$$T_1(h) = \frac{4\,T_0(h/2) - T_0(h)}{3} = I + b_4 h^4 + b_6 h^6 + b_8 h^8 + \dots$$

Repetimos la idea: en $T_1(h)$ reemplazamos h por $\frac{h}{2}$, y hacemos $\frac{16\,T_1(h/2) - T_1(h)}{15}$. Resulta una fórmula para I donde el error comienza en la potencia h^6.

$$T_2(h) = \frac{16\,T_1(h/2) - T_1(h)}{15} = I + c_6 h^6 + c_8 h^8 + \dots$$

En general, si $T_k(h)$ es una fórmula cuyo error comienza en h^{2k+2}, $T_{k+1}(h) = \frac{4^{k+1}\,T_k(h/2) - T_k(h)}{4^{k+1} - 1}$ será una fórmula cuyo error comienza en h^{2k+4}

Este mecanismo se conoce como extrapolación de Richardson.

Si $I = \int_a^b f(x)\,dx$, y $T(h)$ es la regla del trapecio compuesta, tenemos un método iterativo de integración numérica llamado método de Romberg.

$T_0(h)$

$T_0(h/2)$ $\quad T_1(h)$

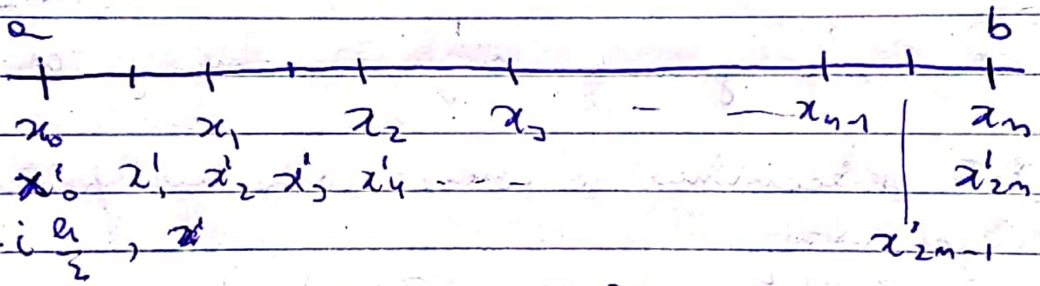
$T_0(h/4)$ $\quad T_1(h/2)$ $\quad T_2(h)$

$T_0(h/8)$ $\quad T_1(h/4)$ $\quad T_2(h/2)$ $\quad T_3(h)$ $\quad$ etc.

Puede demostrarse que:

$T_1(h)$ = regla compuesta de Simpson.

y también:



$x'_j = a + i\frac{h}{2}$,

$$T_0(h/2) = \frac{h/2}{2}\left[f(x'_0) + 2\sum_{j=1}^{2m-1} f(x'_j) + f(x'_{2m})\right] =$$

$$= \frac{h/2}{2}\Big[f(x_0) + 2\big(f(x'_1) + f(x_1) + f(x'_3) + f(x_2) +$$

$$+ f(x'_5) + f(x_3) + \dots + f(x_{m-1}) + f(x'_{2m-1}) + f(x_m)\Big] =$$

$$= \frac{h/2}{2}\left[f(x_0) + 2\sum_{i=1}^{m-1} f(x_i) + f(x_m) + \right.$$

$$\left. + \sum_{j=1}^{m} f(x'_{2j-1})\right] =$$

$$= \frac{1}{2}\left[T_0(h) + \frac{h}{2}\sum_{j=1}^{m} f\left(a + (2j-1)\frac{h}{2}\right)\right] =$$

$$= \frac{1}{2}\left[T_0(h) + \frac{h}{2}\sum_{j=1}^{n} f\left(a + \left(j - \frac{1}{2}\right)h\right)\right].$$

~~[illegible]~~

Generalmente el método de Romberg itera hasta $|T_{k-1}(h) - T_k(h)| < \varepsilon$, tolerancia prefijada.

Si no se ~~tiene~~ conoce a $f(x)$, pero sí se tiene una tabla con los pesos adecuados, se puede iterar hasta determinado punto.

Por ejemplo:

x	$f(x)$
0	—
0,25	—
0,5	—
0,75	—
1	—
1,25	—
1,5	—
1,75	—
2	—

Se empieza iterando con $h=2$, y luego se sigue con $\frac{h}{2}=1$, $\frac{h}{4}=0{,}5$, $\frac{h}{8}=0{,}25$.

Ahí termina el algoritmo.

Integración o cuadratura gaussiana.

Las fórmulas de Newton-Cotes tienen grado de precisión n (n impar $+1$ puntos, n impar) o $n+1$ ($n+1$ puntos, n par). No hace falta conocer $f(x)$ sólo una tabla de la misma ~~información~~ en los nodos equiespaciados.

