Quadratic Lagrangian Integration

By Namir Shammas

Introduction

Numerical Analysis offers various algorithms for numerical integration. Many such algorithms are based on implicit polynomial interpolation. This article looks at using quadratic Lagrangian polynomials to perform numerical integration. The advantage of the Lagrangian integration is that the sampled points need not be at regular intervals for the independent variable x.

The Algorithm

Given three points (x_0, y_0), (x_1, y_1), and (x_2, y_2) you can calculate the following quadratic Lagrangian polynomial:

$y = A x^2 + B x + C$ (1)

Where A, B, and C are calculated using the following set of equations based on the given three points:

$\alpha_0 = y_0 / ((x_0 - x_1)(x_0 - x_2))$ (2)
$\alpha_1 = y_1 / ((x_1 - x_0)(x_1 - x_2))$ (3)
$\alpha_2 = y_2 / ((x_2 - x_0)(x_2 - x_1))$ (4)
$C = \alpha_0 x_1 x_2 + \alpha_1 x_0 x_2 + \alpha_2 x_0 x_1$ (5)
$B = -[\alpha_0 (x_1 + x_2) + \alpha_1 (x_0 + x_2) + \alpha_2 (x_0 + x_1)]$ (6)
$A = \alpha_0 + \alpha_1 + \alpha_2$ (7)

The integral between points (x_0, y_0) and (x_2, y_2) is:

$\int_{x0}^{x2} f(x)dx = A/3\,(x_2^3 - x_0^3) + B/2(x_2^2 - x_0^2) + C(x_2 - x_0)$ (8)

Equation 8 gives the basic rule for the quadratic Lagrangian integration. To apply a chained integration for more than three points, you have to include an even number of additional points.

If the values of the variable x occur in an equal interval, h, then equations 2, 3, and 4 become:

$\alpha_0 = 2 y_0 / h^2$ (9)
$\alpha_1 = -4 y_1 / h^2$ (10)
$\alpha_2 = 2 y_2 / h^2$ (11)

Equations 5, 6, and 7 remain basically the same. Comparing equation 5 with Simpson rule, which is:

$\int_{a}^{b} f(x)dx = (b-a)/6\,[f(a) + f((a+b)/2) + f(b)]$ (12)

Or,

$\int_{x0}^{x0+h} f(x)dx = h/6\,[f(x_0) + f(x_0+h/2) + f(x_0 + h)]$ (13)

It is interesting to point out that the quadratic Lagrangian integration and Simpson's rule yield the same results, using equidistant x values, since both are based on quadratic interpolation. The differences are:

- The quadratic Lagrangian integration uses explicit interpolation, while Simpson's rule uses implicit interpolation.
- The computation effort for Simpson's rule is eloquently less than that of the quadratic Lagrangian integration, when both methods use equidistant values for variable x.

The advantage of using the quadratic Lagrangian integration allows you to change, at will, the difference between the values of variable x, within the integration range. This *fluid* scheme of selecting values of variable x, comes at absolutely no extra cost. By contrast, if you want to mimic this fluid variation in the intervals of x using Simpson's rule, you have to explicitly break down the integral into smaller integrals—each sub-integral uses a consistent change in the values for variable x.

The quadratic Lagrangian integration allows you to change the sampling rate for points (x, y) based on the rate of variation of y. Within regions of large slopes (and variations in function values) you can use more points to calculate the part of the integral in these regions. Likewise, within regions of small slopes, you have the luxury of using fewer points to calculate the integrals in these regions.

When using the fluid sampling scheme with the quadratic Lagrangian integration, you supply the method with an array of even, distinct, and sorted values for x. You can calculate the values for y, as needed, for each value of x. The sampling rate of points (x, y), the type of function, and the integral range, determine the accuracy of the integral. Using the quadratic Lagrangian integration allows you to significantly reduce the overall computational effort, compared to Simpson's rule, especially if you can tolerate a lower level of accuracy.

