Highly Accurate Derivatives: A Matlab Tool for Teaching and Research

Abdulwahab A. Abokhodair
Department of Earth Sciences, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia

Abstract – *Many computational tasks in science and engineering require highly accurate numerical derivatives. This paper presents a method of derivative approximation that avoids the subtractive cancellation problem that plagues ordinary finite difference (FD) and thus yields near-exact first-order derivatives. Extension of the method to second-order derivatives allows approximations with considerable reduction in round-off errors. An overview of the method and a summary of its implementation are discussed and illustrated. Numerical experiments and application examples demonstrate that the method outperforms ordinary FD schemes in accuracy, efficiency, and numerical stability. Hence, the technique has strong potential as an effective teaching and research tool in engineering and scientific computing.*

Keywords: *Numerical differentiation, derivative approximation, optimization, Hessian, Jacobian, Gradient.*

1 Introduction

The approximation of derivatives is a central and costly step in most computational tasks in science and engineering. Data inversion and parameters estimation problems, for example, require repeated computations of gradients, Jacobians and Hessians of the objective functions and associated active constraints. Inaccuracies in the approximation of these derivative structures not only degrade the performance of the optimizing algorithm but also limit the accuracy of achievable solutions. Therefore, development of efficient and accurate methods to approximate derivatives is of paramount importance.

Because of its ease of implementation, finite difference (FD) is routinely used by teachers and researchers to approximate derivatives of functions. However, FD schemes are well known to be neither efficient nor accurate. A major drawback of the FD method is its critical dependence on the differencing interval or step-size. For maximum accuracy, an optimum step size must be sought, often by trial and error, a process that significantly impedes the efficiency of the method (Mark and Workman 2003; Burge and Newman 2003). Moreover, from the perspective of filter theory, generic FD differentiation filters have the undesirable property of noise amplification (Orfanidis 1996). In the context of numerical optimization, Hong, et al. (2012), point out that use of suboptimum FD step-size in iterative procedures introduces statistical noise that adversely impact not only the convergence rate of the optimization algorithm, but more importantly the accuracy of the resulting solution as well as the solution statistical properties and confidence interval.

In this paper, I describe a method based on the theory of complex variable (hereafter referred to as complex perturbation method - CPM) as an alternative to FD with superior qualities. The technique was originally reported for first-order derivatives of univariate functions by Squire and Trapp (1998) who demonstrated that it was highly accurate, extremely robust and very easy to implement. The method has since been gaining recognition in many areas of computational fields, and has successfully been applied in several large scale studies including sensitivity analyses, aerodynamic design optimization and pseudospectral algorithms (e.g. Cerviño and Bewley, 2003; Martin et al., 2001; Burge and Newman, 2003; Vatsa, 2000; Wang, 2004; De Pauw and Vanrolleghem, 2006). Generalization of CP to multi-parameter functions, and extension to second-order derivatives were reported by the present author (Abokhodair, 2007). In addition to bringing the technique to the attention of the wider computational science community, the objective of this article is to present an overview of CPM, demonstrate by examples its accuracy and numerical stability, and illustrate its implementation in some computational problems. The aim is to highlight the simplicity, power and wide applicability of the technique, and thereby facilitate its use by students and researchers.

2 How CPM Works

A simple example illustrates how the CPM method works: Let $f(x) = x\, e^{-\sin x}$ with exact derivative $f'(x) = (1 - x\cos x)e^{-\sin x}$. To estimate $f'(x)$ by the CP method, the key idea is to perturb the target variable x with a pure imaginary step ih, $\left(i = \sqrt{-1},\ \ h \ll 1\right)$, and construct the complex argument $z = x + ih$. Thus, the 'complexified' version of the original real-valued function becomes:

$$f(z) = z\, e^{-\sin z}\ .$$

For $h << 1$,we may write:

$\sin z = \sin(x + ih) \cong \sin x + ih\cos x$,

therefore, $e^{-\sin z} = e^{-\sin x} e^{-i\alpha}$, where $\alpha = h\cos x$.

Now $e^{-i\alpha} = \cos\alpha - i\,\sin\alpha \cong 1 - ih\cos x$,

$e^{-\sin z} = (1 - ih\cos x)e^{-\sin x}$, and,

$f(z) \cong (x + ih)(1 - ih\cos x)e^{-\sin x}$. Finally we have:

$$f\left(z\right)\cong xe^{-\sin x}+ih\left(1-x\cos x\right)e^{-\sin x}$$

This is a remarkable result. The complexified function $f(z)$ returns $O(h^2)$ approximations of the original real-valued function and of its first derivative in the real and imaginary parts respectively. Moreover, the first derivative returned in the imaginary part of $f(z)$ involves no differencing operation, thus precluding any subtractive cancellation and allowing for near-exact approximations. This is the essence of the CP method. Note also how the complex data type serves here as a carrier of both the function approximation in the real part and the derivative approximation in the imaginary part. This closely resembles true automatic differentiation systems implemented in object-oriented programming. The result above is general as may be proven by a Taylor expansion of $f(z)$: $f(z)=f(x+ih)\approx f(x)+ihf'(x)+O(h^2)$, which is the original formula reported by Squire and Trapp (1998).

Figure 1 compares the absolute relative errors of the CPM and centered finite difference (CFD) approximations of $f'(z)$ computed with respective step sizes of h= 10^{-25} and 10^{-8} As seen in the figure, the CPM relative error fluctuates randomly near machine epsilon (10^{-16}), whereas the CFD error is eight orders of magnitude larger, with a mean value of about 10^{-8}. The Matlab script in box 1, computes the CPM derivative of this example.

```
x=pi*(-1:.02:1)';
h=1.0e-25 ;
f=@(x) x.*exp(-sin(x));
z=complex(x,h);
fc=f(z);
cpfp=imag(fc)/h;
```

Box 1: Matlab script to compute f(x) and f'(x) approximations of the heuristic example.

