Algoritmos de integración numérica

Recepción: Agosto de 2007 / Aceptación: Octubre de 2007

[1] Eduardo Raffo Lecca
[2] Rosmeri Mayta Huatuco
[3] Victor Perez Quispe

RESUMEN

En este artículo se desarrollan e implementan, los algoritmos de integración numérica, que permiten solucionar problemas de ciencias e ingeniería; tales como el cálculo de áreas, volúmenes, mecánica aplicada, y ecuaciones diferenciales (sistemas dinámicos).

Dada la necesidad de contar con resultados de gran precisión, se analiza las diferentes reglas de integración numérica; basadas en los errores que ocurren cuando el integrando es reemplazado por un polinomio de interpolación P(x), conocida como las fórmulas de integración de Newton-Cotes. La regla de la extrapolación de Richardson, puede ser aplicada a cualquier fórmula de cuadratura de Newton-Cotes; o cualquier computación que se encuentra basada en una rejilla de ancho h y un error descrito como una potencia de h.

Palabras clave: Integración numérica, Integración automática, Integración adaptativa, recursividad divide y vencerás.

NUMERIC INTEGRATION, ALGORITHMS

ABSTRACT

In this article are developed and implemented, the integration of numerical algorithms, which allow solve problems in science and engineering, such as the calculation of areas, volumes, applied mechanics, differential equations (dynamic systems).

Given the need for a high-precision, he examines the different rules of numerical integration; based on the errors that occur when the integrand is replaced by a polynomial interpolation P (x), known as the formulas of integration Newton - Cotes. The rule of Richardson extrapolation can be applied to any formula squaring Newton-Cotes, or any computer which is based on a grid of wide h and error described as a power h.

Key words: Numerical Integration, Automatic Integration, Adaptive Integration, Divide-and-Conquer Recurrence.

INTRODUCCIÓN

La integración numérica o cuadratura numérica, consiste en evaluar la integral definida

$$I_{[a,b]} = \int_a^b f(x)dx$$

o el equivalente, a resolver I ≤ y(b) en la ecuación diferencial.

$$\frac{dy}{dx} = f(x)$$

con la condición de valor inicial .

Este problema tuvo su origen, antes de la invención del cálculo y hoy, gracias a la computadora, es utilizado para evaluar las integrales que no pueden ser computadas analíticamente; o cuando f(x) es conocida para un conjunto de puntos (cuadro 1).

La evaluación de las integrales se denomina cuadratura; desde un viejo problema en geometría conocido como la cuadratura griega del círculo, por medio de polígonos regulares inscritos y circunscritos; proceso que le valió a Arquímedes para acotar el valor de [1]. Cuadratura es sinónimo de encontrar áreas y volúmenes.

Suponga que f(x) es una función que se encuentra acotada en [a, b]. Al dividir en n subintervalos, se tienen los puntos siguientes:

$$a = x_0 < x_1 < x_2 < \ldots < x_n = b$$

Sea ξ_i, un punto cualquiera en $x_{i-1} \leq \xi_i \leq x_i$ y el ancho $\Delta_i = x_i - x_{i-1}$; se define la suma de Riemann como:

$$\sum_{i=1}^{n} \Delta_i \; f(\xi_i)$$

Para el caso de n , se obtiene la integral de Riemann sobre el intervalo [a, b]; esto es

$$S = \int_a^b f(x)dx$$

La regla generalizada de la cuadratura, corresponde al método de aproximación a la integral, como una combinación lineal de los valores del integrando; y viene dada por

$$I = \int_a^b f(x)dx = \sum_{i=1}^{n} w_i \; f(x_i) + E_n$$

(1) Ingeniero Industrial. Profesor del Departamento de Ingeniería de Sistemas e Informática. UNMSM.
E-mail: eraffol@unmsm.edu.pe
(2) Ingeniera Industrial. Profesora del Departamento de Sistemas e Informática. UNMSM.
E-mail: rmaytah@unmsm.edu.pe
(3) Ingeniero Industrial. Profesor del Departamento de Ingeniería de Sistemas e Informática. UNMSM.
E-mail: vperezq@unmsm.edu.pe

Se han generado muchas reglas de cuadratura destacándose: Punto medio, rectángulo, trapezoide y Simpson.

