The Mathematics Enthusiast

Volume 11
Number 3 *Number 3*

Article 10

12-2014

Numerically Integrating Irregularly-spaced (x, y) Data

B. Cameron Reed

Recommended Citation
Reed, B. Cameron (2014) "Numerically Integrating Irregularly-spaced (x, y) Data," *The Mathematics Enthusiast*: Vol. 11 : No. 3 , Article 10.
DOI: https://doi.org/10.54870/1551-3440.1319
Available at: https://scholarworks.umt.edu/tme/vol11/iss3/10

Numerically Integrating Irregularly-spaced (x, y) Data

B. Cameron Reed[1]
Department of Physics
Alma College

Abstract: This article describes a computer program for numerically integrating under a $y(x)$ curve of experimental data where the abscissa values are not equally-spaced. This technique is based on fitting parabolas to successive groups of three data points, and may be regarded as a generalization of Simpson's rule.

Keywords: integration; Simpson's rule; numerical integration

1. Introduction

A survey of basic techniques of numerical integration is a common element of college-level calculus classes. Virtually all such students can expect to be exposed to the Trapezoidal rule and the somewhat more accurate Simpson's rule, both of which are specific cases of a broader class of techniques known as Newton-Cotes formulas (Press, et al., 1986) More advanced students may encounter more sophisticated techniques such as Gaussian quadrature.

Most textbook examples of these techniques utilize data points that are equally-spaced in the abscissa coordinate. This is not a fundamental requirement, but has the advantage that very compact expressions can be developed for the integral in such cases; a paper previously published in this journal describes how to program such routines into a spreadsheet (El-Gebeily & Yushau, 2007). In many experimental circumstances, however, the (x,y) values are *not* equally

[1] reed@alma.edu

The Mathematics Enthusiast, **ISSN 1551-3440, vol. 11, no. 3**, pp. 643-648

spaced in x. How then can you estimate the area under a "graphical" $y(x)$ curve? Surprisingly, textbooks tend to be silent on this very practical issue. For readers familiar with more advanced numerical methods, one tactic might be to apply an interpolation scheme such as a cubic spline fit. Aside from the issue that such techniques are usually directed more at determining values of the dependent variable at specified values of the independent variable, they demand knowledge of the values of the derivatives of $y(x)$ at the end points of the data - information unlikely to be known in an experimental circumstance.

While the simplest approach to determining the desired integral would be a trapezoidal or "picket fence" - type summation, such a procedure would be aesthetically unsatisfying: physical phenomena are not normally discontinuous. Any sensible approach needs to incorporate some "smoothing," presumably based on some sort of interpolation.

The purpose of this article is to offer an easy-to-use scheme for dealing with such circumstances. The essence of the method, which is an extension of Simpson's rule, is to fit a series of parabolic segments to groups of three successive data points and accumulate the areas under the segments.

Before describing the details of the computation, there is a philosophical issue here that deserves some discussion. This is that if an N-th order polynomial can always be fit exactly through N points, why not build the method to fit higher-order polynomial segments to the data? The answer offered here is that "simplest is best." If there is no model equation for the data, then there is no justification for using a polynomial of *any* specific order, or, for that matter, any particular function at all on which to base computing the integral. Quadratic segments are the lowest-order ones which allow one to build in some "curvature" to the run of $y(x)$. Simpson's

rule is based on fitting parabolic segments to the often presumed equally-spaced data points, so the method developed here can be considered an extension of this time-honored technique.

2. The Integration Method

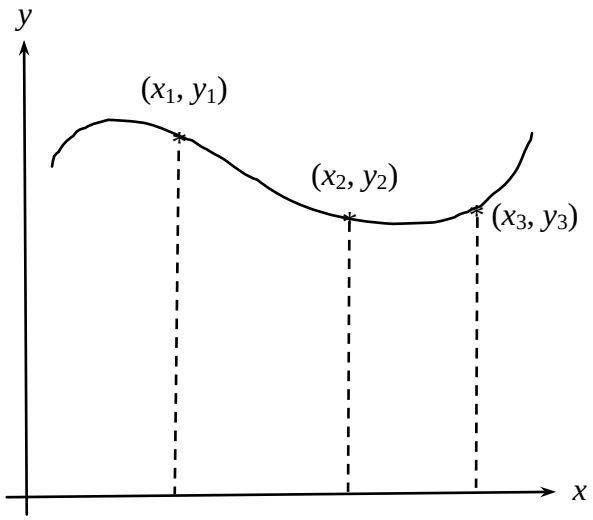

Figure 1. Sketch of a parabolic-segment curve fitted through three successive data points. The curve lies atop two vertically-oriented area segments.

