

TEMA 5: DERIVACIÓN E INTEGRACIÓN NUMÉRICAS

1. INTRODUCCIÓN. OBJETIVOS

El objetivo principal de este tema es estudiar algoritmos numéricos que nos permitan aproximar numéricamente el valor de la derivada de una función $f(x)$ en un determinado punto x_0 , y el de la integral de dicha función en un intervalo $[a, b]$.

Estos métodos nos serán especialmente útiles para situaciones donde la expresión explícita de $f(x)$ es difícil de manejar (a la hora de hacer derivadas y/o integrales), o bien no es conocida. En este último caso, sólo se dispone del valor de dicha función en un conjunto de nodos.

Típicamente, la precisión de los métodos numéricos que requieren de discretización se expresa en la forma $O(h^P)$, se lee O grande de h elevado a p , donde h representa la distancia mínima de separación entre nodos y P es un número natural, normalmente asociado a la regularidad de las funciones que se tratan de aproximar.

La expresión $O(h^P)$ nos dice que el error numérico del método es aproximadamente h^P , es decir, error $\approx h^P$.

Por tanto, teniendo en mente que siempre $0 < h < 1$, el error se podrá reducir disminuyendo h o aumentando P . En el primero de los casos hablamos de un h -método y en el segundo de un p -método.

Estudiamos en primer lugar el problema de derivación numérica mediante un ~~método~~ p -método, también llamados método espectral.

2. DERIVACIÓN NUMÉRICA

El esquema numérico del siguiente método espectral es muy sencillo y consta de dos pasos:

~~Suponemos~~

Para simplificar, suponemos que trabajamos en el intervalo $[-1, 1]$ y que conocemos los valores $u = (u_0, u_1, \dots, u_N)$ de una función dada en los nodos de Chebyshev $x_j = \cos \frac{j\pi}{N}$, $0 \leq j \leq N$.

El objetivo es calcular los valores discretos $w = (w_0, w_1, \dots, w_N)$ de la derivada en los mismos nodos de Chebyshev.

Como decíamos, procedemos en dos pasos:

- 1º) Calculamos el único polinomio de interpolación $P(x)$ de grado $\leq N$ que interpola nuestros datos, es decir, $P(x_j) = u_j$, $0 \leq j \leq N$.
- 2º) Calculamos la derivada del polinomio y evaluamos en los nodos de Chebyshev, es decir,
$$w_j = P'(x_j), \quad 0 \leq j \leq N.$$

Como la derivación es una operación lineal y estamos trabajando con un número finito de datos, los cálculos anteriores se pueden expresar en forma matricial, es decir, existe una matriz D_N , de tamaño $(N+1) \times (N+1)$ tal que

$$w = D_N u$$

Veamos cómo calcular esta matriz de derivación numérica.

Empezemos con el caso $N=1$. Partimos de los nodos $x_0=1$, $x_1=-1$ y de los datos de interpolación v_0, v_1 .

Calculamos el polinomio de interpolación en forma de Lagrange:

$$p(x) = v_0 l_0(x) + v_1 l_1(x)$$

$$l_0(x) = \frac{x - x_1}{x_0 - x_1} = \frac{x + 1}{2}$$

$$l_1(x) = \frac{x - x_0}{x_1 - x_0} = \frac{x - 1}{-2} = \frac{1}{2} (1 - x).$$

$$p(x) = \frac{1}{2} (1 + x) v_0 + \frac{1}{2} (1 - x) v_1.$$

Calculamos la derivada:

$$p'(x) = \frac{1}{2} v_0 - \frac{1}{2} v_1$$

La matriz de derivación numérica $D_{\mathbb{I}}$ es:

$$D_{\mathbb{I}} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

En efecto:

$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \underset{\text{cte.}}{w} = D_{\mathbb{I}} \cdot v = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} v_0 - \frac{1}{2} v_1 \\ \frac{1}{2} v_0 - \frac{1}{2} v_1 \end{bmatrix}$$