MÉTODOS DE INTEGRACIÓN

Cuando el integrando es reemplazado por un polinomio de interpolación P(x), y se cumple

$$\int_a^b f(x)dx \approx \int_a^b P(x)dx$$

entonces se dice que las fórmulas de integración son de Newton-Cotes.

De la teoría de interpolación por polinomios, se tiene que un polinomio de grado n o menos, es;

$$P_n(x_i) = f_i = f(x_i) \quad , \; i = 0,1,2,...,n$$

Por interpolación de Lagrange:

$$P_n(x) = \sum_{i=0}^{n} f_i \, L_i(x)$$

$$L_i(x) = \prod_{k \neq i} \frac{x - x_k}{x_i - x_k}$$

y haciendo x ≤ a + t h

$$\varphi_i(t) = L_i(x) = \prod_{k \neq i}^{n} \frac{t - k}{i - k}$$

Se tiene la integración

$$\int_a^b P(x)dx = \sum_{i=0}^{n} f_i \int_a^b L_i(x)dx \quad , dx = h\,dt$$

$$= h \sum_{i=0}^{n} f_i \, \beta_i$$

con: $\sum_{i=0}^{n} \beta_i = n$

Si $\sigma_i \leq s\beta_i$ donde s es común denominador, tal que

$$\int_a^b P_n(x)dx = h \sum_{i=0}^{n} f_i \, \beta_i = \frac{(b-a)}{ns} \sum f_i \, \sigma_i$$

En el cuadro 1 se consideran los resultados de los valores de Newton-Cotes. Extendiendo el cuadro, a otros valores corresponden a n puntos. Así Boole es también denominado como de 5 puntos (por existir una interpolación con un polinomio de 4 grados).

Cuando el intervalo [a,b] es razonablemente grande, se acostumbra a dividirlo en n segmentos, y aplicar los esquemas de cuadraturas de Newton-Cotes :

Cuadro 1. Valores de Newton Cotes

N	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	ns	Nombre
1	1	1					2	Trapezoide
2	1	4	1				6	Simpson
3	1	3		1			8	Regla 3/8
4	7	32		32	7		90	Boole
5	19	75		50	75	19	288	Regla 6 puntos

Fuente [2]

Cuando el intervalo [a,b] es razonablemente grande, se acostumbra a dividirlo en n segmentos, y aplicar los esquemas de cuadraturas de Newton-Cotes

$$a = x_0 < x_1 < x_2 < ... < x_n = b$$

$$\int_a^b f(x)dx = \sum_{i=1}^{h} \int_{x_{i-1}}^{x_i} P(x)dx$$

La regla del trapezoide extendida, se desarrolla como una suma de áreas:

$$\int_a^b f(x)dx = \frac{h}{2}(y_0 + 2y_1 + 2y_2 + ... + 2y_{n-1} + y_n)$$

Para aplicar la regla de Simpson a 1/3 en forma compuesta, se requiere que n sea un número par:

$$\int_a^b f(x)dx = \sum_{i=1}^{n/2} A_i$$

$$= \frac{h}{3}\left[y_0 + 4(y_1 + y_3 + y_5 + ...) + 2(y_2 + y_4 + y_6 + ...) + y_n \right]$$

$$= \frac{h}{3}\left[y_0 + 4y_{\text{impares}} + 2y_{\text{pares}} + y_n \right]$$

ERRORES

El error en la aproximación según Steffenson es expresado como sigue:

$$\int_a^b P_n(x)dx - \int_a^b f(x)dx = h^{p+1} K f^{(p)}(\xi) \quad , \quad \xi \in [a,b]$$

Los valores de P y K dependen solamente de n y no del integrando f(x).