Si se conoce $f(x)$, puede obtenerse fórmula de mayor precisión o grado de exactitud. Para ello se eligen los nodos de otra manera. Este método selecciona constantes $c_1, c_2, \ldots, c_n$ y nodos $x_1, x_2, \ldots, x_n$ de modo tal que la fórmula:

$$\int_a^b f(x)\,dx \simeq \sum_{i=1}^{n} c_i f(x_i)$$

tenga el mayor grado de exactitud posible. Un polinomio de grado $2n-1$ tiene $2n$ parámetros y juntamente $c_1, c_2, \ldots, c_n$ y $x_1, x_2, \ldots, x_n$ son $2n$, es razonable suponer que puede lograrse que la fórmula sea exacta para polinomios de grado $\leq 2n-1$, o sea, que tenga grado de exactitud $2n-1$.

Supongamos que queremos aproximar $\int_{-1}^{1} f(x)\,dx$. Aproximamos $f(x)$ mediante el polinomio de Lagrange de grado $\leq n-1$:

$$f(x) = \sum_{i=1}^{n} \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)} f(x_i) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=1}^{n} (x-x_i)$$

$$\int_{-1}^{1} f(x)\,dx = \sum_{i=1}^{n} c_i f(x_i) + \int_{-1}^{1} \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=1}^{n} (x-x_i)\,dx$$

con $c_i = \displaystyle\int_{-1}^{1} \prod_{\substack{j=1 \\ j=i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}\,dx$

Esta fórmula es exacta para polinomios de grado $\leq n-1$, pero eligiendo convenientemente los x_i, podemos lograr que tenga precisión $2n-1$.

Sea P un polinomio de grado $2n-1$. Lo dividimos por el polinomio de Legendre de grado n, P_n:

$$P(x) = Q(x)\,P_n(x) + R(x)$$

con $Q(x)$ y $R(x)$ polinomios de grado $\leq n-1$

Como $Q(x)$ es de grado $\leq n-1$, se puede escribir como ~~polinom~~ combinación lineal de polinomios de Legendre de hasta grado $n-1$:

$$Q(x) = \sum_{i=0}^{n-1} \alpha_i P_i(x)$$

Entonces:

$$\int_{-1}^{1} P(x)\,dx = \int_{-1}^{1} Q(x)\,P_n(x)\,dx + \int_{-1}^{1} R(x)\,dx =$$

$$= \int_{-1}^{1} \left(\sum_{i=0}^{n-1} \alpha_i P_i(x)\right) P_n(x)\,dx + \int_{-1}^{1} R(x)\,dx =$$

$$= \sum_{i=0}^{n-1} \alpha_i \int_{-1}^{1} P_i(x)\,P_n(x)\,dx + \int_{-1}^{1} R(x)\,dx$$

Como los polinomios de Legendre son ortogonales, resulta:

$$\int_{-1}^{1} P_i(x)\,P_n(x)\,dx = 0, \quad i = 0, 1, \ldots, n-1$$

$$\int_{-1}^{1} P(x)\,dx = \int_{-1}^{1} R(x)\,dx = \sum_{i=1}^{n} c_i R(x_i)$$

ya que la fórmula es exacta para pol. de grado $n-1$

Sea x_i una raíz del pol. de Legendre de grado n, $i = 1, 2, \ldots, n$:

$$P(x_i) = Q(x_i) P_n(x_i) + R(x_i) = R(x_i)$$

$$\therefore \int_{-1}^{1} P(x) dx = \sum_{i=1}^{n} c_i R(x_i) = \sum_{i=1}^{n} c_i P(x_i)$$

O sea, la fórmula es exacta para polinomios de grado $2n-1$.

Resumiendo: la fórmula:

$$\int_{-1}^{1} f(x) dx \approx \sum_{i=1}^{n} c_i f(x_i)$$ es exacta para

polinomios de grado $2n-1$ si los x_i se eligen como las n raíces del polinomio de Legendre de grado n y los c_i se calculan como:

$$c_i = \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} dx \qquad 1 \leq i \leq n$$

Tanto los x_i como los c_i están tabulados:

n	x_i	c_i
2	$x_1 = 1/\sqrt{3}$	$c_1 = 1$
	$x_2 = -1/\sqrt{3}$	$c_2 = 1$
3	$x_1 = \sqrt{3/5}$	$c_1 = 5/9$
	$x_2 = 0$	$c_2 = 8/9$
	$x_3 = -\sqrt{3/5}$	$c_3 = 5/9$

etc.

