Mat Lab Programmes Models for Irregular Areas within Applications In Calculus, Physics & Engineering

Dr. Faisal Abdulateef Shaghati

Open Educational College Mathematics Department Iraq - Baghdad - Al adhamiya - near the judicial institute

e-mail : fisalshagati@yahoo.com

Abstract
The aim of this paper is to introduce well constructed as well as relevantly simplified models of Mat lab programmes for irregular areas within applications in calculus, physics & engineering, also it will pay attention to the informative aspects related with programmes construction besides the mathematical concepts which may appear as theoretical viewpoints, computational outputs or appropriate clarifications. Moreover, it includes a verification via the stronger form of Fubini's Theorem & Mat lab models once in a while, also it offers a new of its kind models for Fresnels Integrals as well as Euler Spiral, the author would like to confirm that his models - in their own selves and not in comparison within others - have no precedence across the subject literatures.
Keywords : Mat Lab models, dual area, Multi-area, Fresnels, Fresnelc, Euler's Spiral, Fubini's theorem

1. Introduction

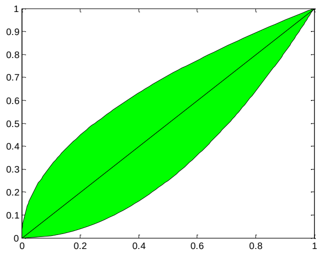
It's usually referred to irregular area as area under a curve, but at this point there is an overlapping between the natural language & the mathematical language, concretely in AmE, a curve means concave or twisted path as (a curve in the road)* or (curve ball, in the game of baseball)*, while in mathematics frequently it assigns to concave up or/and down path, whereby explicitly referred to it as a geometrical figuration of a function, while circles, polygons, arcs and other closed or open geometrical figures aren't underlined as curves, due to that they don't satisfy the function-tests, therefore these figures are considered as relations, although one can considered them as connected functions, moreover for wider & mathematically conceptualized scope, l would like to quote Bruce Simmons [1](A curve is a word used to indicate any path, whether actually curved or straight , closed or open, it can be on a plane or in three dimensional space).This math-linguistic trait seems consistence also within the essential interrelation between integration & summation as notions as well as mathematical rules, whereas formally the integration sign is only vertically stretched (S) i.e. the first letter of (Summation)[8], also both of the idioms above express (bring parts together in a whole)*[14] i.e. unite or integrate all in ONE.
Hereafter, although single, double or triple definite integration will be helpful analytically in the process finding the irregular areas, but it becomes so difficult frequently in scientific applications to obtain it this way, for instance at the case of transcendental or complicated functions, look likes polynomials of high degree within decimal coefficients. Actually there are many methods to obtain an irregular area look likes the trapezoidal one based on Newton-Cotes integration formula, composite trapezoidal rule as its developed version, Simpson rule with its 1/3 & 3/8 developed versions, Riemann Sums, to remain some or say the most important ones, these methods based on taking the summation or say integration of successive divided subintervals as rectangles, trapezoids, or those under -might be well situated- polynomials, the details can be found in multiple resources[5,11,20,7,12], although these methods have their a huge scientific as well as educational importance, often it's difficult & time consuming to apply them in practice, for instance- one may need to divide a specific area into thousand intervals or say hundred ones to obtain a reasonable approximation , how is it going?!, at all built-in functions of Mat lab are technically powerful to find and portrait it saving time, effort and cost [17,19]. A significance essence to say is that the Mat lab functions int,trapez,quad,quadl are developed on the base of those methods & having additional themes that enable one to obtain the best approximation for an irregular area if not the exact one. Therefore one would create or develop certain models of Mat lab programmes in various styles to devote his/her model for the specific tasks related with irregular areas. This paper will include Mat lab programmes models for areas under algebraic, trigonometric, as those between power function &

square rooted one, or those among multi-power functions as well as a specific transcendental ones look likes Fresnels Integrals, while apart of those have its application in physics as well as civil engineering.