El supuesto de f(x), es ser continua y poseer derivadas de alto orden. El término error para Newton-Cotes, tiene la forma $Kh^p f^{(p-1)}(\xi), \; con \; \xi$ en [a,b]; donde K es una constante.

Cuando no existe información acerca de las derivadas de alto orden; es posible estimar el error, si la integral es computada, usando dos diferentes valores[3].

La evaluación de la integral es: $I^* \leq {}_a^b\, f(x)dx$ y puede ser calculada como:

$$I^* \leq I_1 + E_1 \leq I_2 + E_2$$

Sea I_n y E_n, las estimaciones de la integral y su error asociado para una fórmula compuesta con n intervalos de un trapezoide; se tiene:

$$I^* \leq {}_a^b\, f(x)dx \leq I_{n1} + E_{n1} \leq I_{n2} + E_{n2}$$

Donde n_1 y n_2 son valores diferentes de n

$$\frac{E_{n2}}{E_{n1}} \quad \frac{\dfrac{(b \quad a)^3}{12n_2^2} f''(\ _2)}{\dfrac{(b \quad a)^3}{12n_1^2} f''(\ _1)}, \qquad _1\, y \quad _2 \qquad a,b$$

Asumiendo $f''(\ _1)$ y $f''(\ _2)$ iguales: $\dfrac{E_{n2}}{E_{n1}} \quad \left(\dfrac{n_1}{n_2}\right)^2$

El valor de la integral viene dado por: $I^* \leq I_{n1} + E_{n1}$, y desde la igualdad

$$E_{n1} - E_{n2} \leq I_{n2} - I_{n1}$$

$$E_{n1} \quad \left(\frac{n_1}{n_2}\right)^2 E_{n1} \quad I_{n2} \quad I_{n1}$$

$$E_{n1} \quad \frac{I_{n2} \quad I_{n1}}{1 \quad \left(\dfrac{n_1}{n_2}\right)^2}, \quad y$$

$$I_{n1} \quad \frac{I_{n2} \quad I_{n1}}{1 \quad \left(\dfrac{n_1}{n_2}\right)^2}$$

Si $n_2 \leq 2n_1$, la integral I^* se convierte en:

$$I^* \quad I_{n1} \quad \frac{I_{n2} \quad I_{n1}}{\dfrac{3}{4}} \qquad I^* \quad \frac{4}{3} I_{n2} \quad \frac{1}{3} I_{n1}$$

Conocida como la aproximación de Richardson para el Trapezoide.

Para la regla de Simpson a 1/3 para datos compuestos, el error viene dado por:

$$E \quad \frac{\left(b \quad a\right)^5}{2880n^4} f^{(4)}$$

Se cumple que: $I^* \quad I_{n1} \quad \dfrac{I_{n2} \quad I_{n1}}{1 \quad \left(\dfrac{n_1}{n_2}\right)^4}$

Y con $n_2 \leq 2n_1$, se tiene que: $I^* \quad \dfrac{16}{15} I_{n2} \quad \dfrac{1}{15} I_{n1}$

Se puede generalizar estas fórmulas, al plantearse como:

$$T_{n,j\ 1} \quad \frac{4^j\, T_{n\ 1,j} \quad T_{n,j}}{4^j \quad 1}$$

La filosofía de la extrapolación de Richardson, como así es llamada, puede ser aplicada a cualquier fórmula de cuadratura de Newton-Cotes; o cualquier computación que se encuentra basada en una rejilla de ancho h y un error descrito como una potencia de h[4].

Aplicaciones de los métodos de integración

Haciendo uso de los valores del cuadro 1, en la figura 1, se presenta el *script* del M-file Quadratura.m, que evalúa estas reglas.

La regla de Simpson a 1/3, es probablemente la más utilizada, de todas las fórmulas de integración.

Los script para las fórmulas del Trapecio y Simpson a 1/3, son presentadas en las figuras 2 y 3.

La evaluación analítica de la función $f(x) \quad x^2 \quad 2x \quad 2$ en el intervalo corresponde al valor de 2/3. En la figura Nº 4, se presenta los cálculos para las reglas del Trapecio y Simpson, usando 10 intervalos. La regla de Simpson es la que más se aproxima.