Caso $N=2$. Los nodos de Chebyshev son

$x_0 = 1$, $x_1 = 0$, $x_2 = -1$ y el polinomio interpolador de Lagrange

$$P(x) = \frac{1}{2} x (1+x) v_0 + (1+x)(1-x) v_1 + \frac{1}{2} x (x-1) v_2.$$

Su derivada vale

$$P'(x) = (x + \frac{1}{2}) v_0 - 2x v_1 + (x - \frac{1}{2}) v_2,$$

y, por tanto, la matriz de derivación D_2 es

$$D_2 = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix} \begin{array}{l} \leftarrow \text{nodo } x_0 = 1 \\ \leftarrow \text{nodo } x_1 = 0 \\ \leftarrow \text{nodo } x_2 = -1 \end{array}$$

con lo que el valor de la derivada en los nodos de Chebyshev se calcula

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}.$$

En el caso general N , los cálculos son más largos y tediosos por lo que nos limitamos a dar la expresión explícita de D_N .

La función `cheb.m` calcula la matriz de derivación de Chebyshev junto con los propios nodos de Chebyshev x_j . Este código calcula las entradas de D_N fuera de la diagonal mediante la fórmula anterior, y las entradas de la diagonal a partir de la identidad

$$(D_N)_{ii} = - \sum_{\substack{j=0 \\ j \neq i}}^N (D_N)_{ij}.$$

Los detalles pueden encontrarse en el libro:

J.N. Trefethen, *Spectral Methods in Matlab*, SIAM, 2000.

(Chapter 6).

EXTENSIÓN A DIMENSIONES SUPERIORES

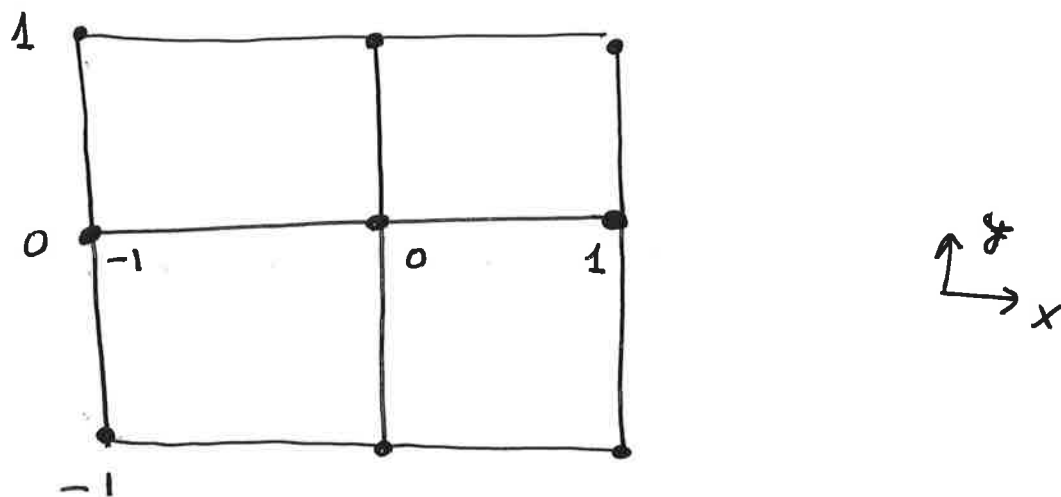
Supongamos ahora que $f = f(x, y)$ depende de dos variables. Nos interesa calcular, numéricamente, el gradiente de f , $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$.

Consideramos el producto cartesiano de los nodos de Chebyshev en las dos direcciones espaciales.

Por ejemplo, para el caso $N=2$, tenemos

$$x_0 = +1, \quad x_1 = 0, \quad x_2 = -1$$

$$y_0 = +1, \quad y_1 = 0, \quad y_2 = -1$$



Nodos bidimensionales = $\{ (-1, -1), (-1, 0), (-1, 1),$
 $(0, -1), (0, 0), (0, 1)$
 $(1, -1), (1, 0), (1, 1) \}$.

Como datos, tendríamos también $f_{ij} = f(x_i, y_j)$,

$$0 \leq i \leq N, \quad 0 \leq j \leq N.$$

Recordemos que la derivada parcial respecto de una variable se calcula como la derivada usual respecto de esa variable cuando se congela la otra variable.

