))$, $\forall n = 0, 1, \dots$
- Ley de recurrencia:

$$t_{n+1}(x) = 2xt_n(x) - t_{n-1}(x), \quad t_0(x) = 1, \quad t_1(x) = x. \quad (1.15)$$

- Sustituyendo: $x = \cos\theta$, se obtiene la fórmula trigonométrica:

$$\cos((n+1)\theta) + \cos((n-1)\theta) = 2 \cos\theta \cos(n\theta).$$

- Condiciones de ortogonalidad:

$$I_{n,m} = \int_{-1}^1 \frac{t_n(x)t_m(x)}{\sqrt{1-x^2}} dx = \int_0^\pi \cos(n\theta) \cos(m\theta) d\theta = \begin{cases} 0 & \text{si } m \neq n, \\ \pi & \text{si } m = n = 0, \\ \frac{\pi}{2} & \text{si } m = n \neq 0. \end{cases}$$

- Coeficiente director de los polinomios ortogonales: $A_n = 2^{n-1}$.
- Ceros: $x_k = \cos\left(\frac{(2k-1)\pi}{2n}\right)$, $\forall k = 1, 2, \dots, n$.

Nota 1.1.16. Si $p_n^{(-\frac{1}{2}, -\frac{1}{2})}(x)$ es un polinomio de Jacobi, difiere de los polinomios de Chebyshev de primera especie en una constante, esto es:

$$p_n^{(-\frac{1}{2}, -\frac{1}{2})}(x) = c_n t_n(x), \text{ siendo } c_n = \frac{\Gamma(2n)}{2^{2n-1}\Gamma(n)\Gamma(n+1)}.$$

Polinomios de Chebyshev de segunda especie

- Función peso: $\sigma(x) = \sqrt{1-x^2}$, $x \in [-1, 1]$. Nótese que es un caso particular de la función peso de Jacobi con $\alpha = \beta = \frac{1}{2}$.
- Polinomio ortogonal: $u_n(x) = \frac{\sin((n+1)\arccos(x))}{\sqrt{1-x^2}}$, $\forall n = 0, 1, \dots$

- Ley de recurrencia:

$$u_{n+1}(x) = 2xu_n(x) - u_{n-1}(x), \quad u_0(x) \equiv 0, \quad u_1(x) = x. \quad (1.16)$$

- Sustituyendo: $x = \cos\theta$, se obtiene la fórmula trigonométrica:

$$\sin((n+2)\theta) + \sin(n\theta) = 2\cos\theta \sin((n+1)\theta).$$

- Condiciones de ortogonalidad:

$$\begin{aligned} I_{n,m} &= \int_{-1}^1 u_n(x)u_m(x)\sqrt{(1-x^2)}dx = \int_0^\pi \sin((n+1)\theta)\sin((m+1)\theta)d\theta \\ &= \begin{cases} 0 & \text{si } m \neq n, \\ \frac{\pi}{2} & \text{si } m = n. \end{cases} \end{aligned}$$

- Coeficiente director de los polinomios ortogonales: $a_n = 2^n$.
- Ceros: $x_k = \cos\left(\frac{k\pi}{n+1}\right)$, $\forall k = 1, 2, \dots, n$.

Nota 1.1.17. Si $P_n^{(\frac{1}{2}, \frac{1}{2})}(x)$ es un polinomio de Jacobi, difiere de los polinomios de Chebyshev de segunda especie en una constante, esto es:

$$p_n^{(\frac{1}{2}, \frac{1}{2})}(x) = c_n u_n(x), \text{ siendo } c_n = \frac{(2n+1)!}{2^{2n}n!(n+1)!}.$$

Nota 1.1.18. Esta familia de polinomios cumplen una conocida propiedad extremal:

$$\int_{-1}^1 |p(x)|dx \geq 2^{1-n}.$$

En particular: $\min_{Q_n \in \mathbb{P}_n} \int_{-1}^1 |Q_n(x)|dx = \int_{-1}^1 |p(x)^{(\frac{1}{2}, \frac{1}{2})}|dx = 2^{1-n}$.

Polinomios de Hermite

- Función peso: $\sigma(x) = e^{-x^2}$, $x \in (-\infty, \infty)$.
- Polinomio ortogonal (Rodrigues): $h_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}$.
- Ley de recurrencia:

$$h_{n+1}(x) = 2xh_n(x) - h'_n(x), \quad \text{con } h_0(x) \equiv 1, \quad h_1(x) = 2x,$$

y como $h'_n(x) = -(2n)h_{n-1}$ se sigue que $h_{n+1}(x) = 2xh_n(x) + (2n)h_{n-1}(x)$.

- Condiciones de ortogonalidad:

$$I_{n,m} = \int_{-\infty}^{\infty} h_n(x)h_m(x)e^{-x^2} dx = \begin{cases} 0 & \text{si } m \neq n, \\ 2^n \sqrt{\pi} n! & \text{si } m = n. \end{cases}$$

- Coeficiente director de los polinomios ortogonales: $a_n = 2^{\frac{n}{2}} (n!)^{-\frac{1}{2}} (\pi)^{\frac{1}{4}}$.
- Condiciones de ortonormalidad: $H_n(x) = \frac{h_n(x)(n!)^{\frac{1}{2}}}{2^{\frac{n}{2}} (\pi)^{\frac{1}{4}}}$.

Polinomios de Laguerre

- Función peso: $\sigma(x) = x^\alpha e^{-x}$, $\alpha \geq -1$, $x \in [0, \infty)$.
- Polinomio ortogonal (Rodrigues): $l_n^\alpha(x) = (-1)^n x^{-\alpha} e^x \frac{d^n}{dx^n} (x^{\alpha+n} e^{-x})$.
Desarrollando para todo $n \in 0, 1, 2, \dots$ se obtiene:

$$\begin{aligned} l_n^\alpha(x) &= x^n - \frac{n}{1!}(n+\alpha)x^{n-1} + \frac{n(n+1)}{2!}(n+\alpha)(n+\alpha-1)x^{n-2} - \dots \\ &= \sum_{m=0}^n (-1)^m \binom{n+\alpha}{n-m} \frac{1}{m!} x^m. \end{aligned}$$

- Ley de recurrencia:

$$(n+1)l_{n+1}^\alpha(x) = [(x+2n+1)-x]l_n^\alpha(x) - (n+\alpha)l_{n-1}^\alpha(x), \quad \text{con } l_0^\alpha(x) \equiv 1, \quad l_1^\alpha(x) = 1+\alpha-x.$$

- Condiciones de ortogonalidad:

$$I_{n,m} = \int_{-\infty}^{\infty} l_n^\alpha(x)l_m^\alpha(x)x^\alpha e^{-x} dx = \begin{cases} 0 & \text{si } m \neq n, \\ n!\Gamma(\alpha+n+1) & \text{si } m = n. \end{cases}$$

- Condiciones de ortonormalidad: $L_n^\alpha(x) = \frac{l_n^\alpha(x)}{\sqrt{(n!\Gamma(\alpha+n+1))}}$.

1.2. Interpolación polinómica

Como ya se ha mencionado anteriormente, un resultado fundamental en esta Memoria es el Teorema de Aproximación de Weierstrass. Las funciones polinómicas son las funciones matemáticas más fáciles de evaluar, pues solamente se deben emplear operaciones aritméticas elementales. En esta sección planteamos otro concepto de vital importancia para la construcción de fórmulas de cuadratura: la interpolación. Más concretamente, planteamos el siguiente problema de Interpolación Polinómica:

Teorema 1.2.1. Sean $\{x_i\}_{i=0}^n$ $(n+1)$ puntos dados e $\{y_i\}_{i=0}^n$ $(n+1)$ cantidades reales fijas. Entonces existe un único polinomio $P_n \in \mathbb{P}_n$ tal que $P_n(x_i) = y_i$, $\forall i = 0, \dots, n$.

Demostración. Veamos la existencia. Sea $P_n(x) = \sum_{\nu=0}^n a_\nu x^\nu$. Imponiendo las condiciones de interpolación se obtiene un sistema lineal cuya matriz asociada tiene un determinante de tipo Vandermonde:

$$W(x_0, x_1, \dots, x_n) = \begin{vmatrix} x_0^0 & x_0^1 & \dots & x_0^n \\ x_1^0 & x_1^1 & \dots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ x_n^0 & x_n^1 & \dots & x_n^n \end{vmatrix} = \begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^n \end{vmatrix} = \prod_{i < j} (x_j - x_i) \neq 0,$$

dado que los nodos son diferentes ($x_i \neq x_j$, $\forall i \neq j$). La existencia queda por tanto garantizada. Para probar que es único, supongamos que hay dos polinomios mónicos de interpolación $P, Q \in \mathbb{P}_n$ que satisfacen las $(n+1)$ condiciones. De esta forma $P - Q \in \mathbb{P}_{n-1}$ y evaluándolo en los nodos x_i se tiene:

$$(P - Q)(x_i) = P(x_i) - Q(x_i) = y_i - y_i = 0, \quad \forall i = 0, \dots, n.$$

Se sigue que $P - Q \in \mathbb{P}_{n-1}$ tiene n ceros, luego $P - Q \equiv 0$, lo cual implica que $P \equiv Q$.

□

A efectos prácticos, daremos la fórmula de Lagrange como una técnica elemental explícita para el cálculo del polinomio P_n . Sea $w(x)$ el polinomio nodal:

$$w(x) := \prod_{i=0}^n (x - x_i) \in \mathbb{P}_n, \text{ mónico}, \quad (1.17)$$

siendo x_i todos los puntos dados. Así se pueden definir para $k = 0, \dots, n$ los polinomios fundamentales de Lagrange $w_k(x)$ correspondientes al nodo x_k como:

$$w_k(x) = \frac{(x - x_0) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)}{(x_k - x_0) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)} = \frac{w(x)}{(x - x_k)w'(x_k)}. \quad (1.18)$$

Obsérvese que $w_k(x_i) = \delta_{i,k}$. De los polinomios fundamentales obtenemos por tanto la siguiente expresión para el polinomio interpolador:

$$P_n(x) = \sum_{k=0}^n \frac{w(x)}{(x - x_k)w'(x_k)} f(x_k).$$

A continuación analizaremos el error en la interpolación polinómica de una función f . Nótese que aunque el polinomio interpolador cumple las condiciones de interpolación, a medida que el grado del polinomio aumenta, éste presentará mayores oscilaciones, y por tanto mayor será el error cometido. En el caso de que la función f presente ciertas condiciones de regularidad en $[-1,1]$ sucederá que el polinomio de interpolación será una buena aproximación. El teorema que sigue proporciona una expresión útil que permite estudiar el error de la interpolación por polinomios cuando la función dada posee derivadas de hasta orden $(n+1)$. Mediante este resultado se puede obtener una estimación del error que estamos cometiendo al aproximar f por P_n . Como nuestro objetivo será aproximar la integral con respecto a σ de f por la integral de P_n , ello favorecerá para obtener una cota del error cometido en la integración numérica.

Teorema 1.2.2. Teorema del resto de Lagrange. Sean $\{x_i\}_{i=0}^n$ $(n+1)$ puntos distintos en el dominio $[-1, 1]$ de la función f y P_n el polinomio de interpolación de grado menor o igual que n que coincide con f en esos puntos. Si $f \in C^{n+1}([-1, 1])$, existe por lo menos un punto interior ξ en $(-1, 1)$ tal que:

$$R_n(x) := f(x) - P_n(x) = \frac{w(x)}{(n+1)!} f^{(n+1)}(\xi). \quad (1.19)$$

Demostración. En el caso de que $x = x_i$ para $i \in \{0, 1, \dots, n\}$, el error será cero porque $w(x_k) = 0$ y entonces la ecuación (1.19) se verifica para todo $\xi \in (-1, 1)$.

Si $x \neq x_i$, manteniendo fijo x se define en $[-1, 1]$ la función

$$G_x(t) := w_n(x)(f(t) - P_n(t)) - w_n(t)(f(x) - P_n(x)). \quad (1.20)$$

La función $G_x(t)$ tiene derivada de orden $(n+1)$. Además, como $P_n(t)$ es un polinomio de grado menor o igual que n , su derivada $(n+1)$ -ésima es idénticamente nula. El polinomio $w_n(t)$ es de grado $n+1$, siendo el término de mayor grado t^{n+1} y tenemos $w_n^{(n+1)}(t) = (n+1)!$. Por lo tanto, si derivamos $(n+1)$ -veces con respecto a t en la ecuación (1.20) se obtiene la fórmula:

$$G_x^{(n+1)}(t) = w_n(x)f^{(n+1)}(t) - (n+1)!(f(x) - P_n(x)). \quad (1.21)$$

De la definición (1.20) se tiene que $G_x(t) = 0$ en $(n+2)$ puntos distintos del intervalo $[-1, 1]$ ($n+1$ puntos de interpolación y el punto x). Por consiguiente, estos puntos determinan $(n+1)$ -subintervalos consecutivos de $[-1, 1]$ y la función G_x se anula en los dos extremos de esos subintervalos. Por el Teorema de Rolle, la derivada $G_x'(t)$ debe anularse por lo menos

en un punto t interior a cada subintervalo. Si elegimos exactamente uno de tales puntos t en cada subintervalo obtenemos $(n+1)$ -puntos distintos en el intervalo $(-1, 1)$ en los que $G'_x(t) = 0$. Esos puntos, a su vez, determinan n subintervalos en cuyos extremos tenemos $G'_x(t) = 0$. Aplicando el Teorema de Rolle a G'_x encontramos que la derivada segunda $G''_x(t)$ es cero por lo menos en n -puntos distintos de $(-1, 1)$. Aplicando reiteradamente el teorema de Rolle $(n+1)$ -veces se deduce que por lo menos existe un punto ξ en $(-1, 1)$ en el que $G_x^{(n+1)}(\xi) = 0$ y sustituyéndolo en la ecuación (1.21) se obtiene

$$(n+1)!(f(x) - P_n(x)) = w_n(x)f^{(n+1)}(\xi),$$

que coincide con (1.19). □

Con carácter ilustrativo, finalizamos esta sección con el siguiente:

Ejemplo 1.2.3. Sean $X=[0, 3, 7, 9, 10]$ cinco puntos dados e $Y=[1, 3, -2, -1, 4]$ cinco cantidades reales fijas. Calculemos el polinomio interpolador de Lagrange $P_4(x)$ asociado a X e Y . El polinomio nodal es :

$$w(x) = (x-0)(x-3)(x-7)(x-9)(x-10).$$

Calculemos los polinomios fundamentales de Lagrange:

$$\begin{aligned} w_1(x) &= \frac{(x-3)(x-7)(x-9)(x-10)}{(0-3)(0-7)(0-9)(0-10)} = \frac{1}{1890}(x-3)(x-7)(x-9)(x-10), \\ w_2(x) &= \frac{(x-0)(x-7)(x-9)(x-10)}{(3-0)(3-7)(3-9)(3-10)} = \frac{-1}{504}x(x-7)(x-9)(x-10), \\ w_3(x) &= \frac{(x-0)(x-3)(x-9)(x-10)}{(7-0)(7-3)(7-9)(7-10)} = \frac{1}{168}x(x-3)(x-9)(x-10), \\ w_4(x) &= \frac{(x-0)(x-3)(x-7)(x-10)}{(9-0)(9-3)(9-7)(9-10)} = \frac{-1}{108}x(x-3)(x-7)(x-10), \\ w_5(x) &= \frac{(x-0)(x-3)(x-7)(x-9)}{(10-0)(10-3)(10-7)(10-9)} = \frac{1}{210}x(x-3)(x-7)(x-9). \end{aligned}$$

Tendremos por tanto que el polinomio interpolador viene dado por:

$$\begin{aligned} P_4(x) &= 1w_1(x) + 3w_2(x) - 2w_3(x) - 1w_4(x) + 4w_5(x) \\ &= \frac{83}{7560}x^4 - \frac{551}{3780}x^3 + \frac{2393}{7560}x^2 + \frac{923}{1260}x + 1 \\ &= 0,0110x^4 - 0,1458x^3 + 0,3165x^2 + 0,7325x + 1,0000. \end{aligned}$$

En el código A.1 del Apéndice A se puede encontrar el código implementado en lenguaje Matlab que permite computar el polinomio interpolador de Lagrange de un conjunto de puntos y valores dados, que se ha implementado en el Ejemplo 1.2.3.

1.3. Ejemplos numéricos

Finalizamos este Capítulo presentando algunos ejemplos numéricos sobre cálculo de ceros de las familias de polinomios ortogonales de la Sección 1.1.7 vía cálculo de autovalores de matrices de Jacobi, Sección 1.1.5. Las leyes de recurrencia para estas familias han permitido caracterizar las entradas de las correspondientes matrices.

1.3.1. Polinomios de Chebyshev de primera especie

La expresión de los polinomios ortogonales (ni mónicos ni ortonormales) se recoge en la expresión (1.15) con $\sigma(x) = \frac{1}{\sqrt{1-x^2}}$. Para determinar la expresión de recurrencia de los polinomios ortonormales similar a (1.9), nótese que las relaciones

$$\int_{-1}^1 \frac{dx}{\sqrt{1-x^2}} = \pi \quad \text{y} \quad \int_{-1}^1 \frac{t_n^2 dx}{\sqrt{1-x^2}} = \frac{\pi}{2}, \quad n \geq 1,$$

implican

$$n \geq 2: \|T_n\|_{2,\sigma}^2 = \langle T_n, T_n \rangle_\sigma = \frac{2}{\pi} \langle t_n, t_n \rangle_\sigma = 1 \Rightarrow T_n = \sqrt{\frac{2}{\pi}} t_n,$$

$$n = 0: \|T_0\|_{2,\sigma}^2 = \langle T_0, T_0 \rangle_\sigma = \frac{1}{\pi} \langle t_0, t_0 \rangle_\sigma = 1 \Rightarrow T_0 = \sqrt{\frac{1}{\pi}} t_0.$$

$n = 1$: De (1.10) se deduce:

$$t_2(x) = 2xt_1(x) - t_0(x) \quad \Rightarrow \quad \sqrt{\frac{\pi}{2}} T_2(x) = 2x \sqrt{\frac{\pi}{2}} T_1(x) - \sqrt{\pi} T_0(x) \quad \Rightarrow$$

$$T_2(x) = 2xT_1(x) - \sqrt{2}T_0(x).$$

De este modo se concluye que la familia ortonormal de polinomios admite las expresiones $xT_0(x) = \frac{1}{\sqrt{2}}T_1(x)$, $xT_1(x) = \frac{1}{2}T_2(x) - \frac{\sqrt{2}}{2}T_0(x)$, que proporciona $T_{-1}(x) \equiv 0$, $T_0(x) \equiv \sqrt{\frac{1}{\pi}}$, $xT_n(x) = \frac{1}{2}T_{n+1}(x) + \frac{1}{2}T_{n-1}(x)$, $\forall n \geq 2$.

Por tanto se obtienen las sucesiones $\{\zeta_k\}_{k=0}^n = \{\frac{1}{\sqrt{2}}, \frac{1}{2}, \dots\}$ y $\{\eta_k\}_{k=0}^n = 0$, correspondientes a la matriz de Jacobi dada por (1.11).

1.3.2. Polinomios de Chebyshev de segunda especie

La fórmula de recurrencia de los polinomios ortogonales se recoge en la expresión (1.16) con $\sigma(x) = \sqrt{(1-x^2)}$ y las condiciones de ortogonalidad:

$$I_{n,m} = \int_{-1}^1 u_n(x)u_m(x)\sqrt{(1-x^2)}dx = \begin{cases} 0 & \text{si } m \neq n, \\ \frac{\pi}{2} & \text{si } m = n. \end{cases}$$

Entonces, la expresión de recurrencia de los polinomios ortonormales se calcula a partir de:

$$\forall n \geq 1: \|U_n\|_{2,\sigma}^2 = \langle U_n, U_n \rangle_\sigma = \langle \sqrt{\frac{2}{\pi}}u_n, \sqrt{\frac{2}{\pi}}u_n \rangle_\sigma = \frac{2}{\pi} \langle u_n, u_n \rangle_\sigma = 1$$

lo cual implica que $U_n = \sqrt{\frac{2}{\pi}}u_n$. Por consiguiente, sustituyéndolo en (1.16) se tiene:

$$\begin{aligned} u_{n+1}(x) &= 2xu_n(x) - u_{n-1}(x) \Rightarrow \sqrt{\frac{\pi}{2}}U_{n+1}(x) = 2x\sqrt{\frac{\pi}{2}}U_n(x) - \sqrt{\frac{\pi}{2}}U_{n-1}(x) \Rightarrow \\ U_{n+1}(x) &= 2xU_n(x) - U_{n-1}(x) \Rightarrow xU_n(x) = \frac{1}{2}U_{n+1}(x) + \frac{1}{2}U_{n-1}(x), \end{aligned}$$

que es la fórmula de recurrencia ortonormal de los polinomios de Chebyshev de segunda especie. Concluimos que la relación de la familia ortonormal es:

$$xU_n(x) = \frac{1}{2}U_{n+1}(x) + \frac{1}{2}U_{n-1}(x), \quad \forall n \geq 2, \quad \text{con} \quad U_0(x) = \sqrt{\frac{1}{2}}, \quad U_1(x) = \sqrt{\frac{3}{2}}x.$$

Así se hallan las sucesiones $\{\zeta_n\}_{k=0}^n = \frac{1}{2}$ y $\{\eta_n\}_{k=0}^n = 0$.

1.3.3. Polinomios de Legendre

Los polinomios ortogonales de Legendre vienen recogidos en la expresión (1.14) A partir de las condiciones de ortogonalidad:

$$I_{n,m} = \int_{-1}^1 p_n^{(0,0)}(x)p_m^{(0,0)}(x)dx = \begin{cases} 0 & \text{si } m \neq n, \\ \frac{2}{2n+1} & \text{si } m = n, \end{cases}$$

y las condiciones de ortonormalidad

$$P_n^{(0,0)}(x) = \sqrt{\frac{2n+1}{2}}p_n^{(0,0)}(x), \quad (1.22)$$

sustituyendo (1.22) en (1.14) se obtiene:

$$\begin{aligned} (n+1)\frac{\sqrt{2}}{\sqrt{2n+3}}P_{n+1}(x) &= (2n+1)\frac{\sqrt{2}}{\sqrt{2n+1}}xP_n(x) - n\frac{\sqrt{2}}{\sqrt{2n-1}}P_{n-1}(x) \Rightarrow \\ \frac{n+1}{\sqrt{2n+3}}P_{n+1}(x) &= \frac{2n+1}{\sqrt{2n+1}}xP_n(x) - \frac{n}{\sqrt{2n-1}}P_{n-1}(x). \end{aligned}$$

De este modo, se tiene la relación de recurrencia a tres términos de los polinomios ortogonales:

$$xP_n^{(0,0)}(x) = \frac{n+1}{\sqrt{2n+1}\sqrt{2n+3}}P_{n+1}^{(0,0)}(x) + \frac{n}{\sqrt{2n-1}\sqrt{2n+1}}P_{n-1}^{(0,0)}(x), \quad \forall n \geq 0,$$

$$P_{-1}^{(0,0)}(x) = 0, \quad P_0^{(0,0)}(x) = \sqrt{\frac{1}{2}}, \quad P_1^{(0,0)}(x) = \sqrt{\frac{3}{2}}x.$$

Por consiguiente se tienen las sucesiones $\{\zeta_n\}_{k=0}^n = \frac{n}{\sqrt{2n-1}\sqrt{2n+1}}$ y $\{\eta_k\}_{k=0}^n = 0$.

1.3.4. Resultados numéricos y análisis de resultados

En los Apéndices A.2, A.3 y A.4 de esta Memoria se encuentra la implementación en lenguaje Matlab de los algoritmos para el cálculo de ceros de los polinomios ortogonales de las tres familias particulares descritas anteriormente mediante un problema de autovalores (Teorema 1.1.14).

En los Cuadros 1.1 - 1.3 se encuentran los resultados obtenidos para algunos valores de n .

n	Ceros de Chebyshev de primera especie
2	$-0.7071067812 < 0.7071067812$
3	$-0.8660254038 < 0 < 0.8660254038$
4	$-0.9238795325 < -0.3826834324 < 0.3826834324 < 0.9238795325$
5	$-0.9510565163 < -0.5877852523 < 0 < 0.5877852523 < 0.9510565163$
6	$-0.9659258263 < -0.7071067812 < -0.2588190451 < 0.2588190451 < 0.7071067812 < 0.9659258263$
7	$-0.9749279122 < -0.7818314825 < -0.4338837391 < 0 < 0.4338837391 < 0.7818314825 < 0.9749279122$
8	$-0.9807852804 < -0.8314696123 < -0.5555702330 < -0.1950903220 < 0.1950903220 < 0.5555702330 < 0.8314696123 < 0.9807852804$
9	$-0.9848077530 < -0.8660254038 < -0.6427876097 < -0.3420201433 < 0 < 0.3420201433 < 0.6427876097 < 0.8660254038 < 0.9848077530$

Cuadro 1.1: Ceros de los polinomios de Chebyshev de primera especie

n	Ceros de Chebyshev de segunda especie
2	$-0.5000000000 < 0.5000000000$
3	$-0.7071067812 < 0 < 0.7071067812$
4	$-0.8090169944 < -0.3090169944 < 0.3090169944 < 0.8090169944$
5	$-0.8660254038 < -0.5000000000 < 0 < 0.5000000000 < 0.8660254038$
6	$-0.9009688679 < -0.6234898019 < -0.2225209340 < 0.2225209340 < 0.6234898019 < 0.9009688679$
7	$-0.9238795325 < -0.7071067812 < -0.3826834324 < 0 < 0.3826834324 < 0.7071067812 < 0.9238795325$
8	$-0.9396926208 < -0.7660444431 < -0.5000000000 < -0.1736481777 < 0.1736481777 < 0.5000000000 < 0.7660444431 < 0.9396926208$
9	$-0.9510565163 < -0.8090169944 < -0.5877852523 < -0.3090169944 < 0 < 0.3090169944 < 0.5877852523 < 0.8090169944 < 0.9510565163$

Cuadro 1.2: Ceros de los polinomios de Chebyshev de segunda especie

n	Ceros de Legendre
2	$-0.5773502692 < 0.5773502692$
3	$-0.7745966692 < 0 < 0.7745966692$
4	$-0.8611363116 < -0.3399810436 < 0.3399810436 < 0.8611363116$
5	$-0.9061798459 < -0.5384693101 < 0 < 0.5384693101 < 0.9061798459$
6	$-0.9324695142 < -0.6612093865 < -0.2386191861 < 0.2386191861 < 0.6612093865 < 0.9324695142$
7	$-0.9491079123 < -0.7415311856 < -0.4058451514 < 0 < 0.4058451514 < 0.7415311856 < 0.9491079123$
8	$-0.9602898565 < -0.7966664774 < -0.5255324099 < -0.1834346425 < 0.1834346425 < 0.5255324099 < 0.7966664774 < 0.9602898565$
9	$-0.96816023951 < -0.83603110733 < -0.61337143270 < -0.32425342340 < 0 < 0.32425342340 < 0.61337143270 < 0.83603110733 < 0.96816023951$

Cuadro 1.3: Ceros de los polinomios de Legendre

De los resultados obtenidos caben analizarse las siguientes propiedades:

- Los autovalores de la matriz de Jacobi son los ceros de los polinomios ortonormales. Véase la Sección 1.1.5.
- Los polinomios ortonormales tienen sus ceros en $(-1, 1)$, simples y distintos. Véase la Sección 1.1.2.
- Los ceros de las familias de polinomios ortogonales se entrelazan. Véase la Sección 1.1.6.
- De los ejemplos numéricos anteriores se aprecia que si σ es una función peso simétrica ($\sigma(x) = \sigma(-x)$, $\forall x \in [-1, 1]$), entonces los polinomios ortogonales tendrán sus ceros distribuidos también simétricamente. Más concretamente, los polinomios ortogonales pares serán funciones pares y los polinomios ortogonales impares serán funciones impares:

$$p_{2n}(x) = p_{2n}(-x) \quad y \quad p_{2n+1}(x) = xq_{2n}(x) \quad con \quad q_{2n}(x) = q_{2n}(-x).$$

En efecto, sea $p_{2n}(x)$ el polinomio ortogonal mónico de grado $2n$, es decir, $\langle p_{2n}(x), x^s \rangle_\sigma = 0$, $\forall s = 0, 1, \dots, 2n - 1$. Haciendo ahora el cambio de variable $t = -x$ y dado que σ es función peso simétrica nos proporciona

$$0 = \int_{-1}^1 p_{2n}(-t)(-t)^s \sigma(-t) dx = (-1)^s \int_{-1}^1 p_{2n}(-t)t^s \sigma(t) dx.$$

Por lo tanto, $p_{2n}(-x)$ es un polinomio mónico de grado $2n$ que cumple las mismas condiciones de ortogonalidad que $p_{2n}(x)$. Se concluye por tanto que $p_{2n}(x) = p_{2n}(-x)$ y el polinomio tendrá por lo tanto sólo potencias pares. El caso impar se demuestra de la misma forma.

