Estimation of derivatives by the method of differenceless derivatives

Y. Mahotin

e-mail: yuri.mahotin@yahoo.com

Abstract—We have developed a numerical algorithm called differenceless derivatives which allows us to calculate an unlimited number of derivatives. One of the applications of this algorithm is the extrapolation and prediction of functions in a wide range of arguments, which can be used in various fields of science and technology.

Keywords: derivative, numerical algorithm

INTRODUCTION

Derivatives and especially numerical derivatives are widely used in mathematics, physics, applied sciences and almost everywhere where there is a need for this. The first and second derivatives, known as speed and acceleration, are used almost everywhere, while the third, fourth, fifth and sixth derivatives are used very rarely in special applications, higher order derivatives are never used anywhere. And there are several reasons for this, in the future we will consider only numerical derivatives. Firstly, everything rests on the fact that it is very difficult to calculate derivatives of a higher order. Secondly, there is no theory and applications that use higher order derivatives, which follows from the first reason. Forward, backward, and central differences are commonly used to calculate derivatives and require the function $f(x)$ to be evaluated with an equidistant step h, $f(x+h)$, which may not be possible if the data comes from a measurement. In addition, the step size h must be very small, which again may not always be feasible. To solve the above problems, we have developed a new algorithm called differenceless derivatives to calculate derivatives from the first to the n-th, where n can be very large 10, 100 and even 100000, and there is no need to have a small step h, the step h can be big and non-equidistant.

THE METHOD OF DIFFERENCELESS DERIVATIVES

Consider a smooth n times differentiable function $f(x)$, then the function $f(x)$ at point $(x+h)$ can be approximated by Taylor series:

$$f(x+h)=f(x)+f^{'}(x)h+\frac{1}{2!}f^{''}(x)h^2+\frac{1}{3!}f^{'''}(x)h^3+\frac{1}{4!}f^{(4)}(x)h^4+\frac{1}{5!}f^{(5)}(x)h^5+\cdots\frac{1}{n!}f^{(n)}(x)h^n \tag{1}$$

If at the point x all derivatives from the first to the nth are known, it is easy to calculate the value of the function at any point $x+h$, such a problem is called the direct problem. However, we will be interested in the inverse problem: on the set of points $x+h_i,(1<i<m)$ the values of the function $f(x+h_i)$ are given, to find the values of the derivatives $f(x)$at point x. For the sake of convenience, we introduce the following notation: $\Delta f(h)=f(x+h)-f(x)$,

$\vec{y}^T=(y_1,y_2,\ldots,y_n)^T=(f^{'}(x),f^{''}(x),\ldots,f^{(n)}(x))^T$, then equation (1) can be written in the following form:

$$hy_1+\frac{1}{2!}h^2y_2+\frac{1}{3!}h^3y_3+\ldots+\frac{1}{n!}h^ny_n=\Delta f(h) \tag{2}$$

This linear equation with respect to the unknown vector $\vec{y}$ must hold for any values of h, such a requirement for this equation leads us to a system of linear equations written in matrix form as:

$$A\vec{y}=\vec{b} \tag{3}$$

where the quadratic matrix $(n\times n)\ A$ is a function of h_i, i.e. $A(h_i)$, and the vector $\vec{b}$ is a function of $\Delta f(h_i)$ and h_i, i.e. $\vec{b}=\vec{b}(\Delta f(h_i),h_i)$. The solution of the equation (3) is: $\vec{y}=A^{-1}\vec{b}$, which represents the values of the derivatives from the first to the nth, for that reason the algorithm is called "differenceless derivatives". A distinctive feature of this algorithm is that it is impossible to calculate, for example, only the first derivative, all derivatives will be calculated, and as a consequence of this, since all components of equation (2) are included in the calculations, the accuracy of calculating the derivatives is much higher than in calculations using, for example, central differences. Another advantage of the differenceless derivatives algorithm is that the requirements for the step size h are reduced, so if the optimal step size h for the forward differences is about 1e-7, then the differenceless derivatives algorithm can use a value of 0.1 and even more, which is very necessary if the data comes from measurements. In what follows, we will refer to all data used as measured data.