Por tanto, por ejemplo, para calcular $\frac{\partial f}{\partial x}(x_i, y_j)$,

tendríamos que aplicar la derivada de Chebyshev 1d a los datos (x_i, y_j) , $0 \leq i \leq N$, con j fija.

De igual forma se procede para $\frac{\partial f}{\partial y}(x_i, y_j)$, esta vez fijando i y variando j .

• Respecto de la "precisión" de la derivación chebyshev, ésta es, en esencia, similar a la interpolación, es decir, polinomial para el caso de funciones diferenciables, y exponencial para funciones analíticas.

• Estabilidad: éste es el gran problema de la derivación, no sólo la numérica, sino la analítica.

En efecto, consideremos una función derivable $f(x)$, la cual perturbamos ligeramente en la forma

$$f_n(x) = f(x) + \frac{1}{\sqrt{n}} \operatorname{sen}(nx).$$

Nótese que como $|\operatorname{sen}(nx)| \leq 1$, entonces, para n grande, el término $\frac{1}{\sqrt{n}} \operatorname{sen}(nx)$ es muy pequeño.

Sin embargo, $f'_n(x) = f'(x) + \sqrt{n} \cos(nx)$.

Ahora, el término $\sqrt{n} \cos(nx) \xrightarrow{n \rightarrow \infty} \infty$.

Es decir, pequeños errores ($\frac{1}{\sqrt{n}} \operatorname{sen}(nx)$) en la entrada, producen grandes errores ($\sqrt{n} \cos(nx)$) en la salida. Dicho de otra manera:

$$|f(x) - f_n(x)| \xrightarrow{n \rightarrow \infty} 0, \text{ pero}$$

$$|f'(x) - f'_n(x)| = |\sqrt{n} \cos(nx)| \xrightarrow{n \rightarrow \infty} \infty.$$

La derivación es, por tanto, una operación altamente inestable. En Matemáticas, se dice que es un problema mal puesto (ill-posed, en inglés).

Desde el punto de vista discreto, numérico, esta falta de estabilidad es consecuencia de que derivar es dividir por números muy pequeños. Por tanto, cualquier pequeño error en el denominador ^{y lo numerador} puede provocar que dicho denominador sea muy próximo a cero, y así el cociente explota. Este problema tiene consecuencias prácticas en Ingeniería muy importantes.

Por ejemplo, un GPS recibe posiciones $x(t)$ con gran precisión. También calcula velocidades $\frac{dx}{dt}$.

¿Cómo lo hace? En principio, mediante el cociente

$$\frac{x(t_2) - x(t_1)}{t_2 - t_1}$$

Pero para que este cálculo sea preciso t_2 tiene que estar muy cerca de t_1 . Pero claro, cualquier pequeño error en las posiciones (ruido en los datos), son amplificados por el denominador $t_2 - t_1$.

En cambio, si pre' calculamos velocidades promedio,

es decir,

$$\int_0^t v(s) ds = \int_0^t \frac{dx}{ds} ds = x(t) - x(0),$$

los cálculos apenas se ven afectados por ruido en los datos porque integrar es sumar, mientras que derivar es ~~dividir~~ multiplicar (dividir).

La integración sí es una operación estable numéricamente, y de ella, nos ocupamos a continuación.

3. INTEGRACIÓN NUMÉRICA

La idea de la integración numérica (también llamada cuadratura numérica) consiste en aproximar el valor de una integral por una suma finita en la forma:

$$I = \int_a^b f(x) dx \approx I_n = \sum_{j=0}^n w_j f(x_j)$$

x_j se llaman nodos de la cuadratura

w_j " " pesos de la cuadratura.

Los pesos se eligen de forma que $I = I_n$ ~~para~~

cualquier $f(x)$ es un polinomio de grado $\leq n$,

es decir $I = I_n$ si $f(x) = p(x)$ polinomio de grado $\leq n$.

La elección de los nodos x_j también juega un papel destacado en términos de precisión, estabilidad y velocidad de cálculo.

