

Laboratorio 7: Integración numérica en MATLAB.

Sebastián Valencia Calderón
201111578

1 Introducción

Desde los cursos de cálculo elemental, se conoce el proceso analítico para hallar las integrales definidas y no definidas de una función analítica. Se conocen métodos analíticos para su solución y las aplicaciones de la integración. Sin embargo, en el desarrollo de las ciencias y de la ingeniería, resulta conveniente hallar métodos numéricos para la solución aproximada de integrales, haciendo uso de algoritmos numéricos. Un ejemplo típico de aplicación indirecta de estos métodos a problemas en ciencias e ingeniería, es la solución numérica o mixta de ecuaciones diferenciales ordinarias. (Referencias [3], [5]).

Existen varios métodos para la solución de este problema. Los más elementales son los del trapecio, y el basado en la regla de Simpson, el presente laboratorio, presenta la solución de algunos problemas para ser resueltos haciendo uso de integración numérica, para integración en \mathbb{R} , \mathbb{R}^2 , y \mathbb{R}^3 . El desarrollo de estos problemas, corresponde más a un estudio de los métodos dispuestos de manera nativa en MATLAB, que un estudio del diseño y desarrollo de los mismos, por lo tanto, se pretende aprender a usar las funciones `trapz()`, `quad()`, `quad2d()`, y `triplequad()`. De manera concreta, el desarrollo de los ejercicios, pretende comprometerse con el siguiente objetivo:

1. Presentar y reconocer el funcionamiento de algunos algoritmos de integración numérica en MATLAB.

2 Procedimiento

Para cumplir los objetivos enumerados anteriormente, se estudia el uso de los métodos básicos de integración numérica. Es decir, los enumerados en la introducción. Con estos, se desarrolla un script para comparar su desempeño sin tener en cuenta su implementación o diseño, sólo su uso, y a partir del uso de ellos en cuatro funciones se compra su desempeño y casos de uso en la práctica, además, se puede inferir sobre su diseño e implementación. Posteriormente, se estudia el uso de las funciones dispuestas en MATLAB, para la integración en dimensiones mayores.

3 Resultados

A continuación, se exponen los resultados y las herramientas de ejecución para cada uno de los problemas propuestos.

3.1 Método del trapecio y de Simpson en una dimensión en MATLAB

Haciendo uso de las funciones disponibles en Matlab para la integración numérica por los métodos del trapecio y de Simpson, se calculan las siguientes integrales en el intervalo $0 \leq x \leq 15$. Se explican y justifican los parámetros empleados para esto.

$$C(x) = \int_0^x \cos\left(\frac{\pi\tau^2}{2}\right) d\tau \quad S(x) = \int_0^x \sin\left(\frac{\pi\tau^2}{2}\right) d\tau$$
$$C_1(x) = \int_0^x \cos(\tau^2) d\tau \quad S_1(x) = \int_0^x \sin(\tau^2) d\tau$$

Script 1: Evaluación de las integrales enunciadas, en el intervalo $0 \leq x \leq 15$, haciendo uso de las dos funciones de MATLAB. (functions.m)

```
1 definitions;           % Defining the functions
2 step = 0.001;         % Defines the steps of the iteration vector for the
   integral
3 eval = 0:step:15;     % Iteration vector for the integral
4 n = length(eval);     % Length og vector
5
6 % Vectors for storing each integral for each item in te eval vector for
7 % each method
8 f1simp = zeros(n, 1); f1trap = zeros(n, 1);
9 f2simp = zeros(n, 1); f2trap = zeros(n, 1);
10 f3simp = zeros(n, 1); f3trap = zeros(n, 1);
11 f4simp = zeros(n, 1); f4trap = zeros(n, 1);
12
13 % Evaluation of each function in the iteration vector
14 f1y = f1(eval); f2y = f2(eval); f3y = f3(eval); f4y = f4(eval);
15
16 for i=1:n
17     % Integration vector up to the current evaluation value
18     fx = 0:step:eval(i);
19     try
20         % Trapezoidal and Simpson's rule application for each function
21         f1trap(i) = trapz(fx, f1(fx)); f1simp(i) = quad(f1, 0, eval(i));
22         f2trap(i) = trapz(fx, f2(fx)); f2simp(i) = quad(f2, 0, eval(i));
23         f3trap(i) = trapz(fx, f3(fx)); f3simp(i) = quad(f3, 0, eval(i));
24         f4trap(i) = trapz(fx, f4(fx)); f4simp(i) = quad(f4, 0, eval(i));
25     catch, continue;
26     end;
27 end;
28
29 % Custom plotting procedures, irrelevant
30 customplot1; customplot2; customplot3; customplot4;
```