As sketched in Figure 1, consider three successive (x ,y) points in your data table; call them (x_1, y_1), (x_2, y_2), and (x_3, y_3). It is assumed that your data are ordered in terms of monotonically increasing or decreasing values of x, and do not include any "degenerate" points, that is, there can be no duplicate values of x. A unique interpolating parabola can always be fit through three non-vertical points in a plane:

$$y = Ax^2 + Bx + C, \tag{1}$$

where the coefficients are given by inverting a 3 by 3 matrix:

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{pmatrix}^{-1} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}. \tag{2}$$

The area under the three-point parabolic segment is then given by

$$\int_{x_1}^{x_3} y(x)dx = \frac{A}{3}\left(x_3^3 - x_1^3\right) + \frac{B}{2}\left(x_3^2 - x_1^2\right) + C\left(x_3 - x_1\right). \tag{3}$$

The basis of the present routine is to accumulate such parabolically-defined areas from one end of the data domain to the other. After processing the segment defined by points (x_1, y_1), (x_2, y_2), and (x_3, y_3), the program moves on to the next group of three points, (x_3, y_3), (x_4, y_4), and (x_5, y_5), and continues until the entire domain has been fit and integrated over. No model equation for the data is necessary, and the abscissa values of the data points do *not* need to be equally spaced.

If the number of data points N is odd, the entire domain of the data can be fit with segments as in Figure 1 with no remaining unaccounted vertical slices. If on the other hand N is even, one last orphan vertical slice will always remain. In the case of N even, the program developed here fits a parabolic segment to the last three points in the data series and computes the area of the orphan slice using Eq. (3) with limits x_{N-1} and x_N.

A FORTRAN subroutine, AREA, has been developed to carry out this computation. This routine could easily be translated into another language and is available upon e-mail request to

the author; it is also publicly available at the author's institutional website at <https://mail.alma.edu/home/reed@alma.edu/Briefcase/AREA.f>. The call statement for AREA involves five arguments:

AREA(X, Y, N, PAR, INTEG)

The user supplies the column arrays X and Y, and the number of points N (maximum = 50,000). As a check, the routine returns the parity (PAR) of N (+1 if even, -1 if odd). The value of the integral is returned in INTEG. Be sure to declare X, Y, and INTEG as double-precision, and PAR and N as integers. Given the matrix inversions involved, this scheme not easily programmed with a spreadsheet.

How well does AREA perform in practice? As an example, consider the integral

$$\int_0^5 e^{-2x}\sin(3x)\,dx. \tag{4}$$

This function is shown in Figure 2.

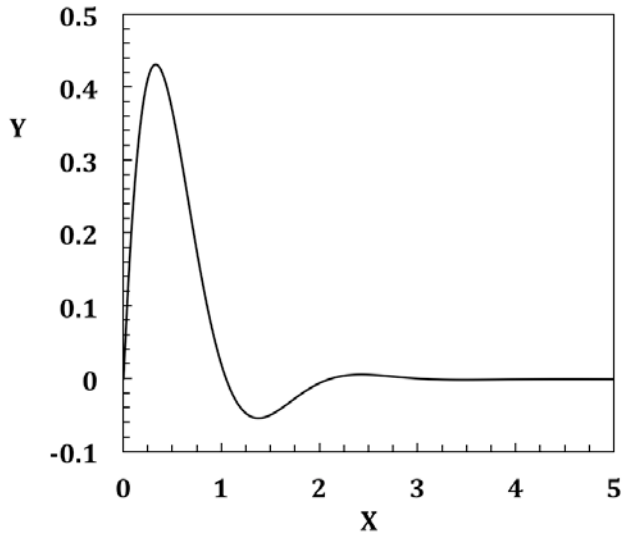

Figure 2. Exponential-sine function of Equation (4).

This integral can be solved analytically, and evaluates to 0.230773. I used a random-number generator to produce N = 101 values of *x* between zero and five, sorted them to be in order of increasing *x*, computed corresponding *y*-values, and then ran the resulting data through AREA. The result was 0.229449, about 0.6% low. I have applied AREA to large data sets (N ~ 44,000) of nuclear-reaction cross-section values where *y*(*x*) can be extremely variable, and routinely obtain agreement within 1% values listed on research-level websites. AREA thus appears to be quite accurate even when rapidly-varying data are involved.

I hope that students and researchers who occasionally need to compute such data-based integrals but who do not wish to become submerged in the minutiae of numerical analysis will find AREA a useful tool. As always, *caveat emptor*: no guarantee is offered as to how the routine will perform for extremely rapidly-varying or poorly-sampled data. But for routine cases of "well-behaved" data, it should prove entirely adequate.

Acknowledgement

I am grateful to Mel Nyman for comments on curve fitting.

References

El-Gebeily, M. & B. Yushau, (2007) *Numerical Methods with MS Excel*. TMME *4*(1), 84-92 (2007)

Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1986). *Numerical Recipes: The Art of Scientific Computing*. Cambridge: Cambridge University Press.