# Half–Function Value Quadratic Integration

By

Namir Clement Shammas

Introduction
--------------

Numerical analysis offers a wealth of algorithms to perform numerical integration. Among the simplest algorithms are the trapezoidal integration and Simpson's Rule. The latter has two versions, namely, the 1/3 rule and the 3/8 rule. The firrst rule is based on a quadratic polynomial that integrates a function f(x) from a to b. The basic Simpson's Rule formula is:

$$\int_{a}^{b} f(x)dx = \frac{(b-a)}{3} * (f(a) + 4 * f\left(\frac{a+b}{2}\right) + f(b))$$
(1)

Equation 1 samples the function f(x) at a, b, and the midpoint between them. Simpson's Rule is very simple to implement and yields reasonably good answers. The competition for this algorithm includes Romberg's method and the Gauss– Legendre quadrature that perform numerical integration for finite limits.

This paper asks the question, "What is a numerical integration algorithm that implements an integral by replacing f((a+b)/2) with f(m) = (f(a) + f(b)) / 2. In other words, the new method locates a value of x=m such that the function value at m is the average between f(a) and f(b). The logic behind this integration algorithm is to integrate an interpolated quadratic polynomial based on a, m, and b. I will call this new algorithm the Half–Function Value Quadratic Integration, or HFVQI algorithm for short.

#### The HFVQI Algorithm

Let's derive the equations for the HFVQI algorithm. The first step is to calculate c as the midpoint between the integral limits a and b:

$$c = (a + b) / 2$$
 (2)

We then calculate f $\mathbb{O}$ . Next, we calculate m, such that f(m) = (f(a) + f(b)) / 2, we perform an inverse Lagrangian interpolation using (a, f(a), (c, f(c), and (b, f(b)):

$$m = a * [(f(m) - f(c))*(f(m) - f(b))]/[(f(a) - f(c))*(f(a) - f(b))] + c * [(f(m) - f(a)) * (f(m) - f(b))]/[(f(c) - f(a))v(f(c) - f(b))] + b * [(f(m) - f(a)) * (f(m) - f(b))]/[(f(b) - f(a))v(f(b) - f(c))]$$
(3)

After calculating m, we calculate the actual value of f(m) and proceed to obtain the quadratic polynomial that we will use for the integration. Again, using Lagrangian interpolation we get:

$$P(x) = f(a)*(x-m)*(x-b)/[(a-m)*(a-b)] + f(m)*(x-a)*(x-b)/[(m-a)*(m-b)] + f(b)*(x-a)*(x-m)/[(b-a)*(b-m)]$$
(4)

We define the following factors:

$$Ca = f(a)/[(a-m)*(a-b)]$$
 (5a)

$$Cm = f(m)/[(m-a)*(m-b)]$$
 (5b)

$$Cb = f(b)/[(b-a)*(b-m)]$$
 (5c)

And,

$$K1 = (Ca + Cb + Cb) \tag{6a}$$

$$K2 = (Ca^{*}(m+b) + Cm^{*}(a+b) + Cb^{*}(m+a))$$
(6b)

$$K3 = (Ca*mb + Cm*ab + Cb*m*a)$$
 (6c)

Using equations 5a, 5b, 5c, 6a, 6b, and 6c, we transform equation 1 into a simpler form:

$$P(x) = K1^*x^2 - K2^*x + K3$$
(7)

Integrating equation 7 yields the following equation:

$$\int f(x)dx = K1^*X^3/3 - K2^*x^2/2 + K3^*x + Konst$$
(8)

Applying the integral limits (a, b) to equation 8, we get

$$\int_{a}^{b} f(x)dx = K1^{*}(b^{3} - a^{3})/3 - K2^{*}(b^{2} - a^{2})/2 + K3^{*}(b - a)$$
  
= (b - a)\*[K1/3^{\*}(b^{2} + a^{\*}b + a^{2}) - K2/2^{\*}(b + a) + K3] (9)

Equation 9 is the formula we use to perform numerical integration for the HFVQI algorithm. For the sake of increased accuracy, we can apply both Simpson's Rule and equation 9 to subintervals of (a, b). In fact, we can use subintervals with any

numerical integration method with defined limits. In the case of Simpson's Rule, we call the interval division method the Composite Simpson's Rule.