Examples

The following table lists (a few) examples of comparing results of the quadratic Lagrangian integration with those of Simpson's rule. The third and fourth columns contain sets of results that represent:

- The calculated integral using either algorithm.
- The calculated integral using analytical expression for the integral.
- The difference between the exact and calculated integrals.
- The number of iterations involved.

Keep in mind that when the quadratic Lagrangian integration uses the same sampling scheme for variable x as does Simpson's rule, it gives results that match those of Simpson's rule. The last column contains sets of results that represent:

- The mean value for the steps used in the quadratic Lagrangian integration.
- The standard deviation of the steps used in the quadratic Lagrangian integration.
- The ratio of the standard deviation to the mean value.

Function	[A,B]	Step	Simpson's Rule	Quadrt Lagrng	Step Stats
1/X	[1,2]	0.071428571	0.693147231 0.693147181 -5.06802E-08 14	0.693148555 0.693147181 -1.37477E-06 14	0.033333333 0.040191848 1.205755429

Function	[A,B]	Step	Simpson's Rule	Quadrt Lagrng	Step Stats
1/X	[1,10]	.9	2.30356496 2.302585093 -0.000979867 10	2.302745865 2.302585093 -0.000160772 10	0.45 0.369921756 0.822048346
1/X	[1,100]	12.375	5.377393388 4.605170186 -0.772223202 8	4.542929894 4.605170186 0.062240292 8	6.1875 3.614208074 0.584114436
1/(X*LN(X)	[1,50]	14.14285714	2.1540272 1.35101767 -0.80300953 6	1.585901989 1.35101767 -0.234884318 7	7.071428571 2.644712743 0.373999782
1/(X*LN(X)	[1,50]	6.125	1.443769068 1.232757489 -0.211011579 8	1.248671904 1.232757489 -0.015914415 8	3.0625 2.694902596 0.879968194
LN(5*X)	[1,10]	0.5625	28.51072914 28.51079214 6.3006E-05 16	28.51078143 28.51079214 1.07076E-05 16	0.28125 0.162515508 0.577832917
LN(5*X)	[1,10]	0.321428571	27.2585711 28.51079214 1.252221045 27	28.51079014 28.51079214 2.00504E-06 28	0.160714286 0.113102643 0.703749777
LN(5X)/X	[1,10]	0.5625	6.356788243 6.356816801 2.85578E-05 16	6.356823642 6.356816801 -6.84196E-06 16	0.28125 0.162515508 0.577832917
LN(5X)/X	[1,10]	0.321428571	6.22953745 6.356816801 0.12727935 27	6.356816832 6.356816801 -3.13982E-08 28	0.160714286 0.113102643 0.703749777
SIN(X)	[1,10]	0.5625	1.379422239 1.379373835 -4.84043E-05 16	1.379760819 1.379373835 -0.000386984 16	0.28125 0.162515508 0.577832917
SIN(X)	[1,10]	0.321428571	1.508274847 1.379373835 -0.128901012 27	1.379631545 1.379373835 -0.00025771 28	0.160714286 0.113102643 0.703749777
X*SIN(X)	[1,10]	0.5625	7.545985683 7.545525501 -0.000460182 16	7.548655661 7.545525501 -0.00313016 16	0.28125 0.162515508 0.577832917
X*SIN(X)	[1,10]	0.321428571	8.816339768	7.549721529	0.160714286

Function	[A,B]	Step	Simpson's Rule	Quadrt Lagrng	Step Stats
			7.545525501 -1.270814267 27	7.545525501 -0.004196028 28	0.113102643 0.703749777
SIN(X)^2	[1,10]	0.5625	4.499087517 4.499088044 5.26976E-07 16	4.496780092 4.499088044 0.002307952 16	0.28125 0.162515508 0.577832917
SIN(X)^2	[1,10]	0.321428571	4.445098964 4.499088044 0.05398908 27	4.497178759 4.499088044 0.001909285 28	0.160714286 0.113102643 0.703749777

The above table has many results of the quadratic Lagrangian integration (where the error of the quadratic Lagrangian integration is marked in red) that show this method doing better than Simpson's rule and using less iterations.