Figura 1. M-file Quadratura.m

```
function Quadratura(f,a,b)
% Cuadraturas básicas
% % Datos
% f  =  el nombre de la función como string
% a  =  límite inferior
% b  =  límite superior
% h = longitud del segmento
% n = numero de segmentos
% Resultados
% p=integración
  n=1;
  h = (b-a)/n;
  p1=h*(feval(f,a)+feval(f,b))/2;
  n=2;
  h = (b-a)/n;
  p2=h*(feval(f,a)+4*feval(f,(a+b)/2)+feval(f,b))/3;
  n=3;
  h=(b-a)/n;
  p3=3*h*(feval(f,a)+3*feval(f,a+h)+3*feval(f,a+2*h)+feval(f,b))/8;
  n=4;
  h=(b-a)/n;
  p4=4*h*(7*feval(f,a)+32*feval(f,a+h)+ …
       12*feval(f,a+2*h)+32*feval(f,a+3*h)+7*feval(f,b))/90;

  fprintf('Area para Trapezoide : %10.5f\n',p1);
  fprintf('Area para Simpson 1/3: %10.5f\n',p2);
  fprintf('Area para Simpson 3/8: %10.5f\n',p3);
  fprintf('Area para Boole      : %10.5f\n',p4);
```

Fuente: Elaboración propia

Figura 2. M-file Trapezoide.m

```
function p≤Trapezoide(f,a,b,n)
% Datos
% f   ≤el nombre de la funcion como string
% a   ≤limite inferior
% b   ≤limite superior
% h   ≤longitud del segmento
% x   ≤es el vector x
% y   ≤es el vector f(x), f en forma vectorial
% n   ≤numero de segmentos, o n+1 puntos
% Resultados
% p   ≤integracion
   h≤(b-a)/n;
   x≤a+h*(0:n);
   y≤feval(f,x);
   p≤0.5*h*(2*sum(y)-y(1)-y(n+1));
```

Fuente: Elaboración propia

Figura 3. M-file Simpson3.m

```
function p≤Simpson3(f,a,b,n)
% Simpson a 1/3
% % Datos
% f  ≤ el nombre de la función(vectorial) como string
% a  ≤ limite inferior
% b  ≤ limite superior
% h  ≤ longitud del segmento
% x  ≤ es el vector x
% y  ≤ es el vector f(x)
% n  ≤ numero de segmentos
% Resultados
% p  ≤ integracion
   h≤(b-a)/n;
   x≤a+h*(0:n);
   y≤feval(f,x);
   p≤h*(2*sum(y)+2*sum(y(2:2:n))-y(1)-y(n+1))/3;
```

Fuente: Elaboración propia

Un esquema de integración automática, ayuda al usuario a calcular cualquier integral.

En [5], se detallan los requisitos, para un integrador automático:

Los límites de integración.
Rutina para evaluar f(x).
La tolerancia.
Cota superior del número de evaluaciones.

Los esquemas de integración automática, pueden ser clasificados en: adaptativa o no adaptativa; e iterativos o no iterativos.

Los esquemas de cuadratura no adaptativa para computadoras, usan generalmente una secuencia de puntos, de acuerdo a un esquema fijo; independiente del integrando.

Cuando se evalúa por la regla del Trapezoide se utiliza la fórmula:

$$T(h) \leq 0.5 \frac{b \quad a}{n} \quad f(a) \quad 2 \sum_{i\ 1}^{n\ 1} f(x_i) \quad f(b)$$

Cuando n ≤ 1, el área es igual a

$$T(h) \leq 0.5 (b \quad a)[f(a) \quad f(b)]$$

Si en la iteración i+1, el número de segmentos se duplica con respecto a i, se encuentra que el número de puntos interiores nuevos es $2^{i\ 2}$ (ver figura 5)

Sean:

$$Area_i \quad 0.5 \frac{b \quad a}{n_i} \quad f(a) \quad 2(Puntos\ Interiores) \quad f(b)$$

$$Area_{i\ 1} \quad 0.5 \frac{b \quad a}{n_{i\ 1}} \quad suma_i \quad 2suma_{i\ 1}$$

Siendo suma$_{i+1}$, el acumulado de los nuevos puntos interiores.