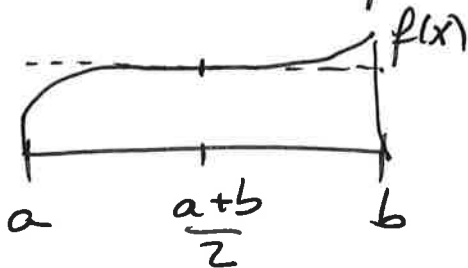
Distinguimos tres tipos de fórmulas:

- Newton-Cotes: nodos x_j equiespaciados
- Gauss: nodos x_j son los ceros de los polinomios de Legendre
- Clenshaw-Curtis: nodos x_j de Chebyshev.

Repasamos a continuación cada una de estas fórmulas:

FÓRMULAS DE CUADRATURA DE NEWTON-COTES.

- Fórmula del punto medio (o de 1 punto)



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

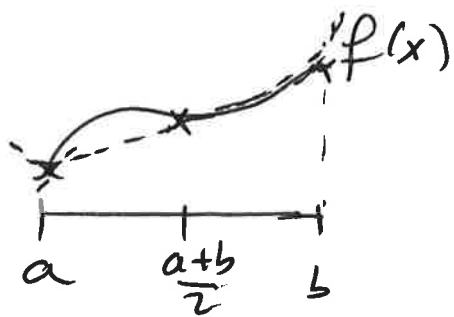
- Fórmula del trapecio. (o de 2 puntos).



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

Se calcula la recta que pasa por $(a, f(a))$ y $(b, f(b))$. Se integra dicha recta y da la fórmula del trapecio. (12)

• Fórmula de Simpson (o de 3 puntos).



$$I = \int_a^b f(x) dx \approx \frac{b-a}{6} (f(a) + 4f(\frac{a+b}{2}) + f(b))$$

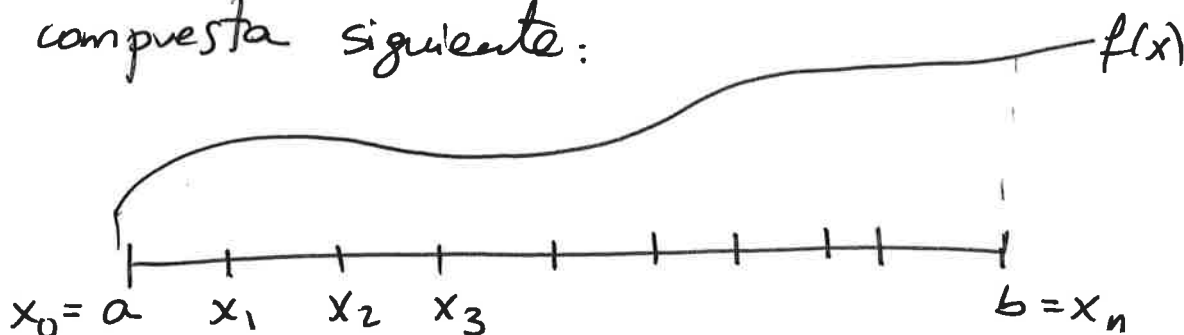
Se calcula la parábola $y = c_0 + c_1x + c_2x^2$ que pasa por $(a, f(a))$, $(\frac{a+b}{2}, f(\frac{a+b}{2}))$, $(b, f(b))$ y se integra dicha parábola. El resultado es la fórmula de Simpson.

~~Siendo $h = \frac{b-a}{2}$, la precisión de dichas fórmulas es del orden de h^{n+3} si~~

Las fórmulas anteriores dan una buena precisión si el intervalo es pequeño, es decir, $b-a = h \ll 1$.

Para intervalos de integración más grandes, descomponemos el intervalo en sub-intervalos más pequeñas y aplicamos las fórmulas anteriores en cada sub-intervalo.

La más utilizada es la fórmula de Simpson compuesta siguiente:



$h = x_{j+1} - x_j$ nodos equiespaciados

$$x_{j+\frac{1}{2}} = x_j + \frac{h}{2}$$

$$f_j = f(x_j), \quad f_{j+\frac{1}{2}} = f(x_{j+\frac{1}{2}})$$

$$\int_a^b f(x) dx \approx \frac{h}{6} \left[f_0 + 2 \sum_{j=1}^{n-1} f_j + 4 \sum_{j=0}^{n-1} f_{j+\frac{1}{2}} + f_n \right]$$

El error en la fórmula anterior es del orden de $\left(\frac{1}{n}\right)^4$ para funciones $f \in C^4([a, b])$.