2 . Multi-area and Multicurve

Frequently in calculus referred to the area between two curves, although it's pure and fundamental, scientists especially in applied fields such as physics, chemistry, biology, many branches of engineering, as well as in various mathematics applications considered & found it so significance for their purposes[21,22,23], hence it seems reasonable to glance it from a little bit specific viewpoint in accordance with its technical value, taking in account the potential abilities of MATLAB as super technical language, it will be possible for one to construct mathematical models through by Mat lab programmes to conceptualize his aimed to be prolific project , hereafter let's firstly deal within a famed but interesting instance, it's the area beneath $y = \sqrt{x}$ & bounded by $y = x^2$.

```
>> % To portrait  a dual area
>> x = 0    :   0.01   :  1;       %  Increments for fill function
>> y = sqrt(x);    z = x.^2;        %  Functions definition
>>fill(y, x, 'g', z, x, 'g')        %  To draw & shade
```



Before going on in the details, let me quote David Tall [2] "concept image before concept definition", in accordance with this theoretical assignment, one may like to say model image before model processing, therefore the model above represents an illustration for the integration process in dual version, it's specifically comeliness & appropriate to portrait it in colors giving attention to how it may be appear in black & white version, this would be not only for its aesthetical value but for its cognitive effects, it makes one recognizes the area to be calculated, to investigate other areas at the plane in relation with that area, also allows someone to gauss its approximated value, or look at it from geometrical viewpoint, or compare the computational result against the analytical one, accept that it will be somewhat attractive for certain audience at the case of Data show or electronic board presentation. The most important issue is the programme created to achieve the task, the first condition one might taking it in consideration that the Mat lab programme has to be accordingly designed within the mathematical concept & aim to be implemented, then it has to be constructed by minimum lines as possible as someone can do it within essential comments, then it will be enough flexible & easily modified for his/her purpose. In relation with the programme instance, the plane is designed in accordance with simultaneous solution of these functions as equations to be precisely in its own square unit plane, for that it seems roughly different with the usual ones, with respect to the programme model it constructed by only a few instructions, the first one proposed a unit square axis and made increments for a Mat lab function FILL [15], interestingly it plays dual role, whereby it draws the defined functions as well as fills the enclosed area they created, the second defines the required functions as integrand, while the third has to co-figurate & shade double enclosed areas in fine green color, it's fit & elegance while by it seems in light grey at uncolored version. it may be more areas at the plane as the design and its content requires, we will see it at the coming model. Therefore one aims to numerically obtain the area enclosed between these curves, there are many methods as those mentioned at the preceding paragraph, but at this paper one would like to make use of built-in Mat lab functions[18], the following programmes models are prepared to do the task :

```
>> % To find the area between y = sqrt(x) & y = x^2
>> % through by MATLAB function int
>> syms  x                    %  Declares  the function's  variable
>> y = (sqrt(x)- x.^2);      %  Defines  the  integrand
>> A1 = int(y, 0, 1)          % Integrates via int function
```

```
A1 = 1/3

>> % To find the area between y = sqrt(x) & y = x^2
>> % through by MATLAB function trapz
>> x = 0 : 0.01 : 1;        %  Increments  for trapz function
>> y = (sqrt(x)- x.^2);   %  Defines  the integrand
>> A2 = trapz(x, y)       %  Integrates via trapz function

A2 = 0.3331

>> % To find the area between y = sqrt(x) & y = x^2
>> % through by MATLAB function quad
>> F =@(x)sqrt(x) - x.^2;    % Defines anonymous function as integrand
>> A3 = quad(F,0,1)          %  Integrates via Lobatto quadrature  function

A3 = 0.3333

>> % To find the area between y = sqrt(x) & y = x^2
>> % through by MATLAB function quadl
>> F =@(x)sqrt(x) - x.^2;   % Defines anonymous function as integrand
>> A4 = quadl(F,0,1)        % Integrates via  Adaptive Lobatto function

A4 = 0.3333
```