De la expresión:

$$suma_i \quad f(a) \quad 2(\text{Puntos interiores}) \quad f(b)$$

$$Area_{i\ 1} \quad 0.5\ Area_i \quad \frac{b \quad a}{n_i} \quad Suma_{i\ 1}$$

$$0.5(Area_i \quad h_i\ Suma_{i\ 1})$$

Un esquema de integración no adaptativa del tipo iterativo, se presenta a continuación. Este hace uso de la regla del trapecio en forma refinada, con su correspondiente aproximación de Richardson. La aproximación para el Trapecio corresponde a:

$$I_i* \quad \frac{4}{3} I_i \quad \frac{1}{3} I_{i\ 1}$$

Figura 4. Comparación entre Trapezoide y Simpson a 1/3

```
f≤inline('x.^2+2*x-2');
>> Trapezoide(f,0,1,10)

ans ≤  -0.66500000000000

>> Simpson3(f,0,1,10)

ans ≤  -0.66666666666667
```

Fuente: Elaboración propia

Figura 5. Proceso del trapecio refinado

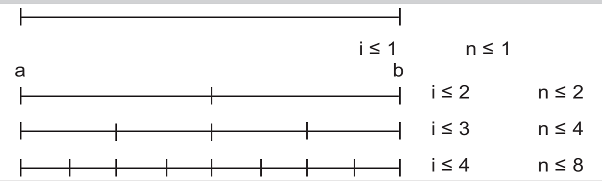


Figura 6. M-file de Trapecio.m

```
function Trapecio(fdex,a,b,j)
% Trapecio Refinado
% Datos
% fdex ≤es la funcion que ingresa como un string
% a,b  ≤los valores extremos del intervalo
% j    ≤el valor de la iteración
% Resultados
% N    ≤el valor de n
% Area ≤valor del area

     global Area N;

    if j≤≤1
      Area≤0.5*(b-a)*(feval(fdex,a)+feval(fdex,b));
      N≤1;
    else
        H≤(b-a)/N;
        x≤a+H/2;
        Suma≤0;
        for i≤1:N
            Suma≤Suma+feval(fdex,x);
            x≤x+H;
        end
        Area≤0.5*(Area+H*Suma);
        N≤2*N;
     End
```

Fuente: Elaboración propia

Figura 7. M-file de TrapecioRichardson.m

```
function [I n]≤TrapecioRichardson(fdex,a,b,ERROR)
% Trapezoide con Richardson
% Datos
% a   ≤limite inferior de la integral
% b   ≤limite superior de la integral
% Resultados
% A ≤área por Trapecio refinado
% I ≤área por extrapolación de Richardson
  global Area N;

  if nargin < 4, ERROR ≤ 0.0005; end
  j≤1;
  Trapecio(fdex,a,b,j);n≤2;
  A≤Area;I≤A;
  while 1
     A0≤A;I0≤I;
     j≤j+1;n≤n+N;
     Trapecio(fdex,a,b,j);
     A≤Area;
     I≤(4*A-A0)/(4-1);
     if abs(I-I0)<≤ERROR;
        break;
     end
  end
```

Fuente: Elaboración propia

el algoritmo encuentra aproximaciones de I_i*, y finaliza con el siguiente criterio:

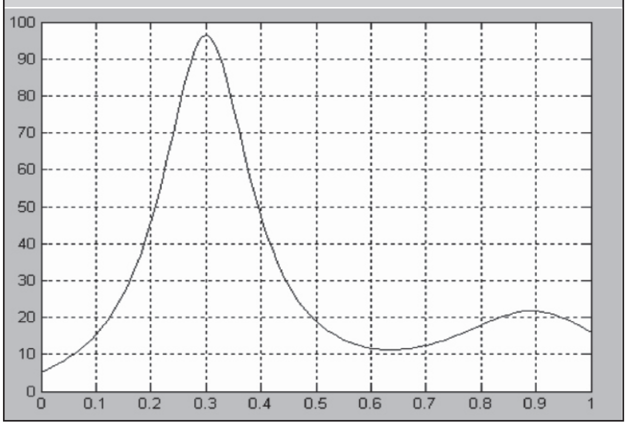
$$\left| I_{i\ 1}* \quad I_i* \right| \quad \varepsilon$$

Las figuras 6 y 7, corresponden a los *script* Trapecio.m y TrapecioRichardson.m. Este último *script* (que invoca Trapecio.m), implementa un algoritmo muy simple y robusto para la integración no adaptativa[1].

La función humps, es un buen *benchmark* para efectuar métricas de pruebas, en muchos procesos de computación.

Declarando la función humps.m para MATLAB:
function y ≤ humps(x)

Figura 8. Función humps



Fuente: Elaboración propia

y ≤ 1,0/((x - 0,3).^2 + 0,01) + 1,0/((x- 0,9).^2 +0,04) - 6;

La función Humps, tiene como forma, la gráfica de la figura Nº 8.

>> X≤ 0:0.01: 1; y≤humps(x);plot(x,y),gris on

Aplicando Trapecio refinado, con la extrapolación de Richardson, se consigue el resultado que aparece en la figura 9. En donde la integral es 29,85832730748868 con 65 invocaciones a la función.

El proceso simple de evaluar numéricamente una integral, es usando puntos equidistantes, de un ancho h. Una cuadratura es adaptativa, cuando involucra la selección de los puntos que van a ser evaluados; de tal manera que se consiga una determinada precisión.

Los esquemas de cuadratura adaptativa para computadoras, usan generalmente la regla del trapezoide o Simpson, dividiendo cualquier región del estado anterior. La critica a las fórmulas compuestas que utilizan nodos equidistantes, es que al integrar una función en un intervalo que contiene regiones,

Figura 9. Ejecución de TrapecioRichardson.m

```
>>[Q,n]≤TrapecioRichardson@humps,0,1)
Q≤ 29,85832730748868
n≤ 65
```

Fuente: Elaboración propia

Figura 10. SimpGold.m

```
function [Q nc]≤SimpGold(f,a,b,TOL)
% Simpson a 1/3 con integración adaptativa
%  Datos
% f   ≤el nombre de la función(vectorial) como string
% a   ≤limite inferior
% b   ≤limite superior
% h   ≤longitud del segmento
% x   ≤es el vector x
% y   ≤es el vector f(x)
% Resultados
% Q  ≤integración
% nc ≤numero de invocaciones a f
    if nargin <4, TOL≤0.00000005; end
    h≤(b-a)/2;
    c≤(a+b)/2;
    x≤[a c b];
    y≤feval(f,x);
    nc≤3;nivel≤1;
    Q0≤h/3*(y(1)+4*y(2)+y(3));
    [Q n]≤SimpRecur(f,a,c,b,y(1),y(2),y(3),nivel,TOL,Q0);
    nc≤nc+n;
    return
```

Fuente: Elaboración propia

Figura 11. SimpRecur.m

```
function [Q nc]≤SimpRecur(f,a,c,b,fa,fc,fb,nivel,TOL,Q0)
% Simpson a 1/3 con recursion Divide y Venceras
   if nivel > 20
      nc≤0;
      Q≤Q0;
      return
   end
   h≤(c-a)/2;
   c1≤a+h;c2≤c+h;
   fc1≤feval(f,c1);fc2≤feval(f,c2);
   Q1≤h/3*(fa+4*fc1+fc);Q2≤h/3*(fc+4*fc2+fb);
   nc≤2;
   Q≤Q1+Q2;
   if abs(Q-Q0) > TOL
       nivel≤nivel+1;
       [Q1 n1]≤SimpRecur(f,a,c1,c,fa,fc1,fc,nivel,TOL/2,Q1);
       [Q2 n2]≤SimpRecur(f,c,c2,b,fc,fc2,fb,nivel,TOL/2,Q2);
       nc≤nc+n1+n2;
       Q≤Q1+Q2;
   End
```

Fuente: Elaboración propia

donde la función varía en gran medida, y en otras en donde la variación es pequeña, tal como ocurre con la función humps, no es factible su aplicación.

El término de cuadratura adaptativa, viene porque estos métodos son capaces de predecir el grado de la variación funcional, y se adaptan a un tamaño del paso de las necesidades.

Tomando a la regla de Simpson a 1/3, con un paso de , se consigue la integral:

$$ {}_{a}^{b} f(x)dx \quad S(a,b) \quad \frac{h^5}{90} f^{(4)}(\quad), \qquad (a,b) $$

Una importante técnica de solución de problema, que hace uso de la recursividad es Divide y Vencerás[6]. Divide y Vencerás, consiste de dos partes:

1. Divide, pequeños problemas, son resueltos recursivamente, excepto el caso base.

2. Vencerás, la solución al problema original está formado desde la solución a los subproblemas.

La regla de Simpson por recursividad, es mostrada como sigue:

$$ \left| {}_{a}^{b} f(x)dex \quad S(a,\frac{a \quad b}{2}) \quad S(\frac{a \quad b}{2},b) \right| $$

Cuando la estimación del error no es válida, se aplica la regla de Simpson a los nuevos subintervalos [a,(a+b)/2] y [(a+b)/2, b], teniendo en consideración que los errores parciales de cada intervalo, corresponden al valor .

En las figuras 10 y 11, se presentan los *script*, que hacen uso del Divide y Vencerás para Simpson 1/3. Este es un algoritmo de integración adaptativa.

En la figura 12, se ejecuta SimpGold.m, encontrándose la precisión arriba del decimal doceavo.

CONCLUSIONES

La crítica a las fórmulas compuestas que utilizan nodos equidistantes, es que al integrar una función en un intervalo que contiene regiones, donde la función varía en gran medida, y en otras en donde la variación es pequeña, tal como ocurre con la función humps, no es factible su aplicación. Cuando la estimación del error no es válida, se aplica la regla de Simpson a los nuevos subintervalos (Figura 11), teniendo en consideración que los errores parciales de cada intervalo, corresponden a la mitad del error anterior.

Los algoritmos que se realiza se basa en el esquema del trapecio refinado, hace uso de la filosofía de Richardson, y ha demostrado en la práctica producir resultados precisos en las

Figura 12.: Ejecución de SimpGold.m

```
>>[Q,n]≤SimpGold(@humps,0,1)
Q≤ 29,85832539558330
n≤ 2124
```

evaluaciones de integrales, utilizando pocas iteraciones y con errores pequeños.

La producción de resultados precisos en integración numérica, con errores muy pequeños, trae consigo el aumento de evaluaciones.

La utilización de la recursividad hace posible construir buenos algoritmos en integración adaptativa. Con la técnica del Divide y Vencerás, se ha obtenido un algoritmo que da resultados precisos, en el menor tiempo.

REFERENCIAS BIBLIOGRÁFÍCAS

1. David, Philip J. y Rabinowitz, Philip, (1975) "Methods of Numerical Integration", Academic Press.

2. Raffo Lecca, Eduardo, (2005) "Métodos Numéricos Para Ciencia e Ingeniería con MATLAB", Raffo Lecca editores.

3. Carnahan, Brice, Luther, H. A., (1969) "Applied Numerical Methods", John Wiley & Sons, Inc.

4. Acton, Forman S., (1970) "Numerical Methods That Work" , Harper & Rows, Publishers, New York.

5. Forsythe, George Elmer, (1977) "Computer Methods for Mathematical Computations" , Prentice-Hall series in automatic computation.

6. Weiss, Mark Allen C., (1996) "Algorithms, Data Structures, and Problem Solving with C++", Addison-Wesley.