El script anterior, muestra la evaluación iterativa para cada punto del vector de integración de cada una de las funciones, el uso de la función de MATLAB para el método del trapecio (`trapz()`), requiere la función a integrar y un vector de la evaluación del vector de integración hasta el punto actual. Asimismo, la regla de Simpson, requiere la función, y ambos límites de integración. Esto es implementado haciendo uso de la función `quad()`. Los scripts para graficar, no se incluyen por su poca relevancia con el problema en cuestión.

El scrip referido, el cual contiene las definiciones de las funciones es:

Script 2: Definición de las funciones a integrar. (`definitions.m`)

```
1 f1 = @(tau) cos(pi / 2 * tau .^ 2);
2 f2 = @(tau) sin(pi / 2 * tau.^ 2);
3 f3 = @(tau) cos(tau .^ 2);
4 f4 = @(tau) sin(tau .^ 2);
```

A continuación, se incluye la gráfica de la integral, obtenida a partir de los métodos numéricos referidos anteriormente. Esto para la primera función.

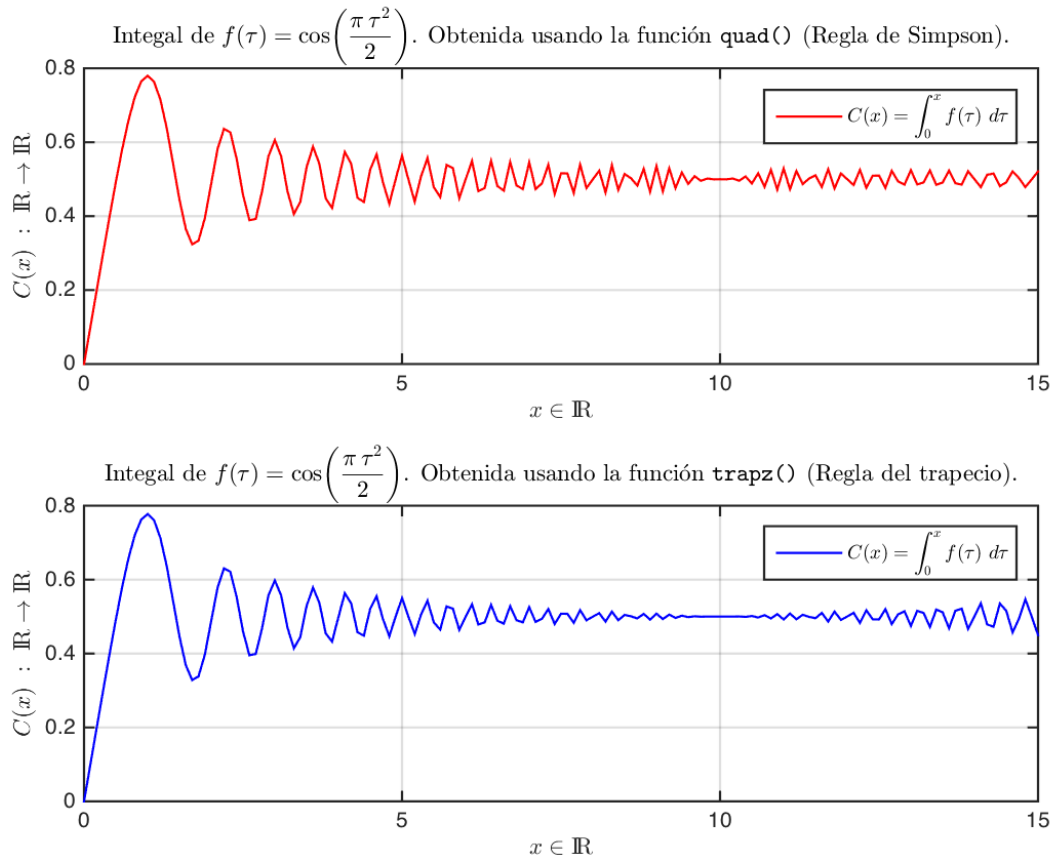


Figura 1: Integral de $f_1(x)$, obtenida numéricamente haciendo uso de las dos funciones referidas, para $0 \leq x \leq 15$, con pasos de 0.1.