Therefore, the results are reasonable to a large extent, whereby int function offers the exact value, also quad & quadl functions do the same, moreover -say even- trapz function makes so interesting approximation, whereas its difference with the exact value equals 0.0002, i.e. so smaller than the tolerated error in numerical analysis which equals 0.005,thus no necessity to evaluate a relative error for it [4], Although it's reasonableness result but the sharp ends of the areas make it more satisfying, therefore the case slightly assigns the superiority of quad & quadl as Mat lab functions against the Mat lab trapz function. It stills significant to detail within this point as it will be seen at the ascending paragraph.
Therefore let's verify those results through by the stronger form of Fubini's theorem once in a while via Mat lab programme model, this theorem offers a mathematical rules to evaluate bounded areas at two dimensional space through by iterated integrals, unusually it proposes variable limits of integration as well as constants saying[22] :

Fubini's Theorem_ Let f(x, y) be continuous on a region R, then :

1. If R is defined by $a \le x \le b$, $g1(x) \le y \le g2(x)$, with g1 and g2 continuous on [a, b],then

$$\iint_R f(x, y)dA = \int_a^b \int_{g1(x)}^{g2(x)} f(x,y)dy\,dx$$

2. If R is defined by $c \le y \le d$, $h_1(y) \le x \le h_2(y)$, with h_1 and h_2 continuous on [c, d], then

$$\iint_R f(x, y)dA = \int_c^d \int_{h1(y)}^{h2(y)} f(x,y)dx\,dy$$

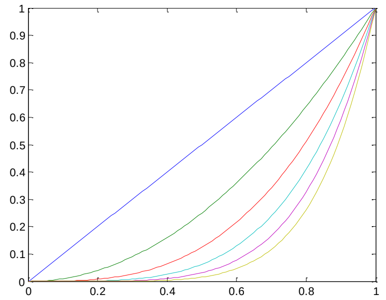
```
>> % Verifying via the stronger form of Fubini's Theorem
>> % To find the enclosed area in f = sqrt(x) - x.^2
>> syms x y                              % Declares the integrand variables
>> f = sqrt(x) - x.^2;                   % Defines the integrand function
>> A1 = int(int(f,x,0,1),y,0,1);         % Integrates accordingly to Fubini's Theorem1
>> A2 = int(int(f,y,0,1),x,0,1);         % Integrates accordingly to Fubini's Theorem2
```

>> disp('A1'), disp(A1),disp('A2'),disp(A2) % Displays the required areas

A1 = 1/3 , A2 = 1/3

The models results as well as the theorem rules precisely pass the verification process.
Moreover let's detail with unusual case, that is a case of multiple areas lay at the 1st quadrant of the two dimensional space, these areas are created by the function F(x) = x^n, $1 \leq n \leq 6$, $0 \leq x \leq 1$, $n \in N$ [6].

```
>> %To portrait multi-area at the 1st quadrant of the plane
>> x = 0 : 0.01 : 1;                % Increments for PLOT function
>> y = x; y1 = x.^2; y2 = x.^3;;    % Sets functions definition
> >y3 = x.^4; y4 = x.^5; y5 = x.^6  % Sets functions definition
>> plot(x,y,x,y1,x,y2,x,y3,x,y4,x,y5) % To draw the functions set
```



Except many areas not so easily visible at the programme-model output, there are twenty one areas under the consequent curves - whereas every curve in its own position- construct them with the successive ones in a form of decreasing arithmetical series within common difference equals one i.e. 6,5,4,3,2,1; here one embraced to avoid colors & shades , then it's better to leave them to a professional painter in abstract art, fortunately the natural sight made it colored & shaded by pure white color to be convenient for one purpose. Distinctly one will deal within five areas sequentially situated among the six successive curves as they appear at the model output. Then ahead to investigate these areas via models-generator of Mat lab programmes:

```
>> % To find multi-area via Mat lab int function
>> syms x                % Declares the multiple functions variable
>> for n = 1 : 5;        % Creates loop model for the integration process
>>F = (x.^n - x.^(n+1)); % Defines the function generator
>>A = int(F,0,1)         % Integrates via Mat lab int function
end                      % Closes the loop model
```