Capítulo 2

Integración numérica

En este capítulo plantearemos el cálculo aproximado del valor de la integral definida de una función f con respecto a una función peso σ en $[-1,1]$ mediante el uso de fórmulas de cuadratura. Empezaremos estudiando algunos métodos particulares de aproximación, para distribuciones de nodos equiespaciadas, conocidos como reglas de Newton-Cotes, para diversos grados de interpolación: reglas de los rectángulos, trapezoidales y de Simpson. Además se estudiarán las fórmulas de tipo interpolatorio, en las que sin imponer condición alguna a los nodos, la aproximación se obtiene integrando directamente el correspondiente polinomio interpolador de Lagrange. La aproximación se acentuará con la caracterización de las fórmulas Gaussianas, que serán de tipo interpolatorio con el máximo grado de exactitud polinómico posible y cuyos nodos coincidirán con los ceros de los correspondientes polinomios ortogonales. El problema clásico de fijar nodos de antemano en la fórmula de cuadratura (Gauss-Radau y Gauss-Lobatto) será también abordado. Quedará claro por tanto que los resultados obtenidos en el Capítulo 1 serán ahora fundamentales para la obtención de fórmulas de cuadratura. Las fórmulas Gaussianas también se tratarán mediante un análisis riguroso desde un punto de vista computacional, transformando el problema de construir la fórmula de cuadratura de una manera eficiente en uno de cálculo de autovalores y autovectores de las matrices de Jacobi asociadas. Como en todo proceso numérico, se establecerán también resultados sobre cotas de los errores cometidos.

Por una fórmula de cuadratura se entiende una expresión de la forma:

$$I_n(f) := \sum_{k=1}^n \lambda_k f(x_k), \quad x_k \neq x_j \quad \text{si} \quad k \neq j, \quad x_k \in (-1, 1), \quad \forall k = 1, \dots, n, \quad (2.1)$$

donde x_k y λ_k son los denominados “nodos” y “pesos” de la fórmula de cuadratura, respectivamente. Vemos por tanto que se trata simplemente de una combinación lineal de valores del integrando.

En la primera sección abordaremos unas reglas muy particulares en la que partiremos de una distribución equiespaciada de nodos del intervalo de integración.

2.1. Fórmulas de Newton-Cotes

Comenzamos estudiando una familia particular de fórmulas de cuadratura conocidas como fórmulas de Newton-Cotes en las que partimos de una distribución de nodos equiespaciada en el intervalo de integración y que están basadas en la interpolación en cada subintervalo por polinomios interpolatorios de grado hasta 3. Éstas se clasifican en:

- Cerradas : cuando los extremos del intervalo son nodos;
- Abiertas : cuando no se toman los extremos del intervalo como nodos.

En este apartado trabajaremos con la función peso $\sigma \equiv 1$ definida en un intervalo general $[a, b]$. Dividiendo el intervalo $[a, b]$ en n segmentos iguales de amplitud $h = \frac{b-a}{n}$ se obtienen los puntos $a, a+h, a+2h, \dots, a+nh = b$. Seguidamente se construye la fórmula de cuadratura usando los puntos anteriores como nodos. Más concretamente, usando la linealidad del operador \int se aproxima el valor de la integral definida como la suma de las integrales definidas sobre todos los intervalos. De este modo, las fórmulas de Newton-Cotes pueden ser:

- Simple: trabajando directamente en todo el intervalo $[a, b]$;
- Compuestas: aumentando el número de intervalos en los que se divide la función a partir de subdivisiones y procediendo a la suma de éstos. En estos casos la precisión aumenta a costa de la eficiencia del método y a posibles errores de redondeo.

A continuación pasamos a describir estas fórmulas, dando cotas del correspondiente error, basándonos en la interpolación polinómica de hasta grado 3:

Interpolación de grado 0

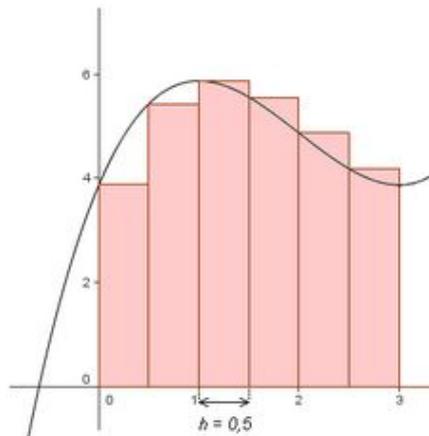
Regla del rectángulo extremo izquierdo

Simple: f se aproxima por el polinomio de grado 0 (constante) que pasa por el punto $(a, f(a))$.

$$\int_a^b f(x) dx \approx (b-a)f(a).$$

Compuesta:

$$\int_a^b f(x) dx \approx \frac{(b-a)}{n} \sum_{k=0}^{n-1} f(x_k).$$



Regla del rectángulo extremo derecho

Simple: f se aproxima por el polinomio de grado 0 (constante) que pasa por el punto $(b, f(b))$.

$$\int_a^b f(x)dx \approx (b-a)f(b).$$

Compuesta:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n} \sum_{k=0}^{n-1} f(x_{k+1}).$$

Regla del punto medio

Simple: f se aproxima por el polinomio de grado 0 (constante) que pasa por el punto $(\frac{a+b}{2}, f(\frac{a+b}{2}))$.

$$\int_a^b f(x)dx \approx (b-a)f\left(\frac{a+b}{2}\right).$$

Compuesta:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n} \sum_{k=0}^{n-1} f\left(\frac{x_k + x_{k+1}}{2}\right).$$

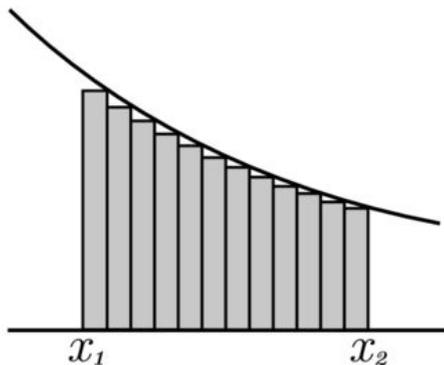


Figura 2.1: Regla del rectángulo derecho

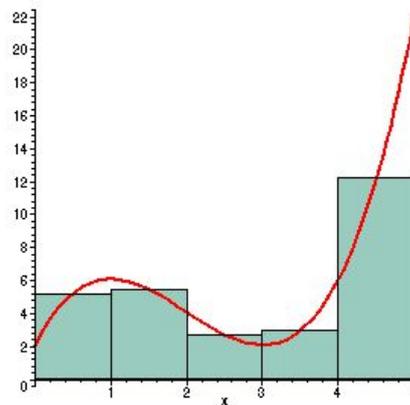


Figura 2.2: Regla del punto medio

Interpolación de grado 1

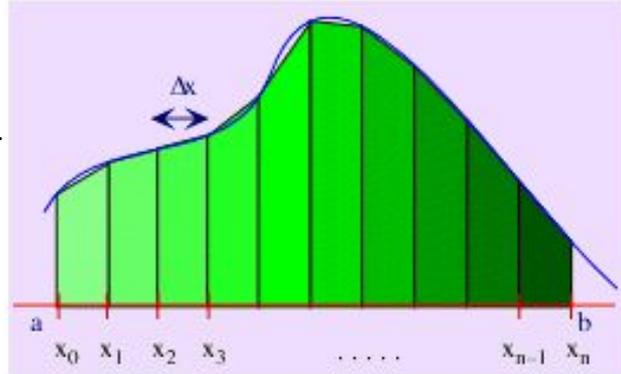
Regla trapezoidal

Simple: f se aproxima por el polinomio de grado 1 que pasa por los puntos $(a, f(a))$ y $(b, f(b))$.

$$\int_a^b f(x)dx \approx (b-a) \left[\frac{1}{2}f(a) + \frac{1}{2}f(b) \right].$$

Compuesta:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n} \left(\frac{f(x_0)}{2} + \sum_{k=1}^{n-1} f\left(x_0 + k \frac{b-a}{n}\right) + \frac{f(x_n)}{2} \right).$$



Interpolación de grado 2

Regla de Simpson un tercio

Simple: f se aproxima por un polinomio de grado 2.

$$\int_a^b f(x)dx \approx \frac{b-a}{3} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right].$$

Compuesta: asumimos que el número de subintervalos es $2n$.

$$\int_a^b f(x)dx \approx \frac{(b-a)}{6n} [f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 2f(x_{2n-2}) + 4f(x_{2n-1}) + f(x_{2n})].$$

Interpolación de grado 3

Regla de Simpson de los tres octavos

Simple: f se aproxima por un polinomio de grado 3.

$$\int_a^b f(x)dx \approx \frac{b-a}{8} \left[f(a) + 3f\left(a + \frac{1}{3}(b-a)\right) + 3f\left(a + \frac{2}{3}(b-a)\right) + f(b) \right].$$

Compuesta: asumimos que el número de subintervalos es $3n$.

$$\int_a^b f(x)dx \approx \frac{3(b-a)}{8n} \left[f(a) + 3 \sum_i f(x_i) + 3 \sum_j f(x_j) + 2 \sum_k f(x_k) + f(b) \right],$$

$$i = 3q+1, \quad j = 3q+2, \quad \text{con } q = 0, \dots, n-1; \quad k = 3l \quad \text{con } l = 1, \dots, n-1.$$

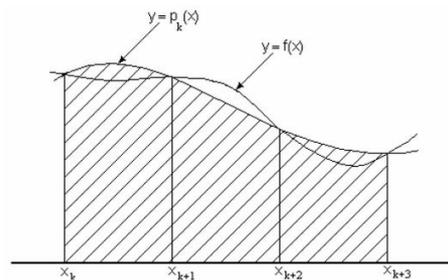
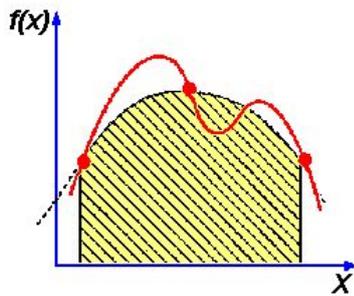


Figura 2.3: Regla de Simpson un tercio

Figura 2.4: Regla de Simpson tres octavos

A continuación daremos con carácter ilustrativo dos ejemplos numéricos en el que dos integrales concreta son aproximadas empleando las fórmulas de Newton-Cotes. El código implementado en lenguaje `Matlab` que se ha empleado se encuentra en el Apéndice A.5.

Ejemplo 2.1.1. *Uso de las fórmulas de Newton-Cotes para el cálculo aproximado de las siguientes integrales (cuyos valores exactos son conocidos):*

a) $I_1 = \int_1^{\frac{5}{2}} (x^3 - 3,3x^2 + 3x)dx = 1,3031250000.$

b) $I_2 = \int_1^3 (e^{1,7x})dx = 93,2635058283.$

- *Reglas simples*

a)

Ext.Izqdo.	Ext.Dcho.	Pto.Med.
1.0500000000	3.7500000000	0.7546875000
Trapez.	Simpson1/3	Simpson3/8
2.4000000000	1.3031250000	1.3031250000

Cuadro 2.1: Aproximaciones numéricas simples a la integral I_1 . Al ser reglas simples en un intervalo donde el recorrido del integrando es muy amplio, los resultados obtenidos son pobres.

b)

Ext.Izqdo.	Ext.Dcho.	Pto.Med.
10.9478947835	328.0438145998	59.9282000948
Trapez.	Simpson1/3	Simpson3/8
169.4958546916	96.4507516271	94.7316509544

Cuadro 2.2: Aproximaciones numéricas simples a la integral I_2 . Se observan resultados similares a los obtenidos para I_1 .