Conclusion

The quadratic Lagrangian integration can be fine-tuned to give good calculated integral by optimizing the sampling of the (x, y) points used in the integration. Depending on the integrated function, integration range, and steps used, the quadratic Lagrangian integration can give better results than Simpson's rule.

Appendix

I used the following Excel VBA listing to conduct thee calculations for the integrals.

```
Option Explicit

Function MyFx(ByVal sFx As String, ByVal X As Double) As Double
  sFx = Replace(sFx, "$X", "(" & CStr(X) & ")")
  MyFx = Evaluate(sFx)
End Function

Sub IntLang()
  Dim X0 As Double, X1 As Double, X2 As Double, XA As Double, XB As Double
  Dim Y0 As Double, Y1 As Double, Y2 As Double
  Dim h As Double, hSqr As Double, A0 As Double, A1 As Double, A2 As Double
  Dim A As Double, B As Double, C As Double, Sum As Double, SumX As Double
  Dim N As Integer, I As Integer, J As Integer, ArrSize As Integer, K As Integer
  Dim sFx As String
  Dim Xarr() As Double

  XA = Range("A2").Value
  XB = [A4].Value
  h = [A6].Value
  hSqr = h * h
  sFx = UCase([A8].Value)
  N = 0
  Sum = 0
  K = CInt((XB - XA) / h - 0.5)
  Do
    X0 = XA + N * h
```

```
    X1 = X0 + h / 2
    X2 = X0 + h
    Y0 = MyFx(sFx, X0)
    Y1 = MyFx(sFx, X1)
    Y2 = MyFx(sFx, X2)
    A0 = 2 * Y0 / hSqr ' Y0/((X0-X1)*(X0-X2))
    A1 = -4 * Y1 / hSqr ' Y1/((X1-X0)*(X1-X2))
    A2 = 2 * Y2 / hSqr ' Y2/((X2-X0)*(X2-X1))
    C = A0 * X1 * X2 + A1 * X0 * X2 + A2 * X0 * X1
    B = -(A0 * (X1 + X2) + A1 * (X0 + X2) + A2 * (X0 + X1))
    A = A0 + A1 + A2
    Sum = Sum + A / 3 * (X2 * X2 * X2 - X0 * X0 * X0) + _
          B / 2 * (X2 * X2 - X0 * X0) + C * (X2 - X0)
    N = N + 1
  Loop Until N >= K
  [C2].Value = Sum
  [C5].Value = N

  ArrSize = 1
  Do While Trim(Cells(ArrSize + 1, 2)) <> ""
    ArrSize = ArrSize + 1
  Loop
  ArrSize = ArrSize - 1
  ReDim Xarr(ArrSize)
  For I = 1 To ArrSize
    Xarr(I) = Cells(I + 1, 2)
  Next I
  N = 1
  Sum = 0
  K = 0
  Do
    K = K + 1
    X0 = Xarr(N)
    X1 = Xarr(N + 1)
    X2 = Xarr(N + 2)
    Y0 = MyFx(sFx, X0)
    Y1 = MyFx(sFx, X1)
    Y2 = MyFx(sFx, X2)
    A0 = Y0 / ((X0 - X1) * (X0 - X2))
    A1 = Y1 / ((X1 - X0) * (X1 - X2))
    A2 = Y2 / ((X2 - X0) * (X2 - X1))
    C = A0 * X1 * X2 + A1 * X0 * X2 + A2 * X0 * X1
    B = -(A0 * (X1 + X2) + A1 * (X0 + X2) + A2 * (X0 + X1))
    A = A0 + A1 + A2
    Sum = Sum + A / 3 * (X2 * X2 * X2 - X0 * X0 * X0) + _
          B / 2 * (X2 * X2 - X0 * X0) + C * (X2 - X0)
    N = N + 2
  Loop Until N >= ArrSize
  [D2].Value = Sum
  [D5].Value = K

  ' Simpson's method
  N = 0
  Sum = 0
  K = CInt((XB - XA) / h - 0.5)
  Do
    X0 = XA + N * h
    X1 = X0 + h / 2
    X2 = X1 + h / 2
    Y0 = MyFx(sFx, X0)
    Y1 = MyFx(sFx, X1)
    Y2 = MyFx(sFx, X2)
    Sum = Sum + (Y0 + 4 * Y1 + Y2)
```

```
    N = N + 1
  Loop Until N >= K
  [E2].Value = h / 6 * Sum
  [E5].Value = N
End Sub
```