A = 1/6, 1/12, 1/20, 1/30, 1/42;

The programme model yields successive exact values of the required areas, factually the Mat lab int function is designed upon the base of the analytical methods for finding the definite integration for continuous functions over a closed interval, thus it yields the exact value , then one can check his calculations & other Mat lab ones against it.

```
>> %To find multi-area via Mat lab trapz function
>> for n = 1 : 5;          % Creates loop model for the integration process
>> x = 0 : 0.01 : 1;       % Increments for trapz- function
>> F = (x.^n - x.^(n+1)); % Defines the function generator
>>A = trapz(x, F)          % Integrates via Mat lab trapz rule
>>end                      % Closes the loop model
```

A = 0.1666, 0.0833, 0.0499, 0.0333, 0.0238

```
>> % To find multi-area via Mat lab quad function
>> for n = 1 : 5;           % Creates loop model for the integration process
>>F = @(x)x.^n - x.^(n+1); % Defines the function generator as anonymous function
>>A = quad(F,0,1)           % Integrates via Mat lab quad function
```

```
>>end                          %  Closes the loop model
```

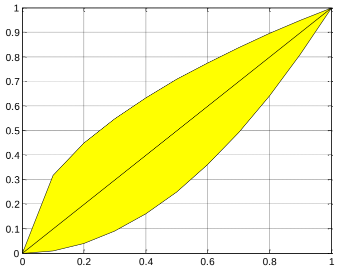
A = 0.1666, 0.0833, 0.0500, 0.0333, 0.0238

```
>> % To find multi-area via Mat lab quadl function
>> for n = 1 : 5;              % Creates loop model for the integration  process
>>F = @(x)x.^n - x.^(n+1);    %  Defines  the function generator as anonymous  function
>>A = quadl(F,0,1)             %  Integrates via Mat lab quadl function
   end                         %  Closes the loop models
```

A = 0.1666, 0.0833, 0.0500, 0.0333, 0.0238

The case of multi-area above also ensure the comparison conclusion mentioned about Mat lab functions and programmes models that finding enclosed areas, in spite of that some areas can be better found by Mat lab trapz than quad &quadl, precisely at the case of a numeric vector of data[16], the following model is constructed to clarify it in comparison with the first model, as we will see Mat lab functions quad, quadl will fail to find reasonable approximation for the requested area, while by good approximation will be obtained via trapz function, hereafter for a matter of consistency, one prefers to construct the first model we detailed with it through by numeric vectors, going on to portrait it by golden color & look for its enclosed area via trapz , quad & quadl to verify the validity of the Mat lab functions at the case of a numeric data.

```
>> % To portrait an area between two curves through by numeric data
>> % Defines  x- values as  numeric data
>> x = [0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1];
>>% Defines a function f as numeric data
>> f = [0   0.3162   0.4472   0.5477   0.6324   0.7071    0.7746   0.8366   0.8944  0.9486   1];
>>% Defines a function g  as numeric data
>> g = [0   0.01     0.04     0.09     0.16     0.25     0.36     0.49     0.64     0.81     1];
 >> fill(x, f, 'y', x, g, 'y')
```




```
>> % A comparison among quad, quadl & trapz Mat lab functions
>> x = [ 0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1];                    % Numeric vector for x
>> y = [ 0  0.30   0.40  0.45  0.47  0.45  0.41  0.34  0.25  0.13  0];   % Numeric vector for y
>> At = trapz(x, y);                        % Evaluates the enclosed area via trapz function
>> f = @(y)y;                               % Defines anonymous function for a vector y
>> A = quad(f,0,1);                         % Evaluates the enclosed area via Lobatto quad
>> Al = quadl(f,0,1);                        % Evaluates the enclosed area via quadl function
>> disp(At), disp(A), disp(Al)              % Displays integration's results
```

At = 0.32 , A = 0.5, Al = 0.5

Moreover, it seems proper to calculate the relative error for each of these approximations, to firstly find it at the case of trapz function, err = abs(0.33- 0.32)/0.33 = 0.030303, while at the case of quad &

Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online)
Vol.5, No.4, 2015

quadl, err = abs(0.33-0.5)/0.33 = 0.51515. Thus distinguish superiority is assigned to Mat lab trapz rule whence one aims to obtain reasonable approximation for a numeric data. Most precisely trapz(x, y) for area enclosed between curves and trapz(y) for single area beneath a curve.
The most significance point that these models have their theoretical as well as applied dimensions, theoretically, as it & will be seen in between of this paper, while applicably it seems difficult to find Mat lab programmes models for multiple areas, it may be find its limited applications recently or in the future as it the nature of mathematics science outputs.

