by Lukas Setiawan

email: lukassetiawan@yahoo.com

(Mar 7, 2025)

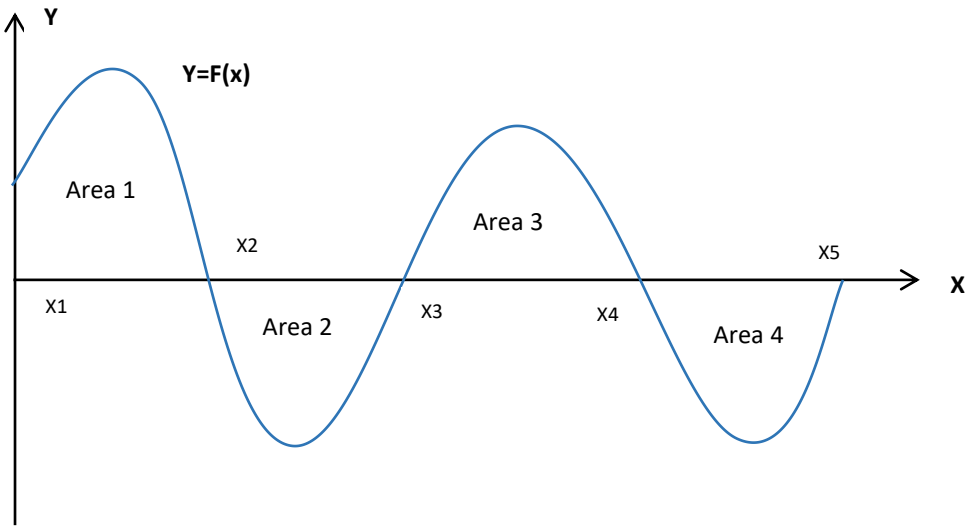
MULTI-AREA CALCULATION USING EXTENDED FALSE POSITION METHOD AND EXTENDED COMPOSITE SIMPSON'S 1/3 RULE

To simultaneously calculate the multi-area under the curve and between two curves of the function on the given interval.

Illustration

1. Singles Curve

The area under single curve(function).

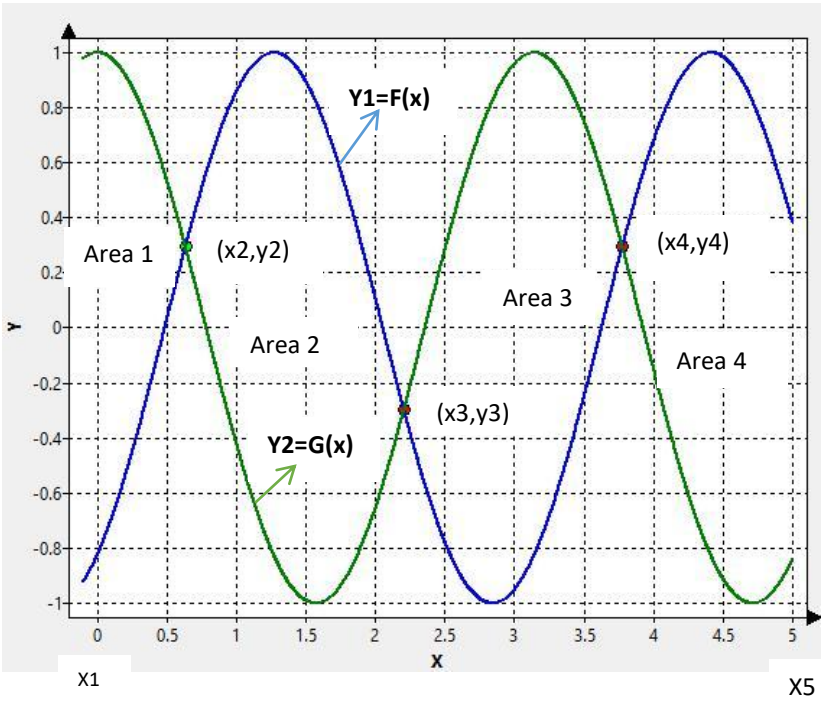


- Area 1 under the Graph Y=F(x) where x1 is lower limit and x2 is upper limit.
- Area 2 under the Graph Y=F(x) where x2 is lower limit and x3 is upper limit.
- Area 3 under the Graph Y=F(x) where x3 is lower limit and x4 is upper limit.
- Area 4 under the Graph Y=F(x) where x4 is lower limit and x5 is upper limit.
- All area bounded by x-axis.