- *Reglas compuestas*

a)

n	Ext.Izqdo	Ext.Dcho	Pto.Med.	Trapez.
2	0.9023437500	2.2523437500	1.1660156250	1.5773437500
3	0.9750000000	1.8750000000	1.2421875000	1.4250000000
4	1.0341796875	1.7091796875	1.2688476563	1.3716796875
5	1.0770000000	1.6170000000	1.2811875000	1.3470000000
6	1.1085937500	1.5585937500	1.2878906250	1.3335937500
7	1.1326530612	1.5183673469	1.2919323980	1.3255102041
8	1.1515136719	1.4890136718	1.2945556671	1.3202636719
9	1.1666666667	1.4666666667	1.2963541667	1.3166666667
10	1.1790937500	1.4490937500	1.2976406250	1.3140937500

Cuadro 2.3: Aproximaciones numéricas compuestas a la integral I_1 . Los resultados obtenidos mejoran considerablemente a los de las reglas simples.

n	Simpson1/3	Simpson3/8
$\forall n \geq 3$	1.3031250000	1.3031250000

Cuadro 2.4: Aproximaciones numéricas compuestas a la integral I_1 , la cual presenta un integrando polinómico de grado 3 ($f \in \mathbb{P}_3$, por lo que las fórmulas de Simpson son exactas para cualquier número de nodos $n \geq 3$).

n	Ext.Izqdo	Ext.Dcho	Pto.Med.	Trapez.
2	35.4380474391	193.9860073473	82.9125161294	114.7120273932
3	50.1894647336	155.8881046724	88.4530413940	103.0387847030
4	59.1752817842	138.4492617383	90.5139440981	98.8122717612
b) 5	65.1202731764	128.5394571397	91.4905708960	96.8298651580
6	69.3212530638	122.1705730332	92.027263407	95.7459130485
7	72.4401759044	117.7395930210	92.3530011930	95.0898844627
8	74.8446129411	114.4816029182	92.5652810685	94.6631079297
9	76.7536197260	111.9864997056	92.7112140620	94.3700597158
10	78.3054220362	110.0150140179	92.8157966242	94.1602180270

Cuadro 2.5: Aproximaciones numéricas compuestas a la integral I_2 . Los resultados obtenidos mejoran considerablemente a los de las reglas simples.

n	Simpson1/3	Simpson3/8
6	93.3149558303	93.3751487554
12	93.2668132875	93.2708777861
18	93.2641626133	93.2649773690
24	93.2637140248	93.2639731562
30	93.2635911789	93.2636975742
36	93.2635470081	93.2635983846
42	93.2635280624	93.2635558160
48	93.2635188639	93.2635351409
54	93.2635139674	93.2635241326

Cuadro 2.6: Aproximaciones numéricas compuestas a la integral I_2 . Los resultados obtenidos mejoran considerablemente a los de las reglas simples. Para considerar conjuntamente las reglas de Simpson 1/3 y 3/8 tomamos n múltiplo de 6.

Destaquemos que el error de aproximación depende del método empleado, así como de la finura de la partición. A continuación expondremos unas estimaciones de los errores cometidos en las fórmulas de Newton-Cotes, las cuales aparecen demostradas en [4, Sección 6.3, pp. 92-98].

Error en las fórmulas de Newton-Cotes

Definamos el módulo del resto como

$$|R(f)| = \left| \int_a^b f(x)dx - I_n(f) \right|,$$

donde $I_n(f)$ es alguna de las fórmulas de Newton-Cotes.

- Restos de la regla del rectángulo extremo izquierdo
 - Simple: $|R(f)| = \frac{(b-a)^2}{2} |f'(\xi)|$, $\xi \in (a, b)$.
 - Compuesta: $|R(f)| \leq \frac{n}{2} \left(\frac{b-a}{n}\right)^2 \max |f'(\xi)|$, $\xi \in (a, b)$.
- Restos de la regla del rectángulo extremo derecho
 - Simple: $|R(f)| = \frac{(b-a)^2}{2} |f'(\xi)|$, $\xi \in (a, b)$.
 - Compuesta: $|R(f)| \leq \frac{n}{2} \left(\frac{b-a}{n}\right)^2 \max |f'(\xi)|$, $\xi \in (a, b)$.
- Restos de la regla del punto medio
 - Simple: $|R(f)| = \frac{(b-a)^2}{2} |f'(\xi)|$, $\xi \in (a, b)$.
 - Compuesta: $|R(f)| \leq \frac{n}{2} \left(\frac{b-a}{n}\right)^2 \max |f'(\xi)|$, $\xi \in (a, b)$.
- Resto de la regla trapezoidal
 - Simple: $|R(f)| = \frac{(b-a)^3}{12} |f''(\xi)|$, $\xi \in (a, b)$.
 - Compuesta: $|R(f)| = \frac{(b-a)^3}{12n^2} |f''(\xi)|$, $\xi \in (a, b)$.
- Restos de la regla de Simpson un tercio
 - Simple: $|R(f)| = \frac{(b-a)^5}{90} |f^{(4)}(\xi)|$, $\xi \in (a, b)$.
 - Compuesta: $|R(f)| = \frac{(b-a)^5}{180n^4} |f^{(4)}(\xi)|$, $\xi \in (a, b)$.
- Restos de la regla de Simpson tres octavos
 - Simple: $|R(f)| = \frac{(b-a)^5}{6480} |f^{(4)}(\xi)|$, $\xi \in (a, b)$.
 - Compuesta: $|R(f)| = \frac{(b-a)^5}{80n^4} |f^{(4)}(\xi)|$, $\xi \in (a, b)$.

Las fórmulas de Newton-Cotes presentan un mínimo coste computacional, pero también tienen una baja precisión, por lo que para obtener buenas estimaciones es necesario que n tome valores grandes. Con el fin de mejorar las estimaciones obtenidas sin necesidad de incrementar demasiado el valor de n , introduciremos en la siguiente sección las fórmulas de tipo interpolatorio.

2.2. Fórmulas de cuadratura de tipo interpolatorio

En esta sección profundizaremos en las fórmulas de cuadratura haciendo uso de la interpolación de Lagrange estudiada en la Sección 1.2 del Capítulo 1 y veremos la expresión que deben tener los pesos a partir de un conjunto de nodos fijos.

El concepto de cuadratura se basa en la búsqueda del valor numérico de la integral definida de una función de una variable sobre un intervalo. Teniendo presente la Sección 1.1.1 del Capítulo 1, asumiremos que el integrando se factoriza como producto de una función suave por una función peso en el intervalo cerrado $[-1, 1]$ que contendrá las posibles singularidades próximas al intervalo de integración. Supongamos que dados n nodos distribuidos en el intervalo $(-1, 1)$ sin ser necesariamente equiespaciados (ahora no tienen por qué estar a la misma distancia), $-1 < x_1 < x_2 < \dots < x_n < 1$, se construye el polinomio que interpola a f en esos puntos. Este planteamiento da lugar a la siguiente expresión:

$$f(x) = P(x) + R(x), \quad (2.2)$$

siendo $P(x) = \sum_{k=1}^n \frac{w(x)}{(x-x_k)w'(x_k)} f(x_k)$ el polinomio de interpolación de Lagrange, con $w(x)$ el polinomio nodal y $R(x)$ el correspondiente resto de Lagrange. Por consiguiente, una fórmula de cuadratura del tipo interpolatorio se introduce integrando la expresión (2.2):

$$I_\sigma(f) = \int_{-1}^1 f(x)\sigma(x)dx = \int_{-1}^1 P(x)\sigma(x)dx + \int_{-1}^1 R(x)\sigma(x)dx.$$

Seguidamente, despreciando la integral del resto se obtiene una aproximación a la integral de partida, en forma de fórmula de cuadratura, que recibe el nombre de “tipo interpolatorio” dado que se ha obtenido directamente integrando el polinomio interpolador de Lagrange:

$$\begin{aligned} \int_{-1}^1 f(x)\sigma(x)dx &\approx \int_{-1}^1 P(x)\sigma(x)dx = \int_{-1}^1 \sum_{k=1}^n \frac{w(x)}{(x-x_k)w'(x_k)} f(x_k)\sigma(x)dx \\ &= \sum_{k=1}^n \left(\int_{-1}^1 \frac{w(x)\sigma(x)dx}{(x-x_k)w'(x_k)} \right) f(x_k) = \sum_{k=1}^n \lambda_k f(x_k) \end{aligned}$$

lo que da lugar a que los pesos de la fórmula de cuadratura (2.1) admitan la expresión

$$\lambda_k = \int_{-1}^1 \frac{w(x)}{(x-x_k)w'(x_k)} \sigma(x)dx. \quad (2.3)$$

A continuación plantearemos otro enfoque para la determinación de los pesos en las fórmulas interpolatorias y estudiaremos el sistema que se forma a partir del criterio de exactitud en \mathbb{P}_{n-1} (espacio de dimensión n). En este caso, los n nodos carecen de condición alguna, salvo que están fijos y son distintos, por lo que será necesario buscar los n pesos,

imponiendo que la fórmula sea exacta para los monomios $f(x) = 1, x, x^2, \dots, x^{n-1}$:

$$\begin{cases} c_0 &= \lambda_1 &+ \lambda_2 &+ \dots &+ \lambda_n \\ c_1 &= \lambda_1 x_1 &+ \lambda_2 x_1 &+ \dots &+ \lambda_n x_n \\ \vdots & & & & \\ c_{n-1} &= \lambda_1 x_1^{n-1} &+ \lambda_2 x_1^{n-1} &+ \dots &+ \lambda_n x_n^{n-1}, \end{cases}$$

siendo $c_k = \int_{-1}^1 x^k \sigma(x) dx$ los momentos de orden k , $\forall k = 0, 1, \dots, n-1$. Este sistema se puede expresar matricialmente como:

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix}.$$

Esto da lugar a un sistema cuya matriz asociada es de tipo Vandermonde, por lo que su determinante es no nulo. Esto implica que existen únicos pesos $\lambda_1, \dots, \lambda_n$ tales que se cumple:

$$I_\sigma(f) = I_n(f), \quad \forall f \in \mathbb{P}_{n-1}$$

El siguiente resultado muestra que las dos vías explicadas anteriormente para obtener los pesos λ_k conducen a la misma solución. En [2, pp. 74-75] se comprueba que realmente los dos métodos son idénticos:

Proposición 2.2.1. *Considérense los siguientes dos métodos para obtener $\lambda_1, \lambda_2, \dots, \lambda_n$:*

- *Interpoliar $f(x)$ en los nodos x_1, \dots, x_n por un polinomio de grado menor o igual que $n-1$ e integrar el polinomio interpolador expresando el resultado en la forma (2.1).*
- *Elegir $\lambda_1, \lambda_2, \dots, \lambda_n$ de manera que el funcional definido por $R(f) = \int_{-1}^1 f(x)\sigma(x)dx - \sum_{k=1}^n \lambda_k f(x_k)$ se anule para $f(x) = 1, x, x^2, \dots, x^{n-1}$.*

Entonces las dos vías para determinar $\lambda_1, \lambda_2, \dots, \lambda_n$ llevan a la misma solución.

Dado que los dos métodos proporcionan la misma solución es preferible obtener los pesos λ_k mediante la Interpolación de Lagrange y no resolviendo el sistema anterior.

En relación al conjunto de funciones donde una fórmula de cuadratura es exacta, tenemos la siguiente

Definición 2.2.2. *Una fórmula del tipo (2.1) se dice que tiene grado de precisión k si integra exactamente a polinomios de grado menor o igual que k , y existe un polinomio de grado $k+1$ que no lo integra exactamente.*

Destaquemos que las fórmulas de tipo interpolatorio tienen grado de precisión al menos $n-1$, con unos nodos x_1, \dots, x_n carentes de restricciones salvo que sean diferentes y situados en el intervalo de integración $(-1,1)$. Como veremos más adelante, los nodos podrán ser elegidos adecuadamente para que el grado de precisión sea mayor. Lo deseable es que toda fórmula de cuadratura (2.1) verifique las siguientes propiedades:

- Estabilidad: dada por la acotación de los errores en la evaluación de f en cada nodo, esto es, asegurar la existencia de una constante $M \geq 0$ tal que $\sum_{j=1}^n |\lambda_j| \leq M$.
- Convergencia: $\lim_{n \rightarrow \infty} I_n(f) = \lim_{n \rightarrow \infty} \sum_{j=1}^n \lambda_j f(x_j) = \int_{-1}^1 f(x) \sigma(x) dx = I_\sigma(f)$, con f perteneciente a un cierto espacio de funciones.

Nótese que para el caso de una sucesión de fórmulas de tipo interpolatorio ambos conceptos son equivalentes en el espacio de las funciones continuas, tal y como se recoge en el siguiente:

Teorema 2.2.3. *Sea $I_n(f) = \sum_{j=1}^n \lambda_j f(x_j)$ una sucesión de fórmulas de tipo interpolatorio para $I_\sigma(f) = \int_{-1}^1 f(x) \sigma(x) dx$. Entonces:*

$$\lim_{n \rightarrow \infty} I_n(f) = I_\sigma(f), \quad \forall f \in C([-1, 1]) \quad \Leftrightarrow \quad \exists M \geq 0 : \forall n \quad \sum_{j=1}^n |\lambda_j| \leq M.$$

La demostración detallada de este resultado puede verse en [4, pp. 264-265].

A continuación daremos un importante teorema que relaciona las fórmulas de cuadratura interpolatorias con su exactitud.

Teorema 2.2.4. *Para que la fórmula de cuadratura (2.1) sea interpolatoria es necesario y suficiente que sea exacta en \mathbb{P}_{n-1} .*

Demostración. \Rightarrow) Cada polinomio $P \in \mathbb{P}_{n-1}$ con $\deg(P(x)) \leq n-1$ se puede representar en la forma

$$P(x) = \sum_{k=0}^n \frac{w(x)}{(x-x_k)w'(x)} P(x_k), \quad w(x) := \prod_{i=0}^n (x-x_i).$$

Tomando los coeficientes λ_k de la expresión (2.3) se verifica que la fórmula de cuadratura es exacta para $P(x)$.