3. Areas beneath transcendental functions

Transcendental, exponential, square rooted as well as those within natural logarithms functions are frequently occurring in practice[13,20,22], whereas extremely difficult or impossible to formally integrate them while they haven't anti-derivatives forms, the actual approach is to evaluate them by numerical methods aiming at reasonable approximations, but these methods as precedingly mentioned are also difficult & time consuming in practice, therefore scientists & engineers in many fields usually make use of computer programming to achieve their tasks, thus it seems convenient for one to investigate those areas can created by the complicated functions above through by Mat lab programmes models, at this paper one would like to introduce a slice of so interesting functions so called Fresnel's integrals via Mat lab programmes models, (They are two transcendental functions named after Augustine-Jean Fresnel that are used in optics, they arise in description of far field Fresnel diffraction phenomena, and are defined through the following integrals)[3,9]:

$$S(x) = \int_0^x \sin(t^2)\,dt \ , \ C(x) = \int_0^x \cos(t^2)\,dt$$

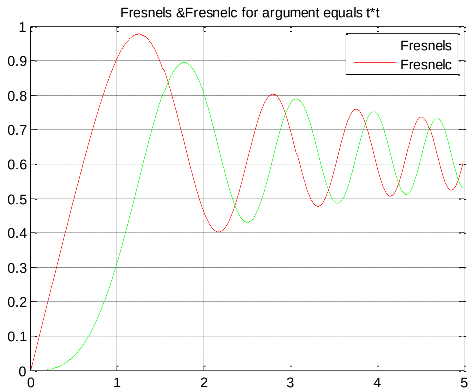
(Interestingly if you plot the points $S(x) = \int_0^x \sin(t^2)$ dx & $C(x) = \int_0^x \cos(t^2)\,dx$ for x∈ $[-\infty,\infty]$, you generate a beautiful twin spiral called Cornu's Spiral (or Euler's Spiral or clothed), this curve has the property that its curvature ρ is proportional to the total arc length from the origin, and it has practical use in designing curvature transition zones in railway & highway construction & even rollercoaster architecture)[9].Therefore it's at least not coincidentally to model a Mat lab programme that tackle this point, here below one is going to model these theoretical approaches through by technical programming constructions:

```
>>% To evaluate Fresnel's Integral  via Mat lab model
>> syms t              % Declares the functions variable
>> f = sin(t.^2);       % Defines Fresnels Integrand
>> g = cos(t.^2);      %  Defines Fresnelc Integrand
>> F = int(f,0,t);      % To symbolically evaluate Fresnels Integral
>> G = int(g,0,t);      % To symbolically evaluate Fresnelc Integral
>> disp (F),disp(G) %To display Fresnel's

F =  (2^(1/2)*pi^(1/2)*fresnels((2^(1/2)*t)/pi^(1/2)))/2

G =  (2^(1/2)*pi^(1/2)*fresnelc((2^(1/2)*t)/pi^(1/2)))/2

>> % To portrait Fresnel's Integrals
>> % Increments for Fresnel's
>> t = 0 : pi/314 : 5;
>>% To define Fresnel sine
>> F = (2^(1/2)*pi^(1/2)*fresnels((2^(1/2)*t)/pi^(1/2)))/2;
>>% To define Fresnel cosine
>> G = (2^(1/2)*pi^(1/2)*fresnelc((2^(1/2)*t)/pi^(1/2)))/2;
>>% Draws Fresnel's & tights the axes
>>plot(t, F, 'g' ,t ,G ,'r'),axis([0 5 0 1])
>>grid on  % Makes gridded plane
```