#### The Excel VBA Code for the Integration Algorithms

```
Listing 1 shows a VBA function that implements Simpson's Rule.
```

```
Function SimpsonArea (ByVal A As Double, ByVal B As Double, ByVal
N As Long)
  Dim Sum As Double
  Dim h As Double, I As Long, k As Long
  If (N \mod 2) = 0 Then N = N + 1
  h = (B - A) / N
  Sum = f(A) + f(B)
  k = 4
  For I = 1 To N - 1
    Sum = Sum + k * f(A + I * h)
    k = 6 - k
  Next
  SimpsonArea = Sum * h / 3
End Function
Listing 1. The Simpson's Rule function.
Function HVFQI Area (ByVal A As Double, ByVal B As Double, ByVal
N As Long)
 Dim Sum As Double
  Dim Fa As Double, Fb As Double, Fc As Double, Fm As Double
  Dim Ca As Double, Cb As Double, Cm As Double
  Dim K1 As Double, K2 As Double, K3 As Double
  Dim Incr As Double, m As Double, C As Double
  Dim I As Long
  Incr = (B - A) / N
  Sum = 0
  For I = 1 To N
    B = A + Incr
    C = (A + B) / 2
    Fa = f(A)
    Fb = f(B)
    Fc = f(C)
    Fm = (Fa + Fb) / 2
```

```
m = A * ((Fm - Fc) * (Fm - Fb)) / ((Fa - Fc) * (Fa - Fb)) +
        C * ((Fm - Fa) * (Fm - Fb)) / ((Fc - Fa) * (Fc - Fb)) +
        B * ((Fm - Fa) * (Fm - Fc)) / ((Fb - Fa) * (Fb - Fc))
    Fm = f(m)
    Ca = Fa / (A - m) / (A - B)
    Cm = Fm / (m - A) / (m - B)
    Cb = Fb / (B - A) / (B - m)
    K1 = (Ca + Cm + Cb)
    K2 = (Ca * (m + B) + Cm * (A + B) + Cb * (m + A))
    K3 = (Ca * m * B + Cm * A * B + Cb * m * A)
    Sum = Sum + Incr * (K1 / 3 * (B ^ 2 + A * B + A ^ 2) - K2 /
2 * (B + A) + K3)
   A = B
 Next I
  HVFQI Area = Sum
End Function
Listing 2. The HVFQI algorithm function
```

Listing 2 shows a VBA function that implements of the HVFQI algorithm.

Testing the Integral of f(x)=1/x

A simple integral is f(x)=1/x which gives an integral function I(x)=ln(x)+C. Table 1 shows the results of integrating f(x) between 1 and 5 for N = 5.

	HFVQI	Simpson 1	Simpson 2
Ν	5	5	88
Integral	1.606402047	1.558430415	1.606428327
Actual Integral	1.609437912	1.609437912	1.609437912
%Err	0.188628905	-3.169274014	0.186996049
%Err	0.188628905	-3.169274014	0.18699604

Table 1. Results using N = 5.

The data under the HFVQI column show the calculated integral, actual integral, and percent error. The third column shows similar results for Simpson's rule. Notice that the error in the latter method is over 20 time greater than the percent error for the HFVQI algorithm. The last column shows the high number of N used with Simpson's rule to achieve a comparable percent error with the HFVQI algorithm.

Version 0.9.0

The ratio of the two percent errors is over 15! I will be using Table 1 as template for other similar calculations in this paper. I will also refer to N in the second and fourth columns as N HVFQI and N Sipmson, respectively.

	HFVQI	Simpson 1	Simpson 2
Ν	10	10	1300
Integral	1.609232799	1.584774355	1.609232879
Actual Integral	1.609437912	1.609437912	1.609437912
%Err	0.01274443	-1.532432994	0.012739451
	Table 2 Res	ults using $N - 10$	

Table 2 shows the results of integrating f(x) between 1 and 5 for N = 10.

Table 2. Results using N = 10.

In Table 2, notice that the error in the latter method is 3 orders of magnitude greater than the percent error for the HFVQI algorithm. The last column shows the high number of N used with Simpson's rule to achieve a comparable percent error with the HFVQI algorithm. The ratio of the two percent errors is over 100!

Tahla	3 ch	owe	tha	raculta	$\mathbf{of}$	integra	ting	$f(\mathbf{v})$	hatwaan	1 4	and 5	for N	J — 1	15
Table	5 SH	.Uw 5	uic	resuits	01	megra	ung	$I(\Lambda)$	UCIWCCII	10	inu J	101 1	N — .	15.

	HFVQI	Simpson 1	Simpson 2
Ν	15	15	6500
Integral	1.609396587	1.591332647	1.609396891
Actual Integral	1.609437912	1.609437912	1.609437912
%Err	0.002567686	-1.12494341	0.002548831

Table 3. Results using N = 15.

In Table 3, notice that the error in the latter method is 3 orders of magnitude greater than the percent error for the HFVQI algorithm. The last column shows the high number of N used with Simpson's rule to achieve a comparable percent error with the HFVQI algorithm. The ratio of the two percent errors is over 430!