(\Leftarrow Recíprocamente, el hecho de que la fórmula sea exacta en \mathbb{P}_{n-1} implica que:

$$\int_{-1}^1 \sum_{k=1}^n \frac{w(x)}{(x-x_k)w'(x)} P(x_k) \sigma(x) dx = \sum_{k=1}^n \lambda_k f(x_k),$$

siendo $\lambda_k = \int_{-1}^1 \frac{w(x)}{(x-x_k)w'(x)} \sigma(x) dx$, por lo que es de tipo interpolatorio. \square

En la siguiente sección plantearemos un caso particular de fórmulas de cuadratura interpolatorias, concretamente daremos condiciones a los nodos y veremos cómo esta elección optimiza la fórmula, esto es, aumenta la dimensión del espacio donde es exacta.

2.3. Fórmulas de cuadratura Gaussianas

Anteriormente hemos recogido que el Teorema de Aproximación de Weierstrass motiva a buscar exactitud en espacios de funciones polinómicas. Las fórmulas de tipo interpolatorio, que se obtienen integrando el polinomio interpolador de Lagrange en los nodos de la fórmula de cuadratura, tienen grado de precisión $k = n - 1$. Como en las fórmulas de tipo interpolatorio no se ha impuesto condición alguna a los nodos, cabe plantearse si una elección apropiada de éstos permitiría aumentar el grado de precisión. En este sentido, veremos que las fórmulas Gaussianas son las únicas que alcanzarán el máximo grado de precisión alcanzable $k = 2n - 1$ (puede comprobarse que $k = 2n$ nunca podrá conseguirse). Además, en este caso los pesos son positivos, lo cual garantiza la convergencia y la estabilidad, de acuerdo con el Teorema 2.2.3:

$$\sum_{j=1}^n |\lambda_j| = \sum_{j=1}^n \lambda_j = I_n(1) = I_\sigma(1) = \int_{-1}^1 \sigma(x) dx = c_0 < \infty.$$

A partir de lo anterior se puede establecer que la exactitud de la fórmula de cuadratura lo determinará la elección de los nodos x_k . Nótese que a partir del Teorema del resto de Lagrange:

$$R_n(x) = \int_{-1}^1 \frac{w(x)}{n!} f^{(n)}(\xi) \sigma(x) dx = \frac{1}{n!} \int_{-1}^1 f^{(n)}(\xi) w(x) \sigma(x) dx, \quad \xi \in (-1, 1). \quad (2.4)$$

Tomando valor absoluto en (2.4) y acotando $|f^{(n)}(\xi)| \leq M_n$ se obtiene:

$$|R_n(x)| \leq \frac{M_n}{n!} \int_{-1}^1 |w(x)| \sigma(x) dx, \quad \xi \in (-1, 1),$$

lo cual representará una expresión integral para el resto.

A continuación se procederá al estudio de las fórmulas de cuadratura con un alto grado de precisión.

2.3.1. Fórmulas de cuadratura con máximo grado de precisión

La cuadratura Gaussiana consiste en escoger los puntos de evaluación de manera que la aproximación sea óptima. De esta manera, se trata de estudiar la fórmula de cuadratura (2.1)

$$\int_{-1}^1 f(x) \sigma(x) dx \approx \sum_{k=1}^n \lambda_k f(x_k)$$

para un n fijo, con los parámetros λ_k y nodos x_k a determinar de manera que sea de tipo interpolatorio. La mejor elección será la que maximice el grado de precisión de la fórmula.

Teorema 2.3.1. *Para que la fórmula (2.1) sea exacta para todos los polinomios de grado menor o igual a $(2n-1)$, es necesario y suficiente que sea interpolatoria y que el polinomio nodal w sea ortogonal con respecto a σ a todo polinomio P de grado menor que n , esto es:*

$$\int_{-1}^1 P(x)w(x)\sigma(x)dx = 0, \quad \forall P \in \mathbb{P}_{n-1}.$$

Demostración. En primer lugar probaremos la necesidad. Si (2.1) es exacta para todos los polinomios de grado menor o igual a $(2n-1)$, también será exacta para los polinomios de grado $n-1$, lo cual implica que es interpolatoria por el Teorema 2.2.4. Sea $P \in \mathbb{P}_{n-1}$ y $f = wP \in \mathbb{P}_{2n-1}$, con la expresión (2.1) interpolatoria. Como $f(x_k) = 0$ para todo $k \in \{1, 2, \dots, n\}$, se sigue que, $\int_{-1}^1 P(x)w(x)\sigma(x)dx = \sum_{k=1}^n \lambda_k f(x_k) = 0$. Esto prueba la necesidad de la condición de ortogonalidad.

Justifiquemos la suficiencia de las condiciones. Sea $f \in \mathbb{P}_{2n-1}$, dividamos f entre w y representétese f según:

$$f(x) = P(x)w(x) + r(x), \quad \deg(f(x)) \leq 2n-1, \quad \deg(P(x)) \leq n-1, \quad (2.5)$$

$$\deg(w(x)) = n, \quad \deg(r(x)) \leq n-1.$$

Evaluando la expresión anterior en x_k , $k \in \{1, 2, \dots, n\}$ se llega a que:

$$f(x_k) = P(x_k)w(x_k) + r(x_k) = r(x_k), \quad \text{pues } w(x_k) = 0,$$

e integrando la expresión (2.5):

$$\int_{-1}^1 f(x)\sigma(x)dx = \int_{-1}^1 P(x)w(x)\sigma(x)dx + \int_{-1}^1 r(x)\sigma(x)dx.$$

Ahora bien, por hipótesis la integral del medio es nula debido a la ortogonalidad. Por lo tanto, dado que la expresión (2.1) es interpolatoria y que el grado de r no es mayor que $(n-1)$ se concluye que:

$$\int_{-1}^1 f(x)\sigma(x)dx = \int_{-1}^1 r(x)\sigma(x)dx = \sum_{k=1}^n \lambda_k r(x_k) = \sum_{k=1}^n \lambda_k f(x_k).$$

Luego la fórmula (2.1) será exacta en \mathbb{P}_{2n-1} . \square

Nótese que la existencia de fórmulas con grado de precisión $2n-1$ está relacionada con la existencia del n -ésimo polinomio ortogonal con respecto a σ : los nodos de las fórmulas de cuadratura son los ceros de los polinomios ortogonales. En este trabajo se ha asumido que $\sigma(x) > 0$ ya que asegura la existencia de los polinomios ortogonales (existencia de ortogonalidad con respecto a una función peso sin signo). No obstante, la existencia de polinomios ortogonales no queda siempre garantizada, verbigracia, el caso de σ función peso con signo variante.

El siguiente teorema justifica que \mathbb{P}_{2n-1} es el máximo dominio de validez posible.

Teorema 2.3.2. *No existen fórmulas de cuadratura que sean exactas en \mathbb{P}_{2n} .*

Demostración. El polinomio $P = w^2 \in \mathbb{P}_{2n}$, cumple que $\sum_{k=1}^n \lambda_k P(x_k) = 0$ mientras que $\int_{-1}^1 P(x)\sigma(x)dx > 0$. \square

Con carácter más general, enunciamos el siguiente resultado que caracteriza a fórmulas de cuadratura con grado de precisión intermedios (véase [2, pp. 109-112]).

Teorema 2.3.3. *Sea $I_n(f) = \sum_{j=1}^n \lambda_j f(x_j)$ una fórmula de cuadratura con grado de precisión $n+k$ ($0 \leq k \leq n-1$). Entonces el polinomio nodal $w(x)$ satisface las condiciones (de ortogonalidad)*

$$\int_{-1}^1 x^j w(x)\sigma(x)dx = 0, \quad j = 0, 1, \dots, k. \quad (2.6)$$

Además, se puede comprobar que (2.6) también es suficiente en el sentido del siguiente:

Teorema 2.3.4. *Sea $I_n(f) = \sum_{j=1}^n \lambda_j f(x_j)$ una fórmula de cuadratura para $I_\sigma(f) = \int_{-1}^1 f(x)\sigma(x)dx$. Entonces, $I_n(f)$ tiene grado de precisión $n+k$ con $0 \leq k \leq n-1$ si y sólo si:*

- $I_n(f)$ es de tipo interpolatorio.
- El polinomio nodal $w(x)$ satisface las condiciones $\int_{-1}^1 x^j w(x)\sigma(x)dx = 0, \quad j = 0, 1, \dots, k$.

Demostración. Procédase como en [4, pp. 101-102] donde se considera el caso $k = n-1$. \square

Nota 2.3.5. *El teorema anterior permite clasificar las fórmulas de cuadratura atendiendo al valor de k , esto es:*

- si $k = 0$ la fórmula de cuadratura es de tipo interpolatorio;
- si $k = n-1$ son fórmulas Gaussianas;
- si $1 \leq k \leq n-2$ las fórmulas son intermedias y están vinculadas al concepto de “cuasi-ortogonalidad”.

Complementaremos esta nota con la siguiente

Definición 2.3.6. *Si $\sigma \geq 0$ y $\{p_k\}_{k=0}^\infty$ es una familia de polinomios ortogonales para σ , se dice que un polinomio $q_n \in \mathbb{P}_n$ es cuasi-ortogonal de grado k si existen constantes a_1, \dots, a_k tales que*

$$q_n = p_n + a_1 p_{n-1}, \dots, a_k p_{n-k}, \quad a_k \neq 0.$$

Como consecuencia del Teorema 2.3.4 surge la siguiente conclusión:

Corolario 2.3.7. Una fórmula de cuadratura $I_n(f) = \sum_{j=1}^n \lambda_j f(x_j)$ tiene grado de precisión $2n - 1$ (el máximo alcanzable) si y sólo si:

- $I_n(f)$ es de tipo interpolatorio.
- El polinomio nodal w coincide, salvo factor multiplicativo, con el mismo polinomio ortogonal con respecto a σ en $[-1, 1]$.

El corolario anterior permite caracterizar la existencia y unicidad de las fórmulas de cuadratura con máximo grado de precisión, llamadas fórmulas de Gauss-Christoffel o simplemente fórmulas Gaussianas. Seguidamente, estudiemos su construcción, considerando el sistema de polinomios ortogonales normalizado (ortonormales) $P_k(x)$, $k \in \{1, 2, \dots, n\}$, en $[-1, 1]$ con respecto a la función peso σ . En este caso se puede observar que el polinomio de grado n de este sistema, esto es, P_n puede diferir de w en una constante. Además, las raíces de $P_n(x)$ proporcionan los nodos de la fórmula Gaussiana de n puntos. Seguidamente, profundizaremos en la expresión de los coeficientes λ_k , los cuales están determinados por:

$$\lambda_k = \int_{-1}^1 \frac{w(x)}{(x - x_k)w'(x_k)} \sigma(x) dx = \int_{-1}^1 \frac{P_n(x)}{(x - x_k)P_n'(x_k)} \sigma(x) dx. \quad (2.7)$$

Para calcular (2.7) se usa la identidad de Christoffel-Darboux sustituyendo $t = x_k$ y además, dividiendo por $x - x_k$:

$$\sum_{s=0}^{n-1} P_s(x)P_s(x_k) = -\frac{a_n}{a_{n+1}} \frac{P_n(x)P_{n+1}(x_k)}{x - x_k}, \text{ con } a_n \text{ coeficiente en } x^n \text{ de } P_n(x). \quad (2.8)$$

A continuación multiplicamos la ecuación (2.8) por σ e integramos sobre $[-1, 1]$. La integral $P_s(x_k) \int_{-1}^1 P_s(x)\sigma(x)dx$ es cero para todo $1 \leq s \leq n - 1$ por la condición de ortogonalidad de $P_s(x)$ y es 1 para $s = 0$ por la normalización de $P_0(x)$. De este modo:

$$1 = -\frac{a_n}{a_{n+1}} P_{n+1}(x_k) \int_{-1}^1 \frac{P_n(x)}{x - x_k} \sigma(x) dx$$

y se concluye:

$$\lambda_k = -\frac{a_{n+1}}{a_n} \frac{1}{P_n'(x_k)P_{n+1}(x_k)}. \quad (2.9)$$

Nótese que expresión (2.9) se puede modificar utilizando la fórmula de recurrencia de los polinomios ortonormales. Sustituyendo las raíces x_k de $P_n(x)$ en x se tiene:

$$\frac{a_n}{a_{n+1}} P_{n+1}(x_k) + \frac{a_{n-1}}{a_n} P_{n-1}(x_k) = 0.$$

Así, la relación (2.9) queda en términos de polinomios ortogonales con grados n y $n - 1$:

$$\lambda_k = \frac{a_n}{a_{n-1}} \frac{1}{P_n'(x_k)P_{n-1}(x_k)}.$$

Por otro lado, es sabido que los pesos son positivos, lo cual queda reflejado en el posterior:

Teorema 2.3.8. Para $k = 1, \dots, n$ se tiene que

$$\lambda_k = \frac{1}{\sum_{j=0}^{n-1} P_j^2(x_k)}, \quad (2.10)$$

(véase [7, pp. 20-21]).

Es necesario resaltar que la positividad de los pesos garantiza la convergencia en la integración. Además, la expresión (2.10) permite relacionar las fórmulas Gaussianas como un problema de autovalores. Para ello, haciendo uso de la Sección 1.1.5 recordamos que los nodos de las fórmulas Gaussianas son los autovalores de la matriz de Jacobi J_n dada por (1.12) y cuyos autovectores son $v_{j,n}$ dados por (1.13), que tienen la forma normalizada:

$$\tilde{v}_{j,n} = (\tilde{P}_0(x_{j,n}) \quad \tilde{P}_1(x_{j,n}) \quad \dots \quad \tilde{P}_{n-1}(x_{j,n}))^T \neq 0, \quad \tilde{v}_{j,n} \in \mathbb{R}^n, \quad \forall j = 1, \dots, n,$$

donde

$$\tilde{P}_0(x_{j,n}) = \frac{P_0(x_{j,n})}{\|P_0\|_2} = \frac{1}{\sqrt{\int_{-1}^1 \sigma(x) dx}} = \frac{1}{\sqrt{c_0}} \neq 0.$$

Dado que los nodos $x = x_{j,n}$ de la fórmula de Gauss son los autovalores de J_n , comprobemos que los coeficientes λ_j de dicha fórmula se expresan en términos de las primeras componentes de los vectores propios normalizados asociados a cada valor propio:

$$\|\tilde{v}_{j,n}\|_2 = 1; \quad \tilde{v}_{1,j} = \frac{\tilde{P}_0(x_j)}{\|v_j\|_2} = \frac{1}{\sqrt{c_0} \|v_j\|_2} \Rightarrow c_0(\tilde{v}_{1,j})^2 = \frac{1}{\|v_j\|_2^2} = \lambda_j, \quad (2.11)$$

siendo c_0 el momento de orden 0 y observamos que efectivamente (2.11) coincide con (2.10).