Fresnels elaboration is somewhat unusual, then one find himself embraced to firstly obtain the area beneath the transcendental functions symbolically, while by the result represents Fresnel's Integral as functions, then to make use of them dually ,once to portrait them through by Mat lab programme as the previous model did it, twice to evaluate their numerical value via Mat lab function(vpa) & obtain the aimed area. Here below a sub-model of Mat lab programme to do the calculation task:

```
>> % To  numerically calculate Fresnels & Fresnelc
>> syms t                       % Declares the functions variable
>> f = sin(t.^2);               % Note the argument value of Fresnels Integrand
>> g = cos(t.^2);               % Note the argument value  of Fresnelc Integrand
>> F = int( f, 0, 5);           % To evaluate Fresnels Integral
>> G = int( g, 0, 5);           % To evaluate Fresnelc Integral
>> disp (F), disp (G)           % Displays Fresnel's

 F = (2^(1/2)*pi^(1/2)*fresnels((5*2^(1/2))/pi^(1/2)))/2

 G = (2^(1/2)*pi^(1/2)*fresnelc((5*2^(1/2))/pi^(1/2)))/2

 >> vpa(F), vpa(G) : Evaluates Fresnel's

 F = 0.52791728116532241384461568718493

 G = 0.61146676639646261179401605046102
```

Here above the numerical values of Fresnels & Fresnelc Integrals while by they seemed dimensionless in theirs symbolic expression, the result is approximated to thirty two decimal places, but one can make use of only fifteen successive decimal digits, to reasonably compare it within those results of quad & quadl for fresnels and fresnelc, therefore one may verify these values via Mat lab functions quad, quadl as the model below do it cogently:

```
>> % To evaluate Fresnels & Fresnelc via quad & quadl
>> f = @(t)sin(t.^2);                        % Defines anonymous function for Fresnels
>> g = @(t)cos(t.^2);                        % Defines anonymous function for Fresnelc
>> F = quad(f,0,5);FL = quadl(f,0,5);        % Evaluates Fresnels via quad & quadl
>> G = quad(g,0,5);GL = quadl(g,0,5);        % Evaluate  Fresnelc via quad & quadl
>> disp(F),disp(FL),disp(G),disp(GL)
F=   0.527917555962510, FL = 0.527917281160141

G = 0.611466793041376, GL = 0.611466766398670
```

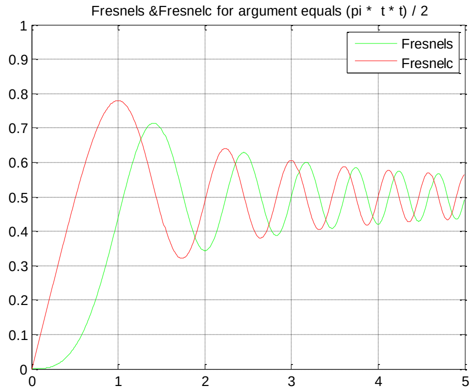
Therefore from the first glance, it seems that the obtained results are comparable, whereas quad result is correspondence within int approximation for Fresnels to six decimals places, while quadl result is similar within the same approximation to eleven decimal places, thus the last offers good approximation to a large extent. Then it remains to investigate that in comparison within int approximation for Fresnelc, while by quad result is correspondence to it for seven decimal places and quadl is similar to it for also eleven decimal digits.