FÓRMULA DE CUADRATURA DE GAUSS

Los nodos x_j son los ceros de una clase de polinomios llamados de Legendre. Algunos de estos polinomios son x , $x^2 - \frac{1}{3}$, $x^3 - \frac{3}{5}x$.

La gran ventaja de la fórmula de Gauss es su precisión. En efecto, con $n+1$ -nodos integran de manera exacta polinomios de grado $2n+1$!

El cálculo de los pesos no es sencillo, El código gauss.m proporciona los nodos y pesos.

La fórmula de Gauss queda:

$$\int_{-1}^1 f(x) dx \approx \sum_{j=0}^n w_j f(x_j)$$

Para obtener x_j y w_j ejecutar en Octave

$$[x, w] = \text{gauss}(n).$$

Nota. - Recordemos que si la integración es en $[a, b]$, los nodos han de transformarse tal y como lo hicimos en el tema anterior.

FÓRMULA DE CUADRATURA DE CLENSHAW-CURTIS

Los nodos x_j son los nodos de Chebyshev.

La estrategia para deducir la fórmula de cuadratura es:

1º) Se calcula el polinomio de interpolación de grado $\leq N$ tal que $p(x_j) = f_j$, $0 \leq j \leq N$, con

$$f_j = f(x_j).$$

2º) La aproximación de $I = \int_{-1}^1 f(x) dx$ viene dada

$$\text{por } I_N = \int_{-1}^1 p(x) dx.$$

La fórmula de cuadratura resultante se denomina cuadratura de Clenshaw-Curtis y adopta la forma

$$I_N = \sum_{j=0}^N w_j f(x_j)$$

Los nodos y pesos se pueden obtener del código `clencurt.m`.

$$[x, w] = \text{clencurt}(N).$$

¿Gauss ó Clenshaw-Curtis?

Aunque la cuadratura de Gauss es más precisa que la de Clenshaw-Curtis, esta diferencia no suele ser apreciable en la mayoría de situaciones prácticas. La gran ventaja de Clenshaw-Curtis es que el algoritmo que se suele usar para calcular los pesos es muy estable con lo que la estabilidad de la fórmula de cuadratura de Clenshaw-Curtis también lo es cuando el número de nodos es muy elevado. Un buen test numérico para evaluar la precisión es calcular la diferencia

$$I_{2N} - I_N.$$

Una estimación del error para funciones de clase C^k , en la fórmula de Clenshaw-Curtis (comparable al error en la fórmula de Gauss) es

$$|I - I_n| \leq \frac{A}{(2n+1-k)^k}$$

con A una cota de $|f^{(k)}(x)|$ en $[a, b]$.

EXTENSIÓN AL CASO DE INTEGRALES DOBLES

Y TRIPLES.

Sea $f: [-1, 1] \times [-1, 1] \rightarrow \mathbb{R}$
 $(x, y) \mapsto f(x, y)$

La idea básica para aproximar $\int_{-1}^1 \int_{-1}^1 f(x, y) dx dy$

es usar el Teorema de Fubini para transformar esta integral doble en dos integrales iteradas en dimensión 1, es decir,

$$\begin{aligned} \int_{-1}^1 \int_{-1}^1 f(x, y) dx dy &= \int_{-1}^1 \left(\underbrace{\int_{-1}^1 f(x, y) dx}_{G(y)} \right) dy \\ &= \int_{-1}^1 G(y) dy. \end{aligned}$$

De esta forma podemos aplicar las reglas de cuadratura 1D ya estudiadas a cada una de estas integrales con lo que resulta:

$$\int_{-1}^1 \int_{-1}^1 f(x,y) dx dy = \int_{-1}^1 \underbrace{\left(\int_{-1}^1 f(x,y) dx \right)}_{G(y)} dy$$
$$\approx \sum_j w_j G(y_j) \approx \sum_j \sum_i w_j w_i f(x_i, y_j)$$

ya que

$$G(y_j) = \int_{-1}^1 f(x, y_j) dx \approx \sum_i w_i f(x_i, y_j).$$

En las fórmulas anteriores w_i, w_j representan los pesos de la cuadratura elegida (Newton-Cotes, Gauss, Clenshaw-Curtis).