Se puede aplicar la integración de Gauss-Legendre a cualquier integral entre a y b, ya que mediante un cambio de variable se puede llevar a una integral entre −1 y 1:

$$x = \frac{b-a}{2} t + \frac{a+b}{2}$$

$$dx = \frac{(b-a)}{2} dt$$

$$\int_a^b f(x)\,dx = \frac{(b-a)}{2} \int_{-1}^{1} f(x(t))\,dt =$$

$$= \frac{(b-a)}{2} \int_{-1}^{1} f\left(\frac{b-a}{2} t + \frac{a+b}{2}\right) dt$$

Ejemplos: estimar $I = \int_{1,8}^{2,6} f(x)\,dx$ donde $f(x)$ viene dada por la siguiente tabla:

x	1,8	2,0	2,2	2,4	2,6
$f(x)$	3,12014	4.42569	6,04241	8,03014	10,46675

Trapecio:

$n=1$ $h=0,8$ $T(h=0,8) = \frac{h}{2}[f(x_0)+f(x_1)] =$

$= \frac{0,8}{2}[f(1,8)+f(2,6)] = 5,434756$

$n=2$ $h=0,4$ $T(h=0,4) = 5,134342$

$n=4$ $h=0,2$ $T(h=0,2) = 5,058337$

Simpson $M=2$ $\quad S(h=0{,}4) = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right] = \frac{0{,}4}{3}\left[f(1{,}8) + 4f(2{,}2) + f(2{,}6)\right] =$

$= 5{,}034204$

$M=4$ $\quad S(h=0{,}2) = 5{,}033002$

Romberg:

$T(h)$		
5,434756		
5,134342	5,034204	
5,058337	5,033002	5,032922

Sea ahora estimar $I = \int_1^9 \ln x\,dx \quad (= 11{,}775021)$

Trapecio simple: $T(h=8) = \frac{h}{2}\left[f(x_0) + f(x_1)\right] =$

$= 8{,}788898$

error: $f(x) = \ln x \quad f'(x) = \frac{1}{x} \quad f''(x) = -\frac{1}{x^2}$

$|E_T| \le \frac{h^3}{12} \max_{1\le x\le 9} |f''(x)| = \frac{h^3}{12} \max_{1\le x\le 9} \frac{1}{x^2} \le$

$\le \frac{h^3}{12} = 42{,}67$

verdadero error: $|11{,}775021 - 8{,}788898| = 2{,}986123$

17

Trapecio $n=4$:

$$T(h=2) = \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{3} f(x_i) + f(x_4)\right] =$$

$$= 11,505145$$

error $|E_T| \leq \frac{h^2(b-a)}{12} \max_{1\leq x\leq 9} |f''(x)| \leq$

$$\leq \frac{h^2(b-a)}{12} = \frac{2^2 \cdot 8}{12} = 2,67$$

Verdadero error : $|11,775021 - 11,505145| =$

$$= 0,269876$$

Simpson $n=4$:

$$S(h=2) = \frac{h}{3}\left[f(x_0) + 4\left(f(x_1) + f(x_3)\right) + \right.$$

$$\left. + 2\, f(x_2) + f(x_4)\right] = 11,729460$$

error $|E_S| \leq \frac{h^4(b-a)}{180} \max_{1\leq x\leq 9} |f^{IV}(x)|$

$f'''(x) = \frac{2}{x^3}$ $\quad f^{IV}(x) = -\frac{6}{x^4}$

$$|E_S| \leq \frac{h^4}{180}(b-a) \max_{1\leq x\leq 9} \frac{6}{x^4} \leq$$

$$\leq 6\,\frac{h^4(b-a)}{180} = 4,2667$$

verdadero error : $|11,775021 - 11,729460| =$

$$= 0,045561$$

Romberg: ejercicio iterar hasta una tolerancia de 10^{-4}.

Gauss-Legendre: $n=3$

Hacemos el cambio: $x = \frac{b-a}{2} t + \frac{a+b}{2}$

$$x = \frac{9-1}{2} t + \frac{9+1}{2}$$

$$x = 4t + 5$$

$$\int_1^9 \ln x \, dx = 4 \int_{-1}^1 \ln(4t+5) dt \cong$$

$$\cong 4 \left[c_1 \ln(4x_1+5) + c_2 \ln(4x_2+5) + c_3 \ln(4x_3+5) \right] =$$

$n=3$

$x_1 = 0{,}774597$	$c_1 = 5/9$
$x_2 = 0$	$c_2 = 8/9$
$x_3 = -0{,}774597$	$c_3 = 5/9$

$$= 11{,}798817$$

verdadero error $= |11{,}775021 - 11{,}798817| =$

$$= 0{,}023796$$

Escaneado con CamScanner