ISSN 1870-4069

Higher Order Numerical Derivatives for Data Processing in Optics and Photonics

Luis David Lopez-Zavala, Oleksiy V. Shulika

Universidad de Guanajuato,
Departamento de Ingeniería Eléctrica,
Mexico

luiszavala@live.com.mx, osulika@ugto.mx

Abstract. Data processing is fundamental for the correct interpretation and analysis of experimental results, so it is common to compute derivatives of noisy sets of data. The same situation is observed when data to be processed are obtained from numerical simulations. Here we present three different methods for numerical calculation of higher-order derivatives, up to 20th order. The complex integration and automatic differentiation show the best results.

Keywords: Higher-order derivatives, numerical approximation, holomorphic function, data processing, automatic differentiation, integration in complex plane.

1 Introduction

When working on any experimental development, the results often includes a mixture of noise and measurement-related phenomena, so it is not easy to draw conclusions from a set of raw data, and this is where data processing plays an important role. Differentiation is the process of determining how quickly a function varies, as the quantity on which it depends change. Geometrically, it can be seen as the slope of a tangent line at a given point within the domain of the function, so the derivative of a functions is capable of giving information about the behavior of functions [4].

The derivative of the result of another derivative is called the second derivative of the original function, and geometrically represents the rate of change of the slope of the original function's tangent line. Higher-order derivatives are useful, especially, when dealing with functions such as those from real experimental data. Here we present three different derivative methods are studied and compared in terms of accuracy and derivative order to choose the best for row data processing applications.

2 Richardson's Extrapolation

Richardson's extrapolation makes a faster convergent sequence from another already convergent sequence in order to get a recursive method that is reliable in convergence speed and accuracy. To do so, two different derivative approximations, $D(h)$ and $D(h_1)$ for two different parameters h and h_1, supposing that $h_1 > h$, are considered.

Generally, an approximation of a derivative, which depends on the length of step h and its truncation error, can be expressed in the series form such as the finite differences method, which arises from the following series expansion:

$$f'(x_0) = D(h) + a_1 h^{n_1} + a_2 h^{n_2} + a_3 h^{n_3} + \cdots n_1 < n_2 < n_3. \quad (1)$$

Expanding the derivative approximation for h_1, and using the ratio between the step sizes $r = h_1/h \Longrightarrow h_1 = rh$ an equation in terms of h can be written as:

$$f'(x_0) = D(rh) + a_1(rh)^{n_1} + a_2(rh)^{n_2} + a_3(rh)^{n_3} + \cdots \quad (2)$$

Then, to eliminate a truncation error term, Eq. (1) is multiplied by r^{n_1}, and Eq. (2) is subtracted from it, to get:

$$f'(x_0) = \frac{r^{n_1} D(h) - D(rh)}{r^{n_1} - 1} + a_2 h^{n_2} \frac{r^{n_1} - r^{n_2}}{r^{n_1} - 1} + a_3 h^{n_3} \frac{r^{n_1} - r^{n_3}}{r^{n_1} - 1} + \cdots \quad (3)$$

The Eq. (3) represents the extrapolation process and substituting:

$$D_1(h) = \frac{r^{n_1} D(h) - D(rh)}{r^{n_1} - 1}, \quad b_2 = a_2 \frac{r^{n_1} - r^{n_2}}{r^{n_1} - 1}, \quad b_3 = a_3 \frac{r^{n_1} - r^{n_3}}{r^{n_1} - 1}, \cdots \quad (4)$$

In the result a new approximation arises, in which an error term has been eliminated, and becomes a formula with error $O(h^{n_2})$. Finally, substituting r into Eq. (3) gives:

$$f'(x_0) = \frac{\left(\frac{h_1}{h}\right)^{n_1} D(h) - D(h_1)}{\left(\frac{h_1}{h}\right)^{n_1} - 1} + b_2 h^{n_2} + b_3 h^{n_3} + \cdots \quad (5)$$

2.1 Recursion

From a set of approximations F, obtained by different values of h, it is possible to apply the Richardson's formula in a recursive way using the following expression:

$$F_m^n = \frac{\left(\frac{h_n}{h_{n+m}}\right)^{\beta} F_{m-1}^{n+1} - F_{m-1}^{n}}{\left(\frac{h_n}{h_{n+m}}\right)^{\beta} - 1}. \quad (6)$$