To test the new differenceless derivatives algorithm, we developed a Linux program written in C called *dld* (DifferenceLess Derivatives). Matrix A is a dense matrix, so we chose the Gauss-Jordan elimination algorithm as a linear solver. The input parameters of the program are the values of the function at the point x ,at which the derivatives will be calculated, and at all points $x+h_i,(1<i<m)$, presented in the form of a table.

TESTING THE ALGORITHM OF DIFFERENCELESS DERIVATIVES.

To test the method and the developed program, we choose several simple functions, the values of the derivatives of which are well known. We took the quadratic function $f(x)=x^2$ as the first function and evaluated the values of ten derivatives at the point 0, where the second derivative equal to 2.0 and all others are 0.0. We used 20 measured points, with a minimum step $h=0.1$ and a maximum step $h=1.0$, which are shown in Figure 1 with red dots, the function x^2 is shown by the black line and the red line shows the function calculated by formula (1), there is no difference between these two curves. True and computed derivatives are shown in Table 1, the maximum error of less than 2e-7 is observed for the tenth derivative, which is considered a good result, which indicates the good performance of the developed algorithm.

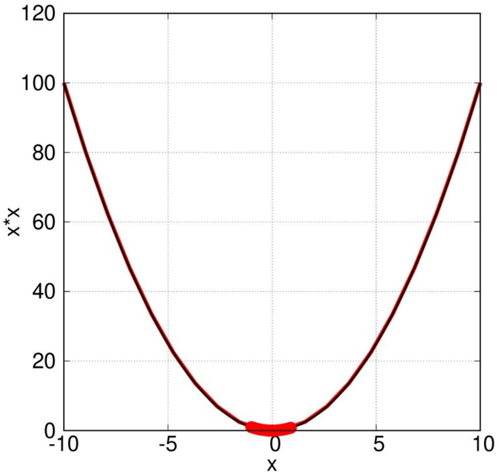

Figure 1

Table 1

n	True derivative	Computed derivative
1	0	2.7904027900559096e-16
2	2	2.0000000000000040e+00
3	0	-3.0421844040265611e-14
4	0	-5.4706735760927060e-13
5	0	2.9452872292501532e-12
6	0	5.7939746415751719e-11
7	0	-2.1727182459944024e-10
8	0	-4.4938076716560169e-09
9	0	8.9952409387038843e-09
10	0	1.9308944831574163e-07

The following test function is an exponential function *exp(x)*. This is a fast-growing function, shown as a black line in Figure 2 on a logarithmic scale, so a large number of derivatives is required to evaluate the exponent using the Taylor series. In this test, we used 16 *x* points in the range [-1, 1], shown as red dots on the Figure 2. We have calculated ten derivatives that extrapolate the exponential function as a red curve. This curve describes well the exponential function in the range [-3,7], the value of this range is 5 times greater than the *x* range used to calculate the derivatives. This result shows that the algorithm is good at predicting the behavior of functions if the values of the function are known within a limited range.