El siguiente teorema proporciona información sobre el carácter positivo de los pesos con fórmulas de cuadratura con grados de precisión intermedios.

Teorema 2.3.9. Si la fórmula de cuadratura (2.1) es exacta para todos los polinomios de grado menor o igual a $(2n - 2)$ entonces todos los coeficientes λ_k son positivos.

Demostración. Sea la función $f_i(x) = \left(\frac{w(x)}{x-x_i}\right)^2 \in \mathbb{P}_{2n-2}$, $\forall i = 1, \dots, n$. Nuestra fórmula de cuadratura es exacta para toda f_i . Además $f_i(x_k) = w'^2(x_i)\delta_{i,k}$, por lo que podemos calcular los pesos: $\int_{-1}^1 \left(\frac{w(x)}{x-x_i}\right)^2 \sigma(x) dx = \lambda_i w'^2(x_i)$ y despejando obtenemos $\lambda_i = \int_{-1}^1 \left(\frac{w(x)}{(x-x_i)w'(x_i)}\right)^2 \sigma(x) dx > 0$. \square

Seguidamente nos centraremos en el estudio del error cometido en la fórmula de cuadratura, teniendo presente que como el segmento de integración es $[-1, 1]$, se debe asumir que el producto $f\sigma$ es integrable.

Teorema 2.3.10. Si f tiene derivada continua de orden $2n$ en $[-1, 1]$, entonces existe un punto $\mu \in (-1, 1)$ para el cual el resto de la fórmula Gaussiana con n puntos viene dado por

$$R(f) := \int_{-1}^1 f(x)\sigma(x)dx - I_n(f) = \frac{f^{(2n)}(\mu)}{(2n)!} \int_{-1}^1 w^2(x)\sigma(x)dx. \quad (2.12)$$

Demostración. Sea $H(x)$ el polinomio interpolador de Hermite que sabemos que existe, véase [6, Sección 2.1.5] con $\deg(H(x)) \leq 2n - 1$ que satisface las condiciones: $H(x_k) = f(x_k)$, $H'(x_k) = f'(x_k)$. El resto $r(x) = f(x) - H(x)$ de la interpolación puede ser expresado como:

$$r(x) = \frac{f^{(2n)}(\xi)}{(2n)!} w^2(x),$$

donde ξ pertenece al segmento que contiene x y los nodos x_k . Por consiguiente:

$$\int_{-1}^1 f(x)\sigma(x)dx = \int_{-1}^1 H(x)\sigma(x)dx + \frac{1}{(2n)!} \int_{-1}^1 f^{(2n)}(\xi)w^2(x)\sigma(x)dx.$$

Precisemos que como la fórmula de cuadratura es exacta para todos los polinomios de grado menor o igual a $2n - 1$, es exacta para $H(x)$:

$$\int_{-1}^1 H(x)\sigma(x)dx = \sum_{k=1}^n \lambda_k H(x_k) = \sum_{k=1}^n \lambda_k f(x_k),$$

así, el resto de la fórmula de cuadratura es $R(f) = \frac{1}{(2n)!} \int_{-1}^1 f^{(2n)}(\xi)w^2(x)\sigma(x)dx$. \square

Finalizamos esta sección con un resultado de convergencia:

Teorema 2.3.11. Si $f \in C([-1, 1])$ entonces:

$$\lim_{n \rightarrow \infty} I_n(f) = \lim_{n \rightarrow \infty} \sum_{k=1}^n \lambda_{k,n} f(x_{k,n}) = \int_{-1}^1 f(x)\sigma(x)dx = I_\sigma(f).$$

Demostración. Dada la función f continua en $[-1, 1]$, para cada $\epsilon > 0$ se puede encontrar un polinomio P con la propiedad de que para todo $x \in [a, b]$ se tiene:

$$\|f(x) - P(x)\| < \epsilon, \quad (2.13)$$

siendo $\|\cdot\|$ una norma definida en $C([-1, 1])$. Luego:

$$\begin{aligned} \left\| \int_{-1}^1 f(x)\sigma(x)dx - \sum_{k=1}^n \lambda_{k,n} f(x_{k,n}) \right\| &\leq \left\| \int_{-1}^1 f(x)\sigma(x)dx - \int_{-1}^1 P(x)\sigma(x)dx \right\| \\ &+ \left\| \int_{-1}^1 P(x)\sigma(x)dx - \sum_{k=1}^n \lambda_{k,n} P(x_{k,n}) \right\| \\ &+ \left\| \sum_{k=1}^n \lambda_{k,n} P(x_{k,n}) - \sum_{k=1}^n \lambda_{k,n} f(x_{k,n}) \right\|. \end{aligned}$$

Sea $c_0 = \int_{-1}^1 \sigma(x)dx$ y por (2.13) se tiene que:

$$\left\| \int_{-1}^1 f(x)\sigma(x)dx - \int_{-1}^1 P(x)\sigma(x)dx \right\| \leq \epsilon c_0,$$

y

$$\left\| \sum_{k=1}^n \lambda_{k,n} P(x_{k,n}) - \sum_{k=1}^n \lambda_{k,n} f(x_{k,n}) \right\| \leq \epsilon \sum_{k=1}^n \lambda_{k,n} = \epsilon c_0.$$

Si el grado de polinomio $P(x)$ es m se tiene que $2n - 1 \geq m$, para un n suficientemente grande, lo que da lugar a:

$$\int_{-1}^1 P(x) \sigma(x) dx = \sum_{k=1}^n \lambda_{k,n} P(x_{k,n})$$

y por lo tanto para n se concluye:

$$\left\| \int_{-1}^1 f(x) \sigma(x) dx - \sum_{k=1}^n \lambda_{k,n} f(x_{k,n}) \right\| < 2\epsilon c_0.$$

□

Nota 2.3.12. *El resultado anterior también es válido si f es Riemann-integrable en $[-1, 1]$. Véase [5, pp. 427-442].*

2.3.2. Fórmulas de cuadratura con nodos preasignados

Para finalizar este Capítulo, consideremos el problema clásico de construir fórmulas de cuadratura con máximo grado de precisión alcanzable y nodos preasignados, esto es, expresiones del tipo:

$$\int_{-1}^1 f(x) \sigma(x) dx \approx \sum_{l=1}^m a_l f(y_l) + \sum_{k=1}^n b_k f(x_k) \quad (2.14)$$

donde los nodos y_l están fijos y las $m + 2n$ constantes a_l, b_k y x_k ($k = 1, \dots, n; l = 1, \dots, m$) deben ser determinadas para que la regla tenga el grado máximo de precisión (en este caso, $m + 2n - 1$), dado que por cada nodo que se fija, la dimensión del dominio de validez será uno menor. Definamos los polinomios:

$$r(x) = \prod_{l=1}^m (x - y_l) \quad y \quad s(x) = \prod_{k=1}^n (x - x_k). \quad (2.15)$$

La demostración de este resultado puede verse en [2, pp. 101-102] y caracteriza tales fórmulas en el caso de que existan.

Teorema 2.3.13. *La fórmula de cuadratura (2.14) es exacta para todos los polinomios de grado menor o igual que $m + 2n - 1$ si y sólo si:*

- *es exacta para todos los polinomios de grado menor o igual que $m + n - 1$;*
- $\int_{-1}^1 r(x) s(x) P_j(x) \sigma(x) dx = 0$ para todo $P_j \in \mathbb{P}_{n-1}$, con r y s dados por (2.15).

Nótese que del Teorema 2.3.13, la existencia de este tipo de fórmulas es equivalente a la existencia de polinomios ortogonales con respecto a una función peso de signo variante, algo que no siempre está garantizado. Sin embargo, hay dos situaciones particulares clásicas donde la elección de nodos fijos garantiza la positividad de la función peso:

- Integración de Gauss-Radau: sea $m = 1$, con $y_1 = a = -1$ ó $y_1 = b = 1$.
 - Fijando $y_1 = a = -1$ se obtienen los n nodos, los cuales son $y_1 = a = -1$ y los $n - 1$ ceros de P_{n-1} con respecto a la función peso positiva $\kappa(x) = (x - a)\sigma(x) = (x + 1)\sigma(x) \geq 0$. Esta fórmula es exacta en \mathbb{P}_{2n-2} .
 - Fijando $y_1 = b = 1$ se obtienen los n puntos, los cuales son $y_1 = b = 1$ y los $n - 1$ ceros de P_{n-1} con respecto a la función peso positiva $\kappa(x) = (b - x)\sigma(x) = (1 - x)\sigma(x) \geq 0$. Esta fórmula es exacta en \mathbb{P}_{2n-2} .

- Integración de Gauss-Lobatto: sea $m = 2$, con $y_1 = a = -1$ e $y_2 = b = 1$.
 - Se obtienen los n nodos, los cuales son $y_1 = a = -1$ y los $n - 2$ ceros de P_{n-2} con respecto a la función peso positiva $\kappa(x) = (x - a)(b - x)\sigma(x) = (x + 1)(1 - x)\sigma(x) = (1 - x^2)\sigma(x) \geq 0$. La fórmula es exacta en \mathbb{P}_{2n-3} .

Estas fórmulas tienen nodos distintos pues son los ceros de los polinomios ortogonales respecto a las funciones peso distintas.

En la próxima subsección, aportaremos un punto de vista práctico a los conocimientos dados anteriormente, particularizando con la obtención de nodos de fórmulas de cuadratura de tipo Gauss que usaremos para obtener estimaciones a algunas integrales definidas.

2.3.3. Resultados numéricos y análisis de resultados

En esta subsección trataremos el procedimiento de obtención de los pesos para diferentes fórmulas de cuadratura, para lo cual utilizaremos los códigos implementados en las secciones A.2, A.3 y A.4 de Apéndice A en lenguaje `Matlab`. Obsérvese que para ello debemos utilizar los ceros de los polinomios de las familias ortogonales que hemos hallado en la Sección 1.3. Los pesos se corresponden con la primera componente al cuadrado del autovector normalizado asociado a cada autovalor (cero del polinomio), multiplicado por el momento de orden cero. Nótese que cuando la función peso es de probabilidad, el momento de orden cero es uno. En este caso hemos usado diez decimales con el propósito de que la fórmula de cuadratura sea lo más precisa posible en el Ejemplo 2.3.14 posterior.

Pesos de los polinomios de Chebyshev de primera especie

n=2	
Nodos	Pesos
± 0.7071067812	1.5707963268

n=3	
Nodos	Pesos
± 0.8660254038	1.0471975512
0	1.0471975512

Cuadro 2.7: Pesos para n=4 y n=5 de Chebyshev de primera especie.

n=4	
Nodos	Pesos
± 0.9238795325	0.7853981634
± 0.3826834324	0.7853981634

n=5	
Nodos	Pesos
± 0.9510565163	0.6283185307
± 0.5877852523	0.6283185307
0	0.6283185307

Cuadro 2.8: Pesos para n=4 y n=5 de Chebyshev de primera especie.

n=6	
Nodos	Pesos
± 0.9659258263	0.5235987756
± 0.7071067812	0.5235987756
± 0.2588190451	0.5235987756

n=7	
Nodos	Pesos
± 0.9749279122	0.4487989505
± 0.7818314825	0.4487989505
± 0.4338837391	0.4487989505
0	0.4487989505

Cuadro 2.9: Pesos para n=6 y n=7 de Chebyshev de primera especie.

n=8	
Nodos	Pesos
± 0.9807852804	0.3926990817
± 0.8314696123	0.3926990817
± 0.5555702330	0.3926990817
± 0.1950903220	0.3926990817

n=9	
Nodos	Pesos
± 0.9848077530	0.3490658504
± 0.8660254038	0.3490658504
± 0.6427876097	0.3490658504
± 0.3420201433	0.3490658504
0	0.3490658504

Cuadro 2.10: Pesos para n=8 y n=9 de Chebyshev de primera especie.

Se observa que los pesos tienen la expresión $\frac{\pi}{n}$, independientemente del nodo.

Pesos de los polinomios de Chebyshev de segunda especie

n=2	
Nodos	Pesos
± 0.5000000000	0.7853981634

n=3	
Nodos	Pesos
± 0.7071067812	0.3926990817
0	0.7853981634

Cuadro 2.11: Pesos para n=2 y n=3 de Chebyshev de segunda especie.

n=4	
Nodos	Pesos
± 0.8090169944	0.2170787134
± 0.3090169944	0.5683194500

n=5	
Nodos	Pesos
± 0.8660254038	0.1308996939
± 0.5000000000	0.3926990817
0	0.5235987756

Cuadro 2.12: Pesos para n=4 y n=5 de Chebyshev de segunda especie

n=6	
Nodos	Pesos
± 0.9009688679	0.0844886909
± 0.6234898019	0.2743330561
± 0.2225209340	0.4265764164

n=7	
Nodos	Pesos
± 0.9238795325	0.0575094490
± 0.7071067812	0.1963495409
± 0.3826834324	0.3351896327
0	0.3926990817

Cuadro 2.13: Pesos para n=6 y n=7 de Chebyshev de segunda especie.

n=8	
Nodos	Pesos
± 0.9396926208	0.0408329477
± 0.7660444431	0.1442256008
± 0.5000000000	0.2617993878
± 0.1736481777	0.3385402271

n=9	
Nodos	Pesos
± 0.9510565163	0.0299995404
± 0.8090169944	0.1085393567
± 0.5877852523	0.2056199086
± 0.3090169944	0.2841597250
0	0.3141592654

Cuadro 2.14: Pesos para n=8 y n=9 de Chebyshev de segunda especie.