The next table represents the Excel worksheet. The worksheet has the following columns:

- Column A is the input column. The cells in red represent user input. Cell A8 contains an expression that the Excel VBA converts into a function value. Notice that the name of the variable used is $X and not just X. Appending the $ to X allows the VBA parser to distinguish between the variable X and the letter X in functions like EXP. The names of the functions in cell A8 must match the names of functions used in Excel formula and NOT in the VBA language. For example to specify a square root you need to use SQRT and not SQR. Likewise to use the natural log type in LN and not LOG.
- Column B is the array of x values used by the quadratic Lagrangian integration. The number of elements in this array must be even.
- Column C shows the results for the quadratic Lagrangian integration when using equidistant points. Cell C2 contains the calculated integral using the quadratic Lagrangian integration. Cells C3 contains an Excel formula (in this example "=LN(LN(5*A4))-LN(LN(5*A2))") that calculates the exact analytical integral using the values in cells A2 and A4. You need to change the formula in cell C3 to match the new contents of cell A8. Cell C4 contains the Excel formula "=C3-C2". Cell C5 shows the number of loops involved in obtaining the integral in C2.
- Column D shows the results for the quadratic Lagrangian integration when using the values in column B. Cell D2 contains the calculated integral using the quadratic Lagrangian integration. Cells D3 copies the value in C2 using the Excel formula "=C3". Cell D4 contains the Excel formula "=D3-D2". Cell D5 shows the number of loops involved in obtaining the integral in D2.
- Column E shows the results for Simpson's rule integration. Cell E2 contains the calculated integral. Cells E3 copies the value in C2 using the Excel formula "=C3". Cell E4 contains the Excel formula "=E3-E2". Cell E5 shows the number of loops involved in obtaining the integral in E2.
- Column F contains the values for the differences in the values of X found in column B. The column uses Excel formulae to calculate these differences, starting with cell F3. This cell contains the formula "=B2-B3". Likewise, cell F4 contains the formula "=B3-B4" and so on. The last entry in column F must calculate the difference for the last value in column B.
- Column G contains labels for the statistics of the values in column F.
- Column H contains the statistics of the values in column F. Cell H2, H3, and H4 contain the formulae "=AVERAGE(F3:F29)", "=STDEV(F3:F29)", and "=H3/H2", respectively. You may need to adjust the range of cells in cells H2 and H3 to include all of data in cells F.

A	B	C	D	E	F	G	H
A	X	Area QLI Eql Spacing	Area QLI	Area Simpson's Rule	Diff X		
1	1	1.443769068	1.248671904	1.443769068		Mean	3.0625
B	2	1.232757489	1.232757489	1.232757489	1	Stdev	2.694902596
50	3	-0.211011579	-0.015914415	-0.211011579	1	Sdev/mean	0.879968194
Step	4	8	8	8	1		
6.125	5				1		
Fx	6				1		
1./$X/LN(5*$X)	7				1		
	8				1		
	9				1		
	10				1		
	15				5		
	20				5		
	25				5		
	30				5		
	35				5		
	40				5		
	50				10		