- Total Area = Area 1 + Area 2 + Area 3 + Area 4

2. Double Curves

The possibilities that could be happen when we want to know the area between two curves(function).



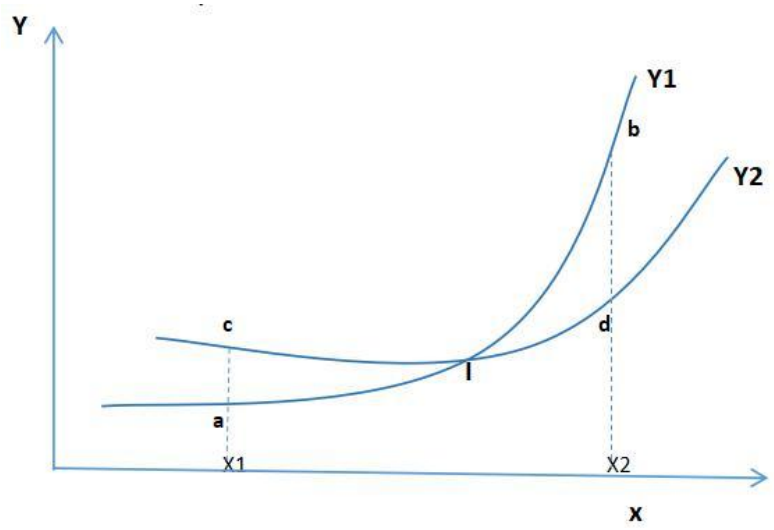
Graph 1: Y1=F(x) is blue.

Graph 2: Y2=G(x) is green.

- All the area bounded by Graph Y1 and Y2.
- Area 1 with the boundary x1 is lower limit and x2 is upper limit.
- Area 2 with the boundary x2 is lower limit and x3 is upper limit.
- Area 3 with the boundary x3 is lower limit and x4 is upper limit.
- Area 4 with the boundary x4 is lower limit and x5 is upper limit.
- Total Area = Area 1 + Area 2 + Area 3 + Area 4

Illustration

If two graphs intersected, there are two possibility.



Here is the Idea(Algorithm for Single Curve)

- Input function Y1=F(x), boundary where multi-area placed, tolerances(delta,epsilon), gamma,max iteration number of subinterval, number of smoothness to create graph and domain graph [x1,x2].
- In multiple area got lower limit & upper limit for each area.
- Boundary of the area could be root or not root.
- To find root, we are starting from left boundary(position x1), then x1+gamma, to know value of y1=f(x1) and y2=f(x1+gamma), with constant gamma at each step. The use of gamma as “blind scanner” of two position y1 and y2, comparing with the classical False Position method as bracketing method, which means position of the root already known in the interior point between interval(bracket). So this method call Extended False Position.
- We set gamma=0.0097(unic number, close to 1% or 0.01) as default, that value to avoid gamma equal root(if gamma=root then we could be missed the root),

what we needed is boundary where the root is placed , then find the root with False Position method.

- We can increase or decrease gamma as input variable.
- If y1 & y2 got opposite sign(y1 positive and y2 negative) or (y1 negative and y2 positive) or in code as (y1*y2<0), means the graph crossing x-axis or have a root.
- If not then increase position x by gamma with different gamma at each process till y1 & y2 got opposite sign, if this condition fulfill, we got boundary for inding Root.
- After got boundary then find the root using False position method that we can set tolerances as we liked.
- After one root found, then repeat the procedure till got all the roots in the interval (from left boundary to right boundary).
- After we got all roots as boundary for each area then we find the area using Composite Simpson's 1/3 rule.
- Finally we got each Area and Total area.
- Then viewing boundary points & area visualization on the graph.
- Done.