The Eq. (6) can only be used when the truncation error of F has the form:

$$\sum_{k=1}^{\infty} a_k h^{\beta k}. \quad (7)$$

So a special case of Richardson's recursive formula can be obtained by using the centered finite difference formula as the base convergent sequence, whose truncation error can be expressed as:

$$2 \sum_{k=1}^{\infty} \frac{h^{2k}}{(2k+1)!} \; f^{2k+1}(x_i). \quad (8)$$

And therefore the condition for the recursive formula is satisfied with $\beta = 2$. Finally considering that $h_n = 2h_{n-1}$ the Expr. (6) becomes:

$$F_m^n = \frac{4^m F_{m-1}^{n+1} - F_{m-1}^n}{4^m - 1}. \tag{9}$$

This particular case is the most widely used form of Richardson's extrapolation due to the oversimplifying of the calculations [1].

3 Complex Integration Method

The concept of holomorphic function in complex number's theory states that if a complex valued function is complex differentiable in a neighborhood of each point in a domain, then it is infinitely differentiable and locally equal to its Taylor series [5].

Consider a function $f(z)$ holomorphic in the simple closed contour C and at all points inside it, then a circle C_0 with center in z_0 and radius r small enough that all points of C_0 are inside C, so the function is holomorphic in C, C_0 and at all the points of the doubly connected domain. Then considering the function $f(z)/(z - z_0)$, which is holomorphic at every point except $z = z_0$, the contour C can be related to C_0 trough the contour deformation principle, since the singularity violates the Cauchy-Goursat theorem the relation is not nullified, and is written as:

$$\oint_C \frac{f(z)}{z - z_0}\, dz = \oint_{C_0} \frac{f(z)}{z - z_0}\, dz. \tag{10}$$

This allows to evaluate the integral, using polar coordinates since the contour C_0 is a circle, to solve for $f(z_0)$. To do so, C_0 is parametrically expressed through the polar angle θ and with the following change of variable $z = re^{i\theta} + z_0$ and $\frac{dz}{d\theta} = ire^{i\theta}$, the Eq. (10) is rewritten as:

$$\oint_{|z-z_0|=r} \frac{f(z)}{z - z_0}\, dz = \int_0^{2\pi} f(z(\theta))\frac{dz}{d\theta} d\theta = i\int_0^{2\pi} f(z_0 + re^{i\theta})\, d\theta. \tag{11}$$

To know the value $f(z)$ in $z = z_0$, the limit when r tends to 0 is applied in the right side of Eq. (11):

$$\oint_C \frac{f(z)}{z - z_0}\, dz = \lim_{r\to 0} i\int_0^{2\pi} f(z_0 + re^{i\theta})\, d\theta = i\int_0^{2\pi} f(z_0)\, d\theta = i\theta f(z_0)|_0^{2\pi} = 2\pi i f(z_0). \tag{12}$$

And solving the Eq. (12) for $f(z_0)$, the Cauchy integral formula is obtained:

$$f(z_0) = \frac{1}{2\pi i}\oint_C \frac{f(z)}{z - z_0}\, dz. \tag{13}$$

Given the holomorphic criteria, $f(z)$ has derivatives of all the orders in the domain in which it is analytic and its derivatives are also analytic in the domain, hence Eq. (13) can be generalized as:

$$f^{(n)}(z_0) = \frac{n!}{2\pi i}\oint_C \frac{f(z)}{(z - z_0)^{n+1}}\, dz. \tag{14}$$

3.1 Parameterized Cauchy Integral Formula

Since the Cauchy's formula is written for complex functions, the contour C must be parameterized as a function of time $C(t) = a + e^{it}$, for $0 \leqslant t \leqslant 2\pi$ where a is the center of the unitary circle, in order to apply Cauchy's formula for real functions. The parameterized Cauchy's integral formula states that if a function $f(t)$ is analytic within the simple connected domain D and C is any simple closed contour wholly located at D, then for any point a within C:

$$f^{(n)}(a) = \frac{n!}{2\pi} \int_0^{2\pi} f(a + e^{it}) e^{-int} \, dt. \tag{15}$$

4 Automatic Differentiation Method

The automatic differentiation method is based in decompose a function in a set elemental operations sequence easily differentiable each and computing the derivative by attach each elemental derivative using the chain rule, taking the problem by a combination of numerical and symbolical techniques which take advantage of the benefits of both. Since elemental function's derivatives are already known the results reach symbolical accuracy [3].