Pesos de los polinomios de Legendre

n=2	
Nodos	Pesos
± 0.5773502692	1.0000000000

n=3	
Nodos	Pesos
± 0.7745966692	0.555555555556
0	0.888888888886

Cuadro 2.15: Pesos para n=2 y n=3 de Legendre.

n=4	
Nodos	Pesos
± 0.8611363116	0.3478548451
± 0.3399810436	0.6521451549

n=5	
Nodos	Pesos
± 0.9061798459	0.2369268851
± 0.5384693101	0.4786286705
0	0.5688888889

Cuadro 2.16: Pesos para n=4 y n=5 de Legendre

n=6	
Nodos	Pesos
± 0.9324695142	0.1713244924
± 0.6612093865	0.3607615730
± 0.2386191861	0.4679139346

n=7	
Nodos	Pesos
± 0.9491079123	0.1294849662
± 0.7415311856	0.2797053915
± 0.4058451514	0.3818300505
0	0.4179591837

Cuadro 2.17: Pesos para n=6 y n=7 de Legendre.

n=8	
Nodos	Pesos
± 0.9602898565	0.1012285363
± 0.7966664774	0.2223810345
± 0.5255324099	0.3137066459
± 0.1834346425	0.3626837834

n=9	
Nodos	Pesos
± 0.96816023951	0.0812743884
± 0.83603110733	0.1806481607
± 0.61337143270	0.2606106964
± 0.32425342340	0.3123470770
0	0.3302393550

Cuadro 2.18: Pesos para n=8 y n=9 de Legendre.

A continuación aplicaremos los pesos anteriores con el objetivo de calcular la aproximación numérica a la integral de una función.

Ejemplo 2.3.14. *Determinar la aproximación a la integral de la función suave*

$$f(x) = \frac{1}{1+x^2} \quad (2.16)$$

en $[-1, 1]$, con respecto a los diferentes pesos $\sigma(x)$ específicos de cada familia de polinomios ortogonales. Así:

- $\int_{-1}^1 \frac{1}{1+x^2} \frac{1}{\sqrt{1-x^2}} dx = 2,2214414691 \dots$
- $\int_{-1}^1 \frac{1}{1+x^2} \sqrt{1-x^2} dx = 1,3012902846 \dots$
- $\int_{-1}^1 \frac{1}{1+x^2} dx = 1,5707963268 \dots$

Nodos	Cheby. 1ª especie	Cheby. 2ª especie	Legendre
2	2.0943951024	1.2566370614	1.5000000000
3	2.2439947526	1.3089969390	1.5833333334
4	2.2175948143	1.2999693739	1.5686274510
5	2.2221021208	1.3015169565	1.5711711712
6	2.2213281390	1.3012513950	1.5707317073
7	2.2214609140	1.3012969570	1.5708074534
8	2.2214381329	1.3012891398	1.5707944125
9	2.2214420415	1.3012904810	1.5707966559

Cuadro 2.19: Aproximaciones numéricas a la integral (2.16).

Las fórmulas de cuadratura utilizadas se ajustan perfectamente a la integral, proporcionando una mejor aproximación a medida que el número de nodos aumenta.

Capítulo 3

Conclusiones y futuras líneas de investigación

A lo largo de esta Memoria hemos podido profundizar en el cálculo aproximado de una integral definida con respecto a una función peso en un compacto de la recta real mediante los recursos de la teoría de ortogonalidad básica y las fórmulas de cuadratura. En el desarrollo de esta Memoria se han adquirido una serie de habilidades, las cuales se exponen a continuación:

- Búsqueda de información y manipulación de la misma, extrayendo resultados precisos acordes a la teoría estudiada.
- Procesamiento de textos científicos mediante la realización de continuos informes y borradores de contenido matemático.
- Análisis e interpretación de modelos de situaciones reales sencillas, utilizando las herramientas matemáticas más adecuadas a los fines que se persiguen.
- Utilización de software matemático `Matlab`, adquiriendo una actitud crítica con el procedimiento y cuestionando los resultados obtenidos. Asimismo, la validación de resultados teóricos, extrayendo conclusiones finales precisas y sintetizando la información.
- Extrapolación de resultados a otras disciplinas lo cual permite establecer que la teoría matemática numérica se extralimita de todo formulismo matemático a una realidad más concreta.
- Comunicación tanto por escrito como de forma oral de conocimientos, procedimientos, resultados e ideas matemáticas que reflejen la base teórica y permitan indagar en contenidos más profundos sobre este campo.
- Por último resaltar el enorme potencial que tiene la utilización de `LATEX`, `Beamer` y `Matlab` para la elaboración documental, la presentación y la creación de algoritmos, respectivamente.

Tras estas conclusiones finales, enunciamos a continuación una serie de problemas abiertos y futuras líneas de investigación relacionados con la *Integración Numérica* para los cuales esta Memoria podría servir como texto básico introductorio:

- **Fórmulas de cuadratura en la circunferencia unidad:** las fórmulas de cuadratura abordadas en esta Memoria se aplican en la resolución aproximada de una integral con respecto a una función peso definida sobre un intervalo de la recta real. Cuando el integrando es periódico parece razonable considerar fórmulas de cuadratura de tipo interpolatorio pero basadas no en polinomios ordinarios sino en polinomios trigonométricos, o en general, fórmulas de cuadratura sobre la *circunferencia unidad*, o fórmulas de Szegő. Este tipo de fórmulas aparecieron a finales de los años 80 y su estudio constituye un importante tema de investigación en la actualidad. Una futura línea de investigación sería la de extender resultados conocidos de la recta real a la circunferencia unidad, y viceversa, y la de caracterizar fórmulas de cuadratura en la circunferencia unidad para familias particulares de funciones peso.
- **Pre-asignación de nodos en una fórmula de cuadratura:** parece sorprendente que, siendo clásicas las fórmulas de Gauss-Radau y Gauss-Lobatto, la caracterización de fórmulas de cuadratura positivas con máximo grado de precisión y uno o dos nodos fijados de antemano no en los extremos sino en todo el intervalo de integración (incluyendo el interior) fueron obtenidas en el año 2010 precisamente por el profesor Ruymán Cruz Barroso en colaboración con los profesores Adhemar Bultheel y Marc Van Barel, de KU Leuven, Bélgica. Este nuevo tipo de fórmulas de cuadratura han demostrado ser de utilidad en la resolución numérica de Ecuaciones Diferenciales Ordinarias. Parece por tanto razonable el plantearse como de sumo interés la caracterización de fórmulas de cuadratura con un número arbitrario de nodos preasignados en todo el intervalo de integración.
- **Fórmulas de Szegő con nodos preasignados:** siguiendo las líneas del apartado anterior, otro problema abierto es la caracterización de fórmulas de cuadratura en la circunferencia unidad con un número de nodos arbitrario fijado de antemano. Cabe decir en este caso que la situación es completamente diferente a lo que ocurre en la recta real, siendo conocidos hasta la fecha la caracterización que fija como máximo dos nodos.
- **Fórmulas de cuadratura múltiples:** el desarrollo de la Teoría de Polinomios Ortogonales ha desarrollado en los últimos años su vertiente múltiple, en la que partiendo de varias funciones peso, se define la ortogonalidad de un polinomio con un grado determinado repartiendo condiciones de ortogonalidad entre las respectivas medidas. Como futura línea de investigación, tanto en la recta real como en la circunferencia unidad, podríamos incluir el uso de la ortogonalidad múltiple para la caracterización de fórmulas de cuadratura simultáneas, es decir, que una misma fórmula de cuadratura sea exacta de manera simultánea para diferentes integrales con respecto a funciones peso diferentes. A día de hoy se conoce muy poco sobre esta Teoría.

- **Álgebra Lineal Numérica:** hemos visto cómo las fórmulas de cuadratura abordadas en esta Memoria pueden computarse como un problema de autovalores y autovectores de matrices de Jacobi. Para otro tipo de fórmulas de cuadratura el proceso es el mismo, pero variando el tipo de matriz estructurada (por ejemplo, en el caso de circunferencia unidad, la matriz de Jacobi se transforma en una matriz de Hessenberg, o pentadiagonal). Sería interesante profundizar en el estudio de algoritmos del Álgebra Lineal Numérica para matrices estructuradas con el fin de obtener una computación eficiente de fórmulas de cuadratura.
- **Funciones Ortogonales Racionales:** tanto en la circunferencia unidad como en la recta real, se ha demostrado numéricamente que cuando deseamos aproximar una integral que contenga polos en las proximidades del recinto de integración, las fórmulas Gaussianas o de Szegő suelen dar malos resultados. Una alternativa que se ha desarrollado en los últimos años consiste en considerar funciones racionales cuyos polos simulen los polos del integrando, en vez de polinomios ordinarios (que son realmente funciones racionales con polos situados en el infinito). Como futura línea de investigación incluimos también el estudio de *Funciones Ortogonales Racionales*, y la caracterización de las fórmulas de cuadratura racionales asociadas.
- **Fórmulas de cuadratura con nodos múltiples:** entendiéndose bajo este nombre el considerar fórmulas de cuadratura que involucren en los nodos dados la información no solo de la función sino también de sus derivadas hasta ciertos órdenes (que en principio podrían variar en cada nodo), planteamos el estudio de las fórmulas de cuadratura asociadas que se pudieran obtener, así como de estimaciones del error, que permitan hacer comparaciones numéricas con las fórmulas usuales. Este tema es también un tema actual en investigación.

Apéndice A

Programación en Matlab

A.1. Polinomio interpolador de Lagrange

```
#####
# Archivo en Matlab:interpolacion_lagrange.m
#####
function interpolacion_lagrange

format long
X=[0, 3, 7, 9, 10] % Puntos dados
Y=[1, 3, -2, -1, 4] % Cantidades reales fijas
% Matlab tiene una función denominada polyfit(X,Y,n)
%dados (n+1)-nodos y (n+1)-valores, que construye directamente el polinomio
%interpolador. El objetivo de este código es comprobar mediante la fórmula
%de Lagrange que recuperamos el mismo polinomio interpolador.
P = polyfit(X,Y,4) %Polinomio que pasa por los puntos (X, Y):
%Así, el polinomio que ajusta en estos puntos es:
%P(x)= 0.010978835978836*x^4 -0.145767195767195*x^3 + 0.316534391534387*x^2
%+ 0.732539682539689*x + 1.000000000000007
P=[ 0.010978835978836 -0.145767195767195 0.316534391534387 0.732539682539689 1.000000000000007]
%Se puede observar que Matlab produce los coeficientes correctos de
%acuerdo con la teoría de interpolación. La función 'polyfit' emplea
%métodos del algebra lineal para resolver sistemas lineales de
%ecuaciones y así hallar los coeficientes.
%Utilicemos el método de Lagrange que proporcionará el mismo resultado
%salvo pequeñas variaciones
W=poly(X) % polinomio nodal W(x)
% Para manipular polinomios de forma simbólica se procede:
syms x
poly2sym(W,x)
%W(x)=1*x^5 -29*x^4 + 301*x^3 -1299*x^2 + 1890*x + 0
%Efectivamente: roots(W)= X
W=[ 1 -29 301 -1299 1890 0]
polyval(W,X(1))%Se comprueba que efectivamente una de sus raíces es X(1)
%Calculemos el polinomio interpolador de Lagrange
L(1,:)=poly([3, 7, 9, 10])/((0-3)*(0-7)*(0-9)*(0-10))
L(2,:)=poly([0, 7, 9, 10])/((3-0)*(3-7)*(3-9)*(3-10))
L(3,:)=poly([0, 3, 9, 10])/((7-0)*(7-3)*(7-9)*(7-10))
```

```

L(4,:)=poly([0, 3, 7, 10])/((9-0)*(9-3)*(9-7)*(9-10))
L(5,:)=poly([0, 3, 7, 9])/((10-0)*(10-3)*(10-7)*(10-9))

L=Y(1)*L(1,:)+Y(2)*L(2,:)+Y(3)*L(3,:)+Y(4)*L(4,:)+Y(5)*L(5,:)
sym x
pretty(poly2sym(L))
poly2sym(L)
%L= 0.010978835978836*x^4 -0.145767195767196*x^3 +0.316534391534392*x^2 +
%0.732539682539682*x + 1.000000000000000

%Evaluando los polinomios L y P en los valores del vector Y
YY1=polyval(L,Y)
%YY1=[1.914285714285714 3.000000000000000 2.142857142857144
%0.740740740740741 2.476190476190477]
YY2=polyval(P,Y)
%YY2=[1.914285714285724 3.000000000000009 2.142857142857113
%0.740740740740736 2.476190476190491]

%Dibujando ambos resultados para compararlos
SL=poly2sym(L)
SP=poly2sym(P)
%SL=83/7560*x^4-551/3780*x^3+2393/7560*x^2+923/1260*x+1
%SP=83/7560*x^4-1312954177079963/9007199254740992*x^3+356386041941053/11258
%99906842624*x^2+3299065441319845/4503599627370496*x+140737488355329/140737488355328
z=[-2:0.000001:10];
plot(z,83/7560*z.^4-551/3780*z.^3+2393/7560*z.^2+923/1260*z+1,'b')
end
#####

```

A.2. Problema de autovalores, autovectores y pesos para los polinomios de Chebyshev de primera especie