Table 4 shows the results of integrating f(x) between 1 and 5 for N = 20.

	HFVQI	Simpson 1	Simpson 2
Ν	20	20	20000
Integral	1.609424737	1.596538305	1.60942458
Actual Integral	1.609437912	1.609437912	1.609437912
%Err	0.000818607	-0.801497657	0.000828422
	<b>—</b> 11 ( <b>)</b> 1		

Table 4. Results using N = 20.

In Table 4, notice that the error in the latter method is 3 orders of magnitude greater than the percent error for the HFVQI algorithm. The last column shows the high number of N used with Simpson's rule to achieve a comparable percent error with the HFVQI algorithm. The ratio of the two percent errors is about 1000!

Notice that the percent error in Simpson's rule that uses the same number of divisions, remains 3 orders of magnitudes compared with that of the HFVQI algorithm. By contrast, the number of divisions needed by Simpson's Rule to match the percent error of the HFVQI algorithm has increased in whopping paces!

	HFVQI	Simpson 1	Simpson 2
Ν	25	25	50000
Integral	1.609432496	1.598621273	1.609432579
Actual Integral	1.609437912	1.609437912	1.609437912
%Err	0.000336524	-0.672075586	0.000331375
	Tuble 5 Deces	ta maina NI 25	

Table 5 shows the results of integrating f(x) between 1 and 5 for N = 25.

*Table 5. Results using* N = 25*.* 

In Table 5, notice that the error in the latter method is 3 orders of magnitude greater than the percent error for the HFVQI algorithm. The last column shows the high number of N used with Simpson's rule to achieve a comparable percent error with the HFVQI algorithm. The ratio of the two percent errors is about 2000!

Table 6 shows the results of integrating f(x) between 1 and 5 for N = 30.

	HFVQI	Simpson 1	Simpson 2
Ν	30	30	102000
Integral	1.609435295	1.600733686	1.609435298
Actual Integral	1.609437912	1.609437912	1.609437912
%Err	0.000162617	-0.540823974	0.00016244
		· · N 20	

Table 6. Results using N = 30.

In Table 6, notice that the error in the latter method is 3 orders of magnitude greater than the percent error for the HFVQI algorithm. The last column shows the high number of N used with Simpson's rule to achieve a comparable percent error with the HFVQI algorithm. The ratio of the two percent errors is about 3400!

	HFVQI	Simpson 1	Simpson 2		
Ν	35	35	190000		
Integral	1.609436498	1.601737323	1.609436509		
Actual Integral	1.609437912	1.609437912	1.609437912		
%Err	8.78848E-05	-0.478464546	8.72046E05		
Table 7. Results using $N = 35$ .					

Table 7 shows the results of integrating f(x) between 1 and 5 for N = 35.

In Table 7, notice that the error in the latter method is 4 orders of magnitude greater than the percent error for the HFVQI algorithm. The last column shows the high number of N used with Simpson's rule to achieve a comparable percent error with the HFVQI algorithm. The ratio of the two percent errors is about 5400!

Figure 1 shows the graph for N for Simpson's Rule vs N for the HVFQI algorithm, where the first N allows Simpson's Rule to generate comparable Error to the HVFQI algorithm.



Figure 1. Plot of N\_Simpson vs N\_HFVQI.

Figure 1 shows that here is a fairly exponential relationship between the values of N, for Simpson's Rule, and N, for the HFVQI algorithm. Table 8 shows the results of the regression of the following log-linear model:

Copyright © 2018 by Namir Clement Shammas

7

$\ln(N_{Simp}) = 4.434289 + 0.241413 * N_{HFVQI}$	(10)
---	------

SUMMARY OUTPU	JT				
Regression S	tatistics				
Multiple R	0.964392				
R Square	0.930052				
Adjusted R					
Square	0.916062				
Standard Error	0.783357				
Observations	7				
ANOVA					
					Significance
	df	SS	MS	F	F
Regression	1	40.79603	40.79603	66.4812	0.000451
Residual	5	3.068238	0.613648		
Total	6	43.86427			
		Standard			
	Coefficients	Error	t Stat	P–value	
Intercept	4.434289	0.662057	6.697743	0.001122	
Slope	0.241413	0.029608	8.153601	0.000451	

Table 8. The results of the regression of the model in equation 10.

We can improve the fit between N (Simpson) and N( HFVQI) using the following log-quadratic model:

 $\ln(N_{Simp}) = 2.255521 + 0.531915 * N_{HFVQI} - 0.00726 * N_{HFVQI^{2}}$ (11)

The quadratic term in equation 11 performs a correction needed to improve the curve fit. Table 9 shows the results of using the model in equation 11.