A good graphical representation of the evaluation path of the method is a graph, which is an important tool based in dynamic coding and very useful to identify the dependency relations between variables, that stores the result of each elementary operation in intermediate variables v_j. The derivative of single dependent variable function with respect to a independent variable can be calculated applying systematically the chain rule from the input vertices to the output vertice of the graph, this evaluation path is called Forward mode since the evaluations are propagated forward from input to output.

Considering that the derivative of a vertice with respect to the independent variable is the sum of the values of the incoming edges, and each contribute with the total derivative of the vertice at the begin of the edge times the partial derivative of the vertice that the edge points with respect to the vertice at the begin of the edge [2]. This evaluation can be generalized as a column of a Jacobian matrix of a function $f : \Re^n \to \Re^m$, if the respective input variable is initialized in 1 and the rest in 0. In this way the entire matrix will be formed after the n-times applications of the forward mode. It is easy to think that if each elemental operations knows its derivative, the recursive application of the method should result in the higher order derivatives of the original function.

5 Results

To compare which method is the best for data processing, the accuracy and maximum order of the higher-order derivatives of a noisy exponential function, computed by each method, are studied. An exponential function is chosen due that all its derivatives are equal to the exponential function, also if results are plotted on a semi-logarithmic scale the derivatives looks like a straight line.

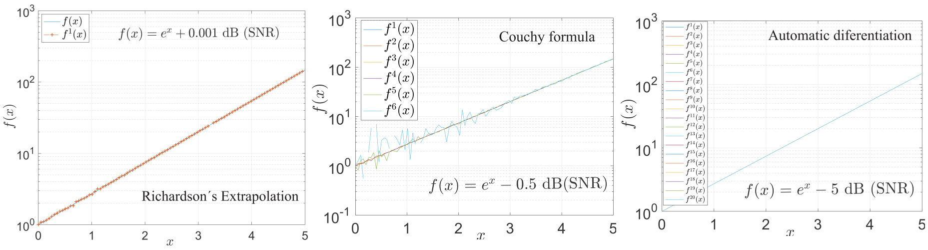

Fig. 1. Higher-order numerical derivatives computed by each of the three methods.

Then, for each method a noisy exponential function, generated by adding a sample noisy vector to the evaluated function vector, is considered. The sample noisy vector is generated by the Matalab's random number generation function which return a standard normally distributed matrix numeric array of any size.

6 Conclusions

Richardson's extrapolation method can compute accurately at least the first four derivative orders of a fundamental function, although its noise tolerance is very low since noise that generates a signal to noise ratio of 0.01 dB is enough to get divergent results for the first derivative. On the other hand, the complex integration method is capable of calculating the first sixteen derivative orders of a fundamental function and the fourth derivative of a noisy signal, filtering a signal with an SNR of up to -0.05 dB.

Finally, the automatic differentiation method is the most accurate way to compute higher order derivatives of fundamental functions, no matter the order, avoiding truncating and round error by its symbolic properties, however, the higher the derivative order, the higher the processing time, so it is not recommended for real time applications. Its noise tolerance is enough to compute accurate derivatives of signals with an SNR up to -7 dB.

Acknowledgments. This work is supported by University of Guanajuato under research grant CIIC 272.2021.

References

1. Burden, R. L., Faires, J. D.: Análisis numérico. Thomson Learning (2002)
2. Callejo-Goena, A., García-de-Jalón-de-la-Fuente, F. J., Hidalgo, A.: Diferenciación automática de fuerzas en la integración implícita de sistemas multicuerpo. In: XVIII Congreso Nacional de Ingeniería Mecánica, pp. 3–5 (2010)
3. Günes-Baydin, A., Pearlmutter, B. A., Andreyevich-Radul, A., Mark-Siskind, J.: Automatic differentiation in machine learning: A survey. The Journal of Machine Learning Research, vol. 18, no. 1, pp. 1–43 (2018)
4. Riley, K. F., Hobson, M. P., Bence, S. J.: Mathematical methods for physics and engineering. American Association of Physics Teachers (1999) doi: 10.1017/cbo9781139164979
5. Wunsch, A. D.: Variable compleja con aplicaciones. Addison-Wesley Longman (1999)