A.2.1. Método 1

```

#####
# Archivo en Matlab: chebyshev_de_primer_tipo.m
#####
function chebyshev_de_primer_tipo

%Procedimiento que permite hallar los ceros a partir de la construcción directa de la matriz
disp('*Ceros de polinomios de Chebyshev de primer tipo') ;
n=input('* Introduzca el número n>1 de ceros que desea hallar = ');

if n>=2
    J(1,2)=1/(sqrt(2));
    J(2,1)=1/(sqrt(2));
end
if n>2
    i=2 ;
    while i<n
        J(i,i+1)=1/2 ;
        J(i+1,i)=1/2 ;
    end
end

```

```

    i=i+1      ;
end          ;
end         ;
J           %J es la matriz de Jacobi de los polinomios de Chebyshev de primer tipo
[X,D]=eig(J) %D es matriz diagonal que contiene los autovalores de la matriz J
           %las columnas de X son autovectores de V
           %si se quieren los autovalores en forma de vector columna se
fprintf('Los autovalores son: \n')
aut0=diag(D) %autovalores
fprintf('Los autovectores por columnas son: \n')
autvect= X   %autovector de los autovalores
for i=1:n
    syms x
    X(:,i)           ; %autovector asociado a cada autovalor
    norma=norm(X(:,i),2) ; %estudio de la norma del autovector
    fprintf('El autovector normalizado es: \n')
    autvect2=X(:,i)./norma ; %autovector normalizado
    norma2=norm(autvect2,2) ; %compruebo que el vector sea unitario
    c_0=int(1/sqrt(1-x^2),-1,1) ; %momento de orden cero
    Peso0(i)=c_0*(autvect2(1)^2); %expresión de peso para función que no es de probabilidad
    Peso1(i)=eval(Peso0(i)) ; %peso
    fprintf('Los pesos son: \n')
    Peso1
end
for i=1:n
    x=aut0
    a=Peso1
    f=1./(1+x.^2) %función que se desea aproximar
    z=a*f
    sum(z)
end
end
#####

```

A.2.2. Método 2

```

#####
# Archivo en Matlab: matrices_de_Jacobi_de_Tchebyshev_primer_tipo.m
#####
function matrices_de_Jacobi_de_Tchebyshev_primer_tipo
n=input('Introduce n= ');
e = ones(n,1) ; %vector con n filas y 1 columna con componentes igual a 1
A = zeros(n,n) ;
a12 = 1/sqrt(2) ;
e1 = ones(1,n-1) ; %vector con 1 fila y n columnas con componentes igual a 1
e2 = (1/2)*e1 ;
c1 = [a12 e2] ; %vector que es la unión de los dos vectores
c11 = [a12 a12 e2(1:n-2)] ;
zeta_{n+1} = c1' ; %se calculan los traspuestos
zeta_{n} = c11' ;
B = spdiags([zeta_{n+1} 0*e zeta_{n}], -1:1, n, n) ; %comando que permite insertar
%una matriz tridiagonal
J =A+B %J es una matriz n x n

```

```
[X,D]=eig(J) %D es matriz diagonal que contiene los autovalores de la matriz J
            %las columnas de X son autovectores de V
            %si se quieren los autovalores en forma de vector columna se
            %procede con aut=eig(J)
end
#####
```

A.3. Problema de autovalores, autovectores y pesos para los polinomios de Chebyshev de segunda especie

```
#####
# Archivo en Matlab: chebyshev_de_segundo_tipo.m
#####
function chebyshev_de_segundo_tipo

disp('*Ceros de polinomios de Chebyshev de segundo tipo') ;
n = input('Introduce el número n de ceros = ') ;
e = ones(n,1) ; %vector con n filas y 1 columna con componentes igual a 1
A = zeros(n,n) ;

B = spdiags([(1/2)*e 0*e (1/2)*e], -1:1, n, n); %se crea una matriz tridiagonal
J =A+B %J es la matriz de Jacobi de los polinomios de Chebyshev de segundo tipo
[X,D]=eig(J) %D es matriz diagonal que contiene los autovalores de la matriz J
            %las columnas de X son autovectores de V
            %si se quieren los autovalores en forma de vector columna se
            %hace aut=eig(J)
fprintf('Los autovalores son: \n')
aut0=diag(D) %autovalores
fprintf('Los autovectores por columnas son: \n')
autvect= X %autovector de los autovalores
for i=1:n
    syms x
    X(:,i) ; %autovector asociado a cada autovalor
    norma=norm(X(:,i),2) ; %estudio de la norma del autovector
    fprintf('El autovector normalizado es: \n')
    autvect2=X(:,i)/norma ; %autovector normalizado
    norma2=norm(autvect2,2) ; %compruebo que el vector sea unitario
    c_0=int(sqrt(1-x^2),-1,1) ; %momento de orden cero
    Peso0(i)=c_0*(autvect2(1)^2) ; %expresión de peso para función que no es de probabilidad
    Peso1(i)=eval(Peso0(i)) ; %peso
    fprintf('Los pesos son: \n')
    Peso1
end
for i=1:n
    x=aut0
    a=Peso1
    f=1./(1+x.^2) %función que se desea aproximar
    z=a*f
    sum(z)
end
end
#####
```

A.4. Problema de autovalores, autovectores y pesos para los polinomios de Legendre

```
#####
# Archivo en Matlab: legendre.m
#####
function legendre

disp('*Ceros de polinomios de Legendre')
n=input('* Introduzca un número natural n para hallar n+1 ceros: ');
for i=1:n
    J(i,i+1)=(i)/(sqrt(2*i-1)*sqrt(2*i+1));
    J(i+1,i)=(i)/(sqrt(2*i-1)*sqrt(2*i+1));
end
J          ; %J es la matriz de Jacobi de los polinomios de Legendre
[X,D]=eig(J); %D es matriz diagonal que contiene los autovalores de la matriz J
           %las columnas de X son autovectores de V
           %si se quieren los autovalores en forma de vector columna se
           %hace aut=eig(J)
fprintf('Los autovalores son: \n')
aut0=diag(D) %autovalores
fprintf('Los autovectores por columnas son: \n')
autvect= X %autovector de los autovalores
for i=1:(n+1)
    syms x
    X(:,i)          ; %autovector asociado a cada autovalor
    norma=norm(X(:,i),2) ; %estudio de la norma del autovector
    fprintf('El autovector normalizado es: \n')
    autvect2=X(:,i)./norma %autovector normalizado
    norma2=norm(autvect2,2) ; %compruebo que el vector sea unitario
    c_0=int(x/x,-1,1) ; %momento de orden cero
    Peso0(i)=c_0*(autvect2(1)^2); %expresión de peso para función que no es de probabilidad
    Peso1(i)=(Peso0(i)) ; %peso
    fprintf('Los pesos son: \n')
    Peso1
end
for i=1:n
    x=aut0
    a=Peso0
    f=1./(1+x.^2) %función que se desea aproximar
    z=a*f
    sum(z)
end
end
#####
```

A.5. Integración numérica: fórmulas de Newton-Cotes

```
#####
# Archivo en Matlab: metodos_newton_cotes.m
#####
function metodos_newton_cotes
```

```

%Aproximación numérica de una función dada a través de los métodos de Newton-Cotes
%Presentación de los problemas y los métodos
fprintf(' |-----|\n ') ;
fprintf(' | SELECCIÓN DE MÉTODOS                |\n ') ;
fprintf(' |1.-Integral de la función              |\n ') ;
fprintf(' |2.-Regla del rectángulo extremo izquierdo |\n ') ;
fprintf(' |3.-Regla del rectángulo extremo derecho   |\n ') ;
fprintf(' |4.-Regla del punto medio                  |\n ') ;
fprintf(' |5.-Regla trapezoidal                       |\n ') ;
fprintf(' |6.-Regla de simpson 1/3                    |\n ') ;
fprintf(' |7.-Regla de simpson 3/8                    |\n ') ;
fprintf(' |-----|\n ') ;
nprob1=input('Indique el método de aproximación   |\n ') ;
fprintf(' |1.-Simple                                  |\n ') ;
fprintf(' |2.-Compuesta                               |\n ') ;
nprob2=input('Indique el problema a estudiar \n ');
[a1,b1,n]=inicial(nprob2)
syms x t;
f=func(x);
kk=(b1-a1)/2;
x=((b1-a1)*t+(b1+a1))/2 ;%Transformación lineal  t=(2*x-(b1+a1))/(b1-a1);
a2=-1;
b2=+1;
g=func(x) ;%expresión de la transformación lineal
g1=inline(g) ;%"función transformada"
%comprobamos que los resultados de la función transformada y la original
%son los mismos aunque estén definidas en diferentes intervalos
    if      nprob1==1
        integral2=integr(f,a1,b1); %Obtenemos el resultado de la integral
        integra=vpa(integral2)
    elseif nprob1==2
        rectizq=rectizqui(g1,nprob2,a2,b2,n,kk)
    elseif nprob1==3
        rectder=rectderec(g1,nprob2,a2,b2,n,kk)
    elseif nprob1==4
        ptomedi=pntomedio(g1,nprob2,a2,b2,n,kk)
    elseif nprob1==5
        trapez =trapezoid(g1,nprob2,a2,b2,n,kk)
    elseif nprob1==6
        simp1_3=simpso1_3(g1,nprob2,a2,b2,n,kk)
    elseif nprob1==7
        simp3_8=simpso3_8(g1,nprob2,a2,b2,n,kk)
    end;
end

function [a1,b1,n]=inicial(nprob2)
    n=0;
    a1=input('Indique el valor de a= '); %Intervalo de integración
    b1=input('Elige el valor de b= '); %Intervalo de integración
    if nprob2==2
        n=input('Indique el número de pasos \n ');
    end;
end
end

```

```

function integral2=integr(f,a1,b1) %función de la integral definida
format long
integral2=int(f,a1,b1);
end

function f=func(x)
f=x^3-3.3*x^2+3*x;
end

function rectizq = rectizqui(g1,m,a2,b2,n,kk)
if m==1
rectizq=kk*(b2-a2)*g1(a2);
return
elseif m==2
h=(b2-a2)/n;
sum=0;
for k=0:(n-1)
sum=sum+g1(a2+k*h);
end
rectizq=kk*h*sum;
end
end

function rectder=rectderec(g1,m,a2,b2,n,kk)
if m==1
rectder=kk*(b2-a2)*g1(b2);
return
elseif m==2
h=(b2-a2)/n;
sum=0;
for k=1:n
sum=sum+g1(a2+k*h);
end
rectder=h*kk*sum;
end
end

function ptomedi=pntomedio(g1,m,a2,b2,n,kk)
if m==1
ptomedi=kk*(b2-a2)*g1((a2+b2)/2);
return
elseif m==2
h=(b2-a2)/n;
sum=0;
for k=0:(n-1)
sum=sum+g1(((a2+k*h)+(a2+(k+1)*h))/2);
end
ptomedi=kk*h*sum;
end
end

function trapez=trapezoid(g1,m,a2,b2,n,kk)
if m==1
trapez=kk*(b2-a2)/2*(g1(a2)+g1(b2));

```

```

    return
elseif m==2
    h=(b2-a2)/n;
    sum=0;
    for k=1:(n-1)
        sum=sum+g1(a2+k*h);
    end
    trapez=kk*(h/2*(g1(a2)+ 2*sum + g1(b2)));
end
end

function simp1_3=simpso1_3(g1,m,a2,b2,n,kk)
if m==1
    simp1_3=kk*((b2-a2)/6)*(g1(a2)+4*g1((a2+b2)/2)+g1(b2));
    return
elseif m==2
    h=(b2-a2)/(n);
    sum1=0;
    sum2=0;
    for k=2:2:n-2
        sum1=sum1+g1((a2+k*h));
    end
    for k=2:2:n
        sum2=sum2+g1((a2+(k-1)*h));
    end
    simp1_3=kk*h*(1/3)*(g1(a2) +2*sum1 +4*sum2 +g1(b2))
end
end

function simp3_8=simpso3_8(g1,m,a2,b2,n,kk)
if m==1
    simp3_8=kk*((b2-a2)/8)*(g1(a2)+3*g1(a2+(1/3)*(b2-a2))+3*g1(a2+(2/3)*(b2-a2))+g1(b2));
    return
elseif m==2
    h=(b2-a2)/n;
    sum1=0;
    sum2=0;
    sum3=0;
    for k=0:(n/3-1)
        sum1=sum1+g1(a2+(3*k+1)*h);
    end
    for k=0:(n/3-1)
        sum2=sum2+g1(a2+(3*k+2)*h);
    end
    for k=1:(n/3-1)
        sum3=sum3+g1(a2+3*k*h);
    end
    simp3_8=kk*(3*h/8)*(g1(a2) + 3*sum1 + 3*sum2 + 2*sum3 + g1(b2));
    return
end
end
#####

```

Bibliografía

- [1] T.S. Chihara. *An Introduction to Orthogonal Polynomials*. Mathematics and Its Applications Series. Gordon and Breach, New York, 1978.
- [2] P.J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, New York, 1984.
- [3] W. Gautschi. *A survey of Gauss-Christoffel quadrature formulae*. In: E.B. Christoffel; The Influence of his Work in Mathematics and the Physical Sciences, P. L. Eatzer and F. Fehér editors, Birkhäuser, Base (1981), 72-147.
- [4] V.I. Krylov. *Approximate Calculation of Integrals*. The MacMillan Company, New York, 1962.
- [5] I.H. Sloan and W.E. Smith. *Properties of interpolatory product integration rules*. SIAM J. Numer. Anal. 19 (1982), 427-442.
- [6] J. Stoer and R. Burlisch. *Introduction to Numerical Analysis*. Springer-Verlay, 1980.
- [7] W. Van-Assche. *Analytic aspects of Orthogonal Polynomials*. Katholieke Universiteit Leuven, 1993.