If one like to replace the argument of the transcendental functions to be pi*t^2/2, the model will yields a little bit vertically & horizontally compressed portrait[10], this is an instance for what one can call it model flexibility, thus here below a modified model for Fresnel's functions:

```
 >> % Fresnel's  Integrals  within argument equals pi*t.^2/2
```

```
>> syms t                  % Declares the function variable
>> f = sin(pi*t.^2/2);     % Defines Fresnels integrand
>> g = cos(pi*t.^2/2);     % Defines Fresnelc  integrand
>> F = int(f,0,t);         % To evaluate Fresnels Integral
>> G = int(g,0,t);         % To evaluate Fresnelc Integral
>> disp (F),disp(G)        % Display Fresnel's Integral
```

F = fresnels(t), G = fresnelc(t)




```
>> % portrait Fresnel's Integrals
>> t = 0 : pi/314 : 5;                       % Increments for F & G
>> F = fresnels(t);                          % Defines fresnels function
>> G = fresnelc(t);                          % Defines fresnelc function
>> plot(t, F, 'g', t, G, 'r'),axis([0 5 0 1])  % Draws F,G & tights axes
>> grid on                                   % Makes gridded plane
```

```
>> % To  numerically calculate Fresnels & Fresnelc
>> syms t                  % Declares the functions variable
>> f = sin(pi*t.^2/2);     % Note the argument value of Fresnels Integrand
>> g = cos(pi*t.^2/2);     % Note the argument value of Fresnelc Integrand
>> F = int( f, 0, 5);      % To evaluate Fresnels
>> G = int( g, 0, 5);      % To evaluate Fresnelc
>> disp (F), disp (G)      % To display Fresnel's
>>vpa(F), vpa(G)           % To evaluate Fresnel's
```

F = fresnels(5) = 0.49919138191711688675192838046599

G =fresnelc(5) = 0.56363118870401223110210740441301

The results appear more readable and well ordered in this form than the normal output suite. Therefore it seems proper to verify the obtained results through by Mat lab functions quad, quadl , that's of course possible & coming in accordance with the aims of this work. The Mat lab model programme below has to achieve the task:

```
>>  >> % To evaluate Fresnels & Fresnelc via quad & quadl
>> f = @(t)sin(pi*t.^2/2);                    % Defines anonymous function f
>> g = @(t)cos(pi*t.^2/2);                    % Defines anonymous function g
>> F = quad(f,0,5);FL = quadl(f,0,5);         % Evaluates Fresnels via quad & quadl
>> G = quad(g,0,5);GL = quadl(g,0,5);         % Evaluates Fresnelc via quad & quadl
>> disp(F),disp(FL),disp(G),disp(GL)          % Display numerical values
```

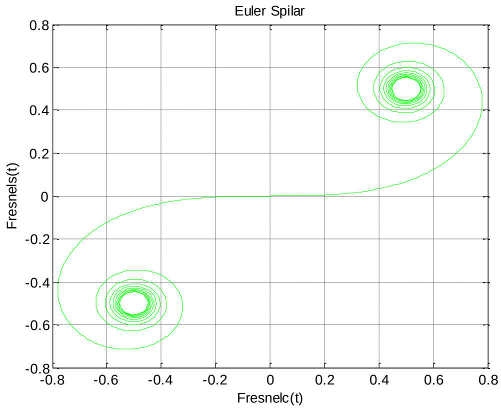
F = 0.499191400077301, FL = 0.499191381818519

G = 0.563631503435295, GL = 0.563631188745787

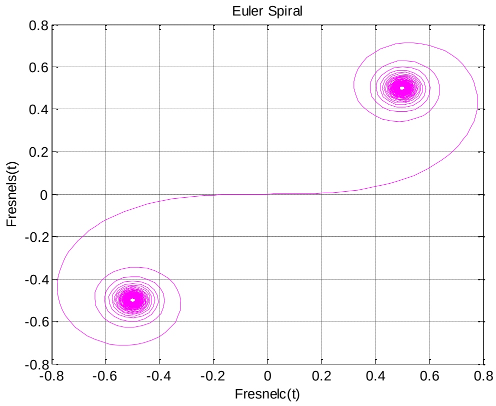
Hereafter one will verify by passing through the same comparison process, quad result is correspondence with int approximation for Fresnels to 6-decimals while by quadl is similar to 9-decimals with the same approximation, then if one compares quad & quadl results for Fresnelc by same way, he will find that the first is correspondence to the int approximation to 6-decimals while by the second is similar with the same approximation to 10-decimals, from all one can conclude that the best approximation for Fresnels as well as Fresnelc can be obtained via quadl.