Here is the Idea(Algorithm for Double Curves)

- Just like Single curve, but got different in condition to found intersection point as boundary to find area.
- At end points(left endpoint & right endpoint) the boundary of the area could be intersection point or not, but at interior point, it must be intersection point. If got no intersection point(only got one area bounded by left boundary & right boundary).
- If intersection point, then we found it first with the condition as follow(look illustration above where two graphs intersected):

  There are two possibilities conditions:

  1. F1(x1) < F2(x1) and F1(x2) > F2(x2) or
  2. F1(x1) > F2(x1) and F1(x2) < F2(x2)
- When two curves intersected, Y1=Y2 so Y1-Y2=0. It's a finding root problem.

The method is the same using False position method.

- After got all boundary of all area then we do the same process like under the Single Curve, where to find area using Composite Simpson's 1/3 rule.
- Last step is viewing boundary points & two graph as area visualization on the graph.
- Done.

False Position formula :

$$c_i = b_i - \frac{f(b_i)(b_i - a_i)}{f(b_i) - f(a_i)}$$

for i = 0,1,2,…..,n

…look next page

Program 2.3 (False Position or Regula Falsi Method). To approximate a root of the equation $f(x) = 0$ in the interval $[a, b]$. Proceed with the method only if $f(x)$ is continuous and $f(a)$ and $f(b)$ have opposite signs.

```
function [c,err,yc]=regula(f,a,b,delta,epsilon,max1)
%Input  - f is the function input as a string 'f'
%       - a and b are the left and right end points
%       - delta is the tolerance for the zero
%       - epsilon is the tolerance for the value of f at the zero
%       - max1 is the maximum number of iterations
%Output - c is the zero
%       - yc=f(c)
%       - err is the error estimate for c
ya=feval(f,a);
yb=feval(f,b);
if ya*yb>0
   disp('Note: f(a)*f(b)>0'),
   break,
end
for k=1:max1
   dx=yb*(b-a)/(yb-ya);
   c=b-dx;
   ac=c-a;
   yc=feval(f,c);
   if yc==0,break;
   elseif yb*yc>0
      b=c;
      yb=yc;
   else
      a=c;
      ya=yc;
   end
   dx=min(abs(dx),ac);
   if abs(dx)<delta,break,end
   if abs(yc)<epsilon,break,end
end
c;
err=abs(b-a)/2;
yc=feval(f,c);
```

(The algorithm of the False Position Method, took from the book “Numerical method using Matlab “ third edition 1999 by John H. Mathews and Kurtis D. Fink page 60).

Composite Simpson’s 1/3 rule formula :

$$\int_a^b f(x)dx \approx \frac{h}{3}\left[f(x_0) + 2\sum_{k=1}^{\frac{M}{2}-1} f(x_{2k}) + 4\sum_{k=1}^{\frac{M}{2}} f(x_{2k-1}) + f(x_M)\right]$$

h=(b-a)/M ; x_0 =a ; x_M= b ; x_k=a+k h ; for k=0,1,2…n

The formula took from website wikipedia.org at :

https://en.wikipedia.org/wiki/Simpson%27s_rule

With the file MultiArea.pdf in the same folder included app MultiArea.exe and source code that implemented using Lazarus IDE 3.2 and added component ArtFormula as Math Parser.

Links of mine:

- Visit web: https://nix97.github.io/numericalmethods
- Facebook search: Metode Numerik-Plus Programnya: https://web.facebook.com/profile.php?id=100069640586760
- My Work :
  - On bitbucket.org: https://bitbucket.org/nixz97/nix/downloads/
  - Other Repositories on GitHub: https://github.com/nix97