Ahora, se incluyen las gráficas para las demás funciones.

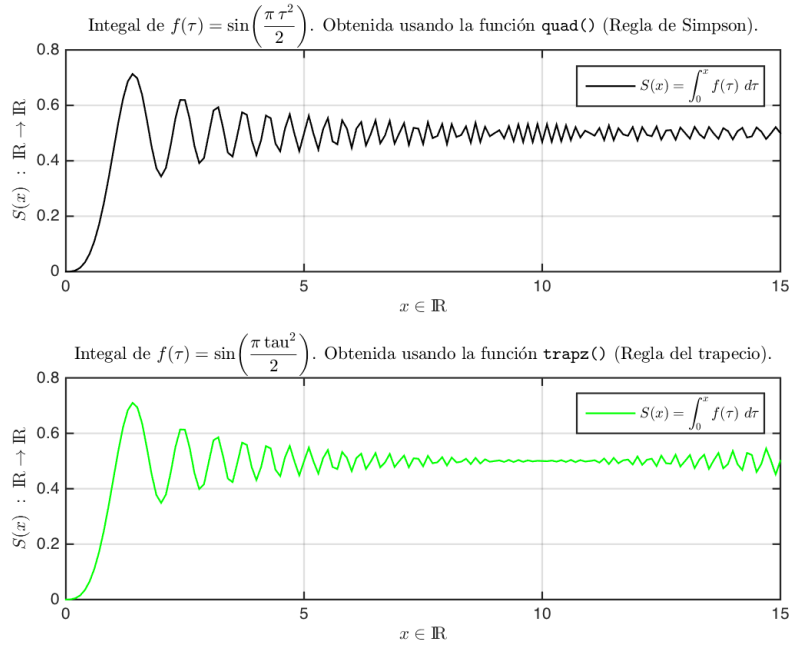


Figura 2: Integral de $f_2(x)$, obtenida numéricamente haciendo uso de las dos funciones referidas, para $0 \leq x \leq 15$, con pasos de 0.1.

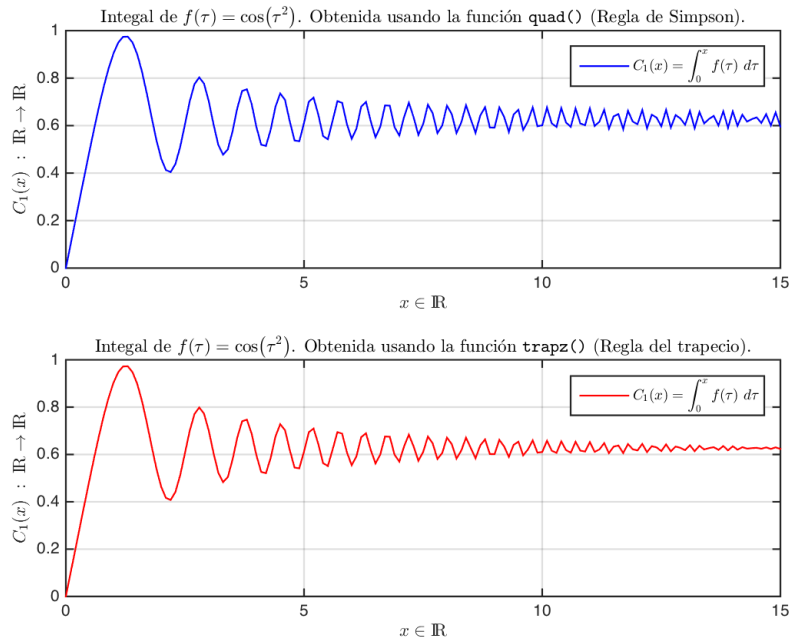


Figura 3: Integral de $f_3(x)$, obtenida numéricamente haciendo uso de las dos funciones referidas, para $0 \leq x \leq 15$, con pasos de 0.1.