After all, it seems proper to detail with Euler Spiral while it is strongly related with Fresnels & Fresnelc Integrals as one can see at the following models:

```
>> % To portrait  Euler spiral / green model
>> t = -2*pi: 0.01: 2*pi;  % Increments for Fresnels &Fresnelc
>> x = fresnelc(t);         % Defines x-coordinate for Euler spiral
>> y = fresnels(t);         % Defines y-coordinate for Euler spiral
>> w = vpa(x);              % Determines value of x-coordinate
>> u = vpa(y);              % Determines value of y-coordinate
>> plot(w, u, 'g')          % Draws and colors the spiral path
>> grid on                  % Does  gridded plane
```




```
>>%To portrait Euler Spiral/ magnet model
>> t = -2*pi: 0.01: 2*pi;  % Increments for Fresnels & Fresnelc
>> x = fresnelc(pi*t);      % Defines x-coordinate for Euler Spiral
>> y = fresnels(pi*t);      % Defines y-coordinate for Euler spiral
>> plot(x, y, 'm')          % Draws & colors the spiral path
>> grid on                  % Does gridded plane
```



Euler Spiral has the property as that its curvature at any point is proportional to the distance along the spiral, this makes it useful as transition curve in highway & railway[9], mathematically interesting that every point of Euler Spiral is constructed by two functions , they are Fresnelc(t) as x-coordinate for any of its points & Fresnels(t) as y-coordinate for that point, this isn't only to emphasize what it's assigned at the programme model as well as its output, but to significantly assign that spiral path can be changed in accordance to the argument value of Fresnel's integrand, also if one makes use of vpa Mat lab function to evaluate Fresnel's numerically, it is observable that the final portrait somewhat differs from that without such additional instruction, the matter which means that vpa ignore many values at the inner circled petals of the Spiral. Therefore, it appeared that these models are not only productive in their own selves, but the most significance matter is that they are applicable as models generator for multiple mathematical & scientific tasks.

REFRENCES

1. Bruce Simmons, Math words Terms & Formulas from Algebra, 2014
2. David Tall & VINNER Shalom , Concept Image & concept definition, ESM 12, 1980
3. Eugene Hecht, Optics, http://en. wikipedia.org, modified-2015
4. Francis Scheid, Numerical Analysis, 1981
5. GERALD.H, ALFRED S.P, STEPHEN E.M, Computers in Mathematics, 1983
6. GEMIGNANI, Elementary Topology, 1972
7. H. K. DASS , Mathematical Physics, 2011
8. http://en.wikipedia.org/wiki/ History of Calculus, 2010
9. http://en.wikipedia.org/wiki/, How to evaluate Fresnel's Integrals-2015.
10. http://en.wikipedia.org/wiki/Fresnel_ Integral, modified-2015.
11. Lila F . Roberts, Area Between Two Curves as Limits of Riemann Sums, IMS,2004
12. Murray BOUME , Riemann Sums, square Circle Z, The int math blog,2010
13. N.p.Bali, Engineering Mathematics, 2011
14. OXFORD Advanced Learner's Dictionary, A S Hornby,2003
15. PROFESSORS TEAM, Math Works. Inc, MAT LAB 2010, Filled 2-D polygons, 2010
16. PROFESSORS TEAM, Math Works. Inc, MAT LAB 2014,Numerical Integration, 2014

17. PROFESSORS TEAM, The Language of Technical Computing 2014a, MAT LAB 2014
18. PROFESSORS TEAM, Math Works inc, MAT LAB 2010b,Integration Commands,2010
19. PROFESSORS TEAM, The Language of Technical Computing2010b, MAT LAB 2010
20. Steven Chapra, Applied Numerical Method with MATLAB, 2012
21. Stomy Attawy , Mat lab Programming & Problem Solving, 2011
22. THOMAS.JR, M.D.WIER, J.R. HASS, Thomas' Calculus, 2014
23. Winfried Just , Area Below the Graph of a Function, Ohio University,2004