Desde el punto de vista de la implementación numérica, el proceso anterior se traduce en un ~~loop~~ "bucle for" donde en cada paso del bucle hemos de usar una cuadratura 1D.

Nota. - Aunque hay algoritmos numéricos para el cálculo directo (sin usar Fubini) de

de integrales múltiples (la fórmula de Sincidiak es la más emblemática), que hacen uso de "mallas sparse", estas fórmulas escapan de los objetivos de un primer curso de Cálculo Numérico.

Nota. Para integrales triples se usa la misma idea del Teorema de Fubini, es decir,

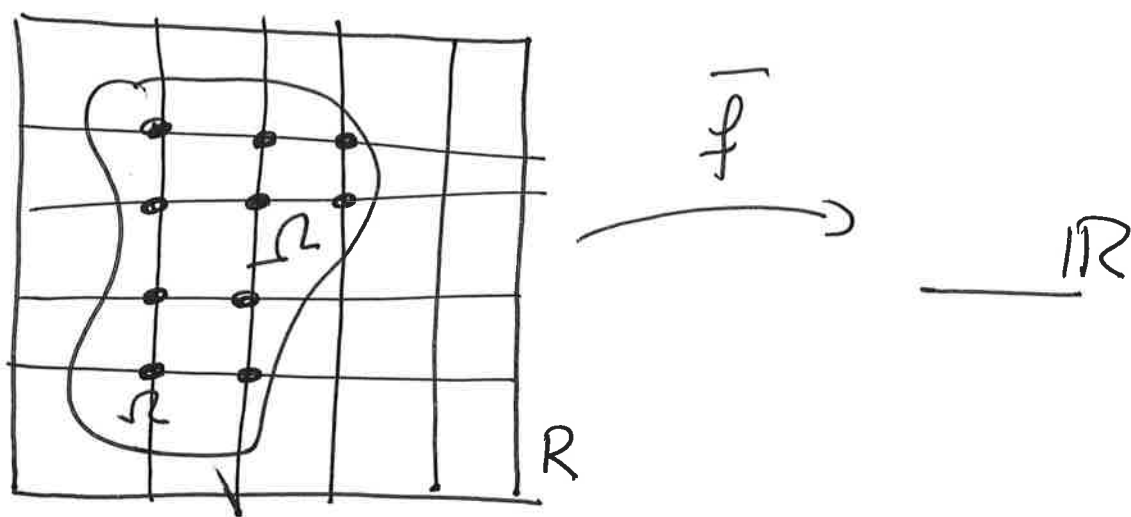
$$\begin{aligned} & \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(x, y, z) dx dy dz = \\ & = \int_{-1}^1 \left(\int_{-1}^1 \underbrace{\left(\int_{-1}^1 f(x, y, z) dx \right)}_{G(y, z)} dy \right) dz \\ & \qquad \qquad \qquad \underbrace{\hspace{10em}}_{H(z)} \end{aligned}$$

$$\approx \sum_i w_i H(z_i) \approx \sum_i \sum_j w_i w_j G(y_j, z_i)$$

$$\approx \sum_{i,j,k} w_i w_j w_k f(x_k, y_j, z_i)$$

Nota.- Para el caso de dominios no rectangulares, la idea ξ consiste en embeber el dominio en un rectángulo y extender la función a integrar con valor cero fuera del rectángulo.

Graáficamente,



$$\iint_{\Omega} f(x,y) dx dy = \iint_{R} \bar{f}(x,y) dx dy$$

$$\bar{f}(x,y) = \begin{cases} f(x,y) & \text{si } (x,y) \in \Omega \\ 0 & \text{si } (x,y) \notin \Omega. \end{cases}$$

El procedimiento anterior se aplica ahora sobre la función \bar{f} en el rectángulo R .