SUMMARY OUTPUT				
----------------	--	--	--	--

Regression S	tatistics				
Multiple R	0.996584				
R Square	0.99318				
Adjusted R					
Square	0.989771				
Standard Error	0.273467				
Observations	7				
ANOVA					
					Significance
	df	SS	MS	F	F
Regression	2	43.56514	21.78257	291.272	4.65E-05
Residual	4	0.299137	0.074784		
Total	6	43.86427			
		Standard			
	Coefficients	Error	t Stat	P–value	
Intercept	2.255521	0.426168	5.292565	0.006118	
Slope1	0.531915	0.048846	10.88953	0.000404	
Slope2	-0.00726	0.001194	-6.08505	0.003687	

Table 9. The results of the regression of the model in equation 11.

Figure 2 shows the plot of equation 11.



Figure 2. Plot model in equation 11

### Testing other Integrals

Let's look at the test case of integration of  $f(x)=\exp(x)$  with an integral  $I(x)=\exp(x)+C$ . Table 10 shows the results of integrating f(x) between 0 and 1 for N = 10.

	HFVQI	Simpson 1	Simpson 2
Ν	10	10	1700000
Integral	1.718281291	1.639651917	1.718281295
Actual Integral	1.718281828	1.718281828	1.718281828
%Err	3.12662E-05	-4.57607772	3.10191E-05
<b>T</b> 11 10 D			

Table 10. Results for f(x)=exp(x) integral from 0 to 1, using N = 10.

The results of Table 10 shows how superior the HFVQI algorithm is to Simpson's Rule. The ratio between N\_HFVQI and N\_Simpson is over 150,000! These results show that HFVQI excels by leaps and bounds over the legacy Simpson's Rule for monotonic functions that exhibit a quick change in function value.

Let's look at the test case of integration of  $f(x)=\sqrt{x}$  with an integral  $I(x)-(2/3)*x^{3/2}$ . Table 11 shows the results of integrating f(x) between 1 and 2 for N = 10.

	HFVQI	Simpson 1	Simpson 2
Ν	10	10	1000000
Integral	1.218951427	1.176583269	1.218950945

Half-Function Value Quadratic Integration

Actual Integral	1.218951416	1.218951416	1.218951416
%Err	-8.76989E-07	-3.475786359	-3.86729E-05
T 11 11 D			· N 10

Table 11. Results for  $f(x) = \sqrt{x}$  integral from 1 to 2, using N = 10.

The results of Table 11 shows how better the HFVQI algorithm is to Simpson's Rule. Even with N = 1 million, the Simpson's Rule produces a percent error that is 2 orders of magnitude higher than the HFVQI algorithm.

Let's examine the test case of integration of  $f(x)=x^3$  with an integral  $I(X)=x^4/4+C$ . Table 12 shows the results of integrating f(x) between 0 and 2 for N = 10.

	HFVQI	Simpson 1	Simpson 2
Ν	10	10	14000
Integral	3.999615714	3.581176149	3.999619116
Actual Integral	4	4	4
%Err	0.009607142	-10.47059627	-0.009522109
$T_{n} = \frac{1}{2} \frac{1}$			

Table 12. Results for  $f(x) = x^3$  integral from 0 to 2, using N = 10.

The results of Table 12 shows how superior the HFVQI algorithm is to Simpson's Rule. The ratio of N\_HFVQI to N\_SIMPSON that achieves comparable percent error is 1400. The percent error of the HFVQI algorithm is 3 orders of magnitude better than the percent error of the Simpson's Rule for N=10.

Let's consider the test case of integration of  $f(x)=\sinh(x)$  with an integral  $I(x)=\cosh(x)+C$ . Table 13 shows the results of integrating f(x) between 1 and 2 for N = 10.

	HFVQI	Simpson 1	Simpson 2
Ν	10	10	500000
Integral	2.219114363	2.114389558	2.219112638
Actual Integral	2.219115056	2.219115056	2.219115056
%Err	3.12606E-05	-4.719245971	-0.000108958

Table 13. Results for  $f(x) = \sinh(x)$  integral from 1 to 2, using N = 10.

The results of Table 12shows that the HFVQI algorithm is superior is to Simpson's Rule. The percent error of the HFVQI algorithm is 5 orders of magnitude better than the percent error of the Simpson's Rule for N=10.

#### Conclusion

The new Half–Function Value Quadratic Integration algorithm is able to compete very effectively against the legacy Simpson's Rule. While the new algorithm required more calculations per integration step, the calculations yield more accurate results. In other words, one integration step is worth tens of thousands small Simpson Rule steps! Such a superiority is true for monotonic function that have a high curvature (i.e. high second derivative).

## Document History

Date	Version	Comment
June 3, 2018	0.9.0	Initial release.