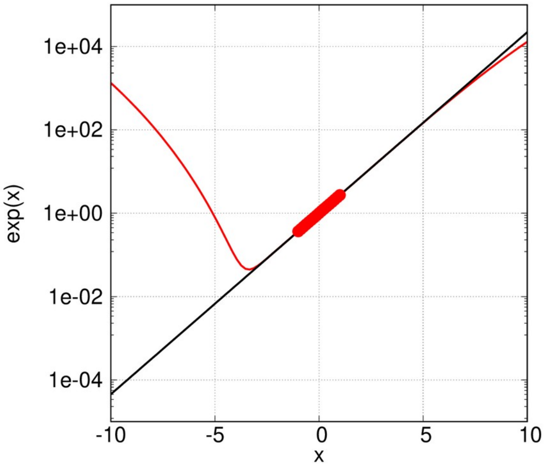

Figure 2

In the last example, we tested the periodic function sin(x) shown by the black curve in Figure 3. We have chosen 10 measured points *x* in the range [-π/2, π/2], shown in Figure 3 by red points, and calculated 10 derivatives. Approximating function , the red curve in Figure 3, describes well the sinusoidal function in the range [-4, 4], which is 2.5 times greater than the range of used for calculation *x*.

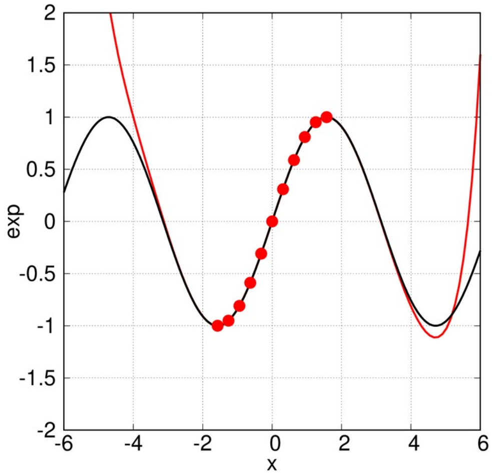

Figure 3

PREDICTION OF THE TRAJECTORY OF BALLISTIC OBJECT

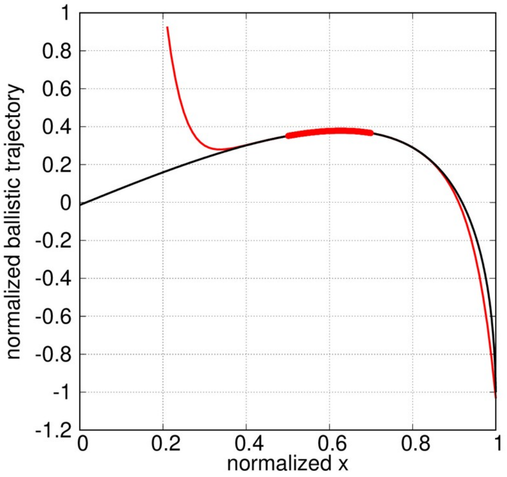

Figure 4

As described above, we tested the differenceless derivatives algorithm on well-known analytic functions, now we will test the algorithm on a function unknown to us. To get the data we needed, we used a ballistic simulator, which is freely available on the web, the details of which we do not know. We modeled the trajectory of a ballistic object shown in figure 4 as a black curve in a normalized form. As measured data, we chose 241 points in the range (0.5,0.7), shown in the Figure 4 as large red dots, but they all look like a wide red line, due to their large number. We calculated 9 derivatives. The red thin curve represents the predicted trajectory. The trajectory is described quite well in the range (0.375, 1.0), if we compare the size of this range with the range of the measured data, then it will be more than 3 times larger. In other words, if we know part of the trajectory, we can predict three times as much of the trajectory. By a similar method, one can calculate the trajectory of a non-ballistic flying object, like non ballistic missile. We believe that in order to improve the accuracy of the simulation, it is necessary to increase the number of derivatives used, but we cannot do this using 64-bit floating point arithmetic, it is necessary to apply high precision floating point arithmetic.

CONCLUSIONS

We have developed a differenceless derivatives algorithm to calculate a large number of derivatives that are not available in modern numerical methods. Testing of the algorithm and an implementation of the developed *dld* program showed the great potential of the new method. We easily calculated 10 derivatives, we have never seen such results before. Our plans for the future include the development of a program based on high precision floating point arithmetic. In conclusion, we note that we have developed an algorithm for one-dimensional functions, a similar algorithm can be applied to multidimensional functions, which expands the scope of its application.