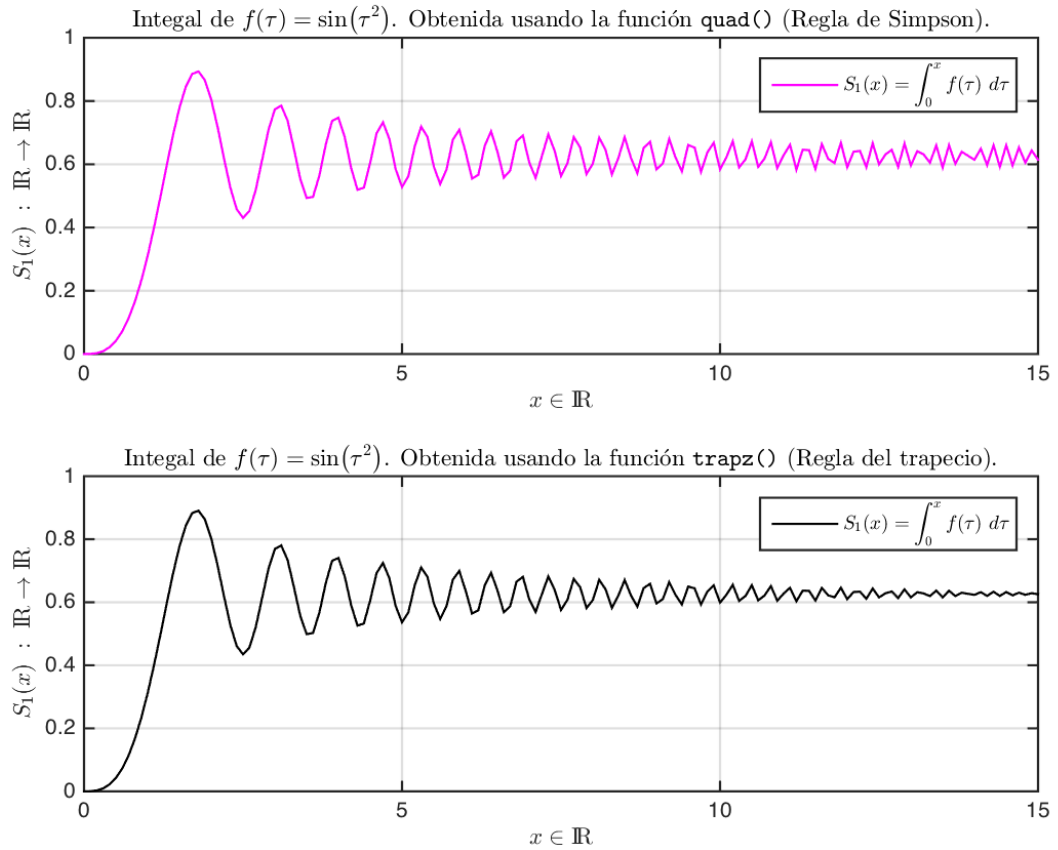


Figura 4: Integral de $f_4(x)$, obtenida numéricamente haciendo uso de las dos funciones referidas, para $0 \leq x \leq 15$, con pasos de 0.1.

Haciendo uso de estas visualizaciones gráficas para cada una de las integrales con los dos métodos, es difícil inferir a cerca del desempeño de cada una en comparación con la otra. Es decir, con estas gráficas, no se observa a primera vista si una es mejor a la otra para el cálculo de la integral numérica de una función analítica. Para mitigar este problema, se hace más preciso el vector de integración, es decir, el paso se agudiza para tener mayores valores en el intervalo de integración, esto corresponde a cambiar el valor de `step` en el primer script (línea 2), a 0.001. El script, toma más tiempo en correr.

Las figuras de esta última simulación, se encuentran a continuación. En estas se observa una especie de ruido obtenido mediante el uso de la regla de Simpson, mientras que emplear la regla del trapecio, resulta en una curva mucho más suave y definida que la otra. Esto último, permite inferir que el desempeño en resultados de la regla del trapecio, haciendo la comparación mediante la forma de la función integral, es mucho mayor. Sin embargo, el resultado numérico es aproximado.

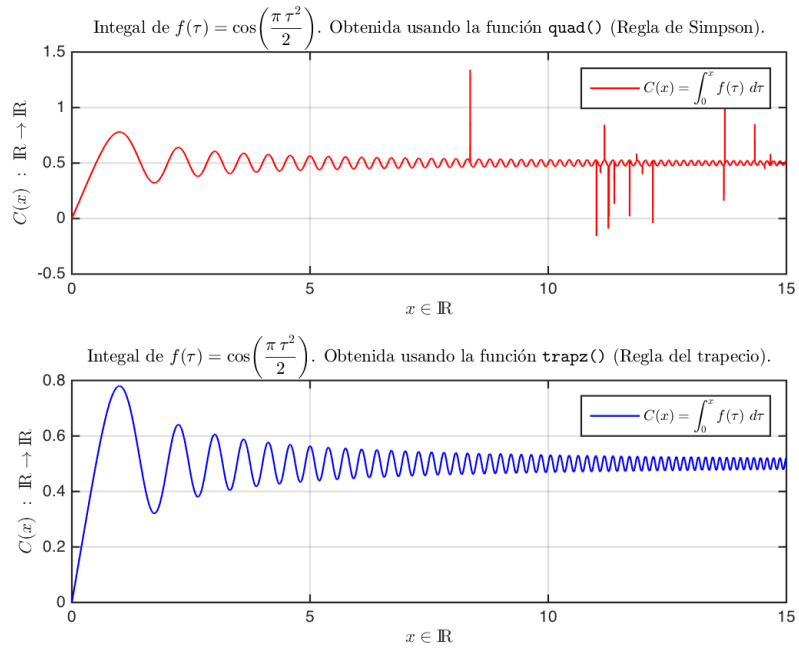


Figura 5: Integral de $f_1(x)$, obtenida numéricamente haciendo uso de las dos funciones referidas, para $0 \leq x \leq 15$, con pasos de 0.001.

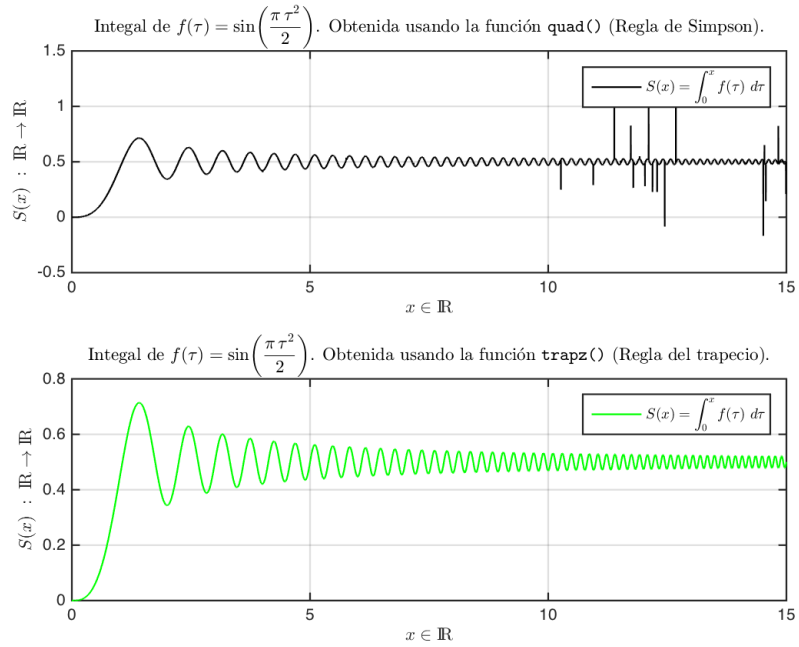


Figura 6: Integral de $f_2(x)$, obtenida numéricamente haciendo uso de las dos funciones referidas, para $0 \leq x \leq 15$, con pasos de 0.001.

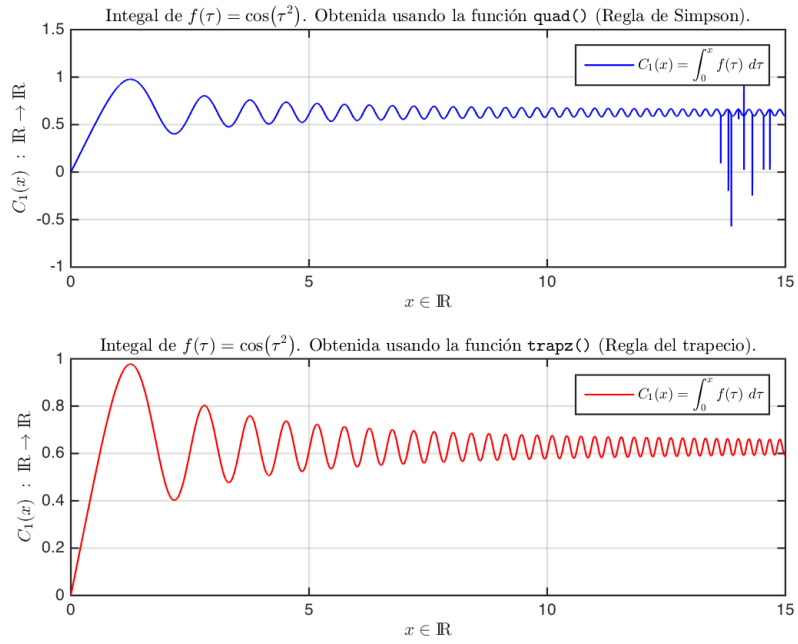


Figura 7: Integral de $f_3(x)$, obtenida numéricamente haciendo uso de las dos funciones referidas, para $0 \leq x \leq 15$, con pasos de 0.001.

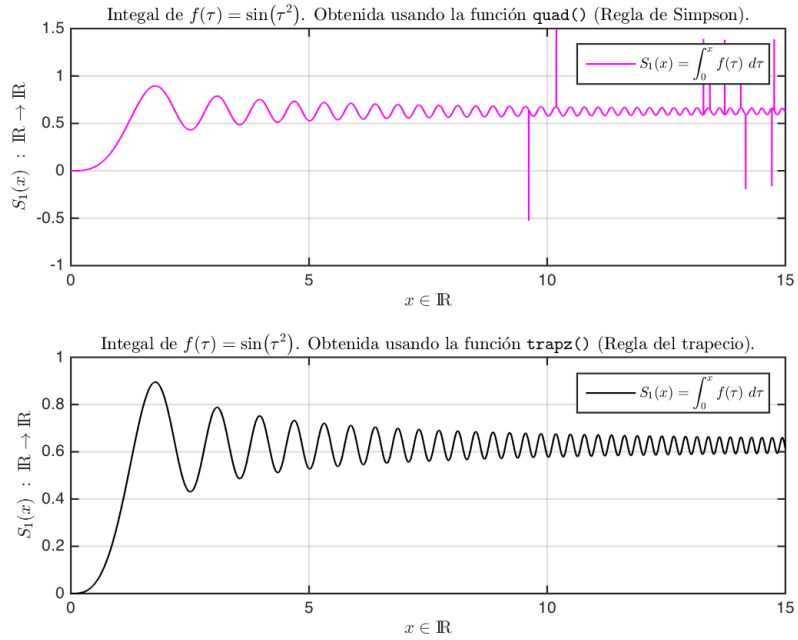


Figura 8: Integral de $f_4(x)$, obtenida numéricamente haciendo uso de las dos funciones referidas, para $0 \leq x \leq 15$, con pasos de 0.001.

3.2 Método del trapecio y de Simpson en varias dimensiones en MATLAB

Las funciones `dblquad()`, `quad2d()`, y `triplequad()`, proveen los medios necesarios para calcular la integral iterada, doble o triple de una función en las dimensiones requeridas. A continuación, se incluye la solución de algunas integrales mediante los procedimientos dispuestos por MATLAB para esta labor.

- Considérese la siguiente integral:

$$\int_0^1 \int_0^1 \frac{1}{1-xy} dy dx$$

Esta integral, puede resolverse en MATLAB, haciendo uso de la función `quad2d`, la cual evalúa de manera numérica una doble integral.

La forma general es: `q = quad2d(fun,a,b,c,d)`, donde se aproxima la función `fun(x, y)`, sobre una región en \mathbb{R}^2 , $a \leq x \leq b$, y $c(x) \leq y \leq d(x)$. La función, es la función a integrar, y `a`, `b`, `c`, y `d`, pueden ser valores numéricos. La solución en MATLAB de esta integral es a partir de:

```
1 f1 = @(x,y) 1 ./ (1 - x.*y);    % Function to handle
2 i1 = quad2d(f1, 0, 1, 0, 1);    % Double integral between the given
    plane
```

El valor de `i1`, es 1.6449.

- Para la integración triple, es decir la integración iterada en tres dimensiones, la función `triplequad`, puede ser usada con la forma general

`triplequad(fname,xl,xu,yl,yu,zl,zu)`

Donde se tienen el nombre de la función, y los intervalos en cada dimensión. Por ejemplo, la siguiente integral, puede resolverse de la siguiente manera.

$$\int_0^1 \int_0^1 \int_0^1 64xy(1-x^2) dz dy dx$$

```
1 f2 = @(x,y) 64.*x.*y.*(1 - x).^2; % Function to handle
2 i2 = triplequad(f2, 0, 1, 0, 1, 0, 1); % Triple integral between the
    given surfave
```

El valor de `i2`, es 1.3333.

- Por último, la función `quad2d`, permite también la integración en el rango de y como funciones de x . El siguiente ejemplo ilustra esto:

$$\int_1^2 \int_{x^2}^{x^4} x^2 y \, dy \, dx$$

```

1  f3 = @(x,y) x.^2.*y;           % Function for the integral
2  c = @(x) x.^2;                 % Lower limit of integration
3  d = @(x) x.^4;                 % Upper limit of integration
4  i3 = quad2d(f3, 1, 2, c, d);   % Double integral over the specified
    plane

```

El valor de `i3`, es 83.9740. La ejemplificación de estos métodos, se encuentra de manera más detallada en las referencias [4] y [7].

A partir del desarrollo de los dos últimos scripts, es posible contestar las siguientes preguntas:

- ¿A qué hacen referencia los parámetros `AbsTol` y `RelTol` de las funciones?
Estos parámetros, hacen referencia a la tolerancia absoluta y relativa del método numérico respectivamente, es decir, que tanta tolerancia sobre cada uno de los dos tipos de errores es posible manjar, controlando así, el numero de iteraciones del algoritmo y las presicion de su resultado. El valor por defecto de ellos es $1 \times E^5$.
- ¿Qué métodos usan estos procedimientos?
El método de solución de las integrales triples, resulta de reducir la integral a una doble sobre una función alterada, por lo que es posible entender el desarrollo de las cuadraturas a partir del entendimiento de los métodos que solucionan las integrales dobles en MATLAB. El método de solución de las integrales dobles, es a partir de cuadraturas, el cual consiste en aproximar la función dividiendo el plano o el espacio en un grid y calculando el área o el volumen del grid aproximado por la función de integración. El diseño y análisis de estas herramientas de cuadratura, se encuentran en las referencias [1], [2], y [6].
- ¿Qué sucede cuando la función se encuentra indeterminada en algún punto dentro de los límites de integración? ¿Cómo maneja MATLAB esta situación?
En este caso, la integral es aproximada mediante la función más cercana a este valor que este definida en este valor.

4 Conclusiones

El desarrollo de los ejercicios propuestos para el laboratorio, sirvieron para conceptualizar los algoritmos fundamentales para la obtención de la integral de algunas funciones analíticas, definidas como funciones en MATLAB, el caso tratado en este laboratorio, es mucho más cercano a cualquier persona por la familiaridad de estas con el cálculo infinitesimal de variables reales. Hallar los valores numéricos de las integrales, resulta muy provechoso para el tratamiento numérico de algunos problemas básicos y algunos avanzados en computación científica.

No es suficiente con saber usar las funciones dispuestas en una herramienta de computación científica para el cálculo de las integrales, es necesario conocer la pertinencia de cada método, conociendo su funcionamiento interno y los casos de uso. Asimismo, es necesario conocer las ventajas y desventajas de cada uno de los métodos, puede que las desventajas de uno no sea tan perjudicial por la naturaleza del problema a aplicar. Desarrollar herramientas gráficas o estadísticas para visualizar o entender estas diferencias, estas ventajas y desventajas, resulta de gran ayuda en el estudio de los distintos algoritmos numéricos.

5 Bibliografía

- [1] Lambers, J. V. *Algorithm QUAD2D: General Two-Dimensional Quadrature*. 1991.
- [2] L.F. Shampine. *Matlab program for quadrature in 2D*. 2008. Applied Mathematics and Computation.
- [3] Yang, W.Y. and Cao, W. and Chung, T.S. and Morris, J. *Applied Numerical Methods Using MATLAB*. 2005. Wiley. Pags. 50 - 57.
- [4] Lopez, C. *MATLAB Symbolic Algebra and Calculus Tools*. 2014. MATLAB Solutions Series - Apress. Pags. 73 - 75.
- [5] Bradie, B. *A Friendly Introduction to Numerical Analysis*. 2006. Pearson Prentice Hall - Featured Titles for Numerical Analysis Series. Página 150 - 174.
- [6] Moler, C.B. *Numerical Computing with MATLAB: Revised Reprint*. 2010. Society for Industrial and Applied Mathematics. Pags. 65 - 98.
- [7] Lindfield, G. and Penny, J. *Numerical Methods: Using MATLAB*. 2012. Elsevier Science. Pags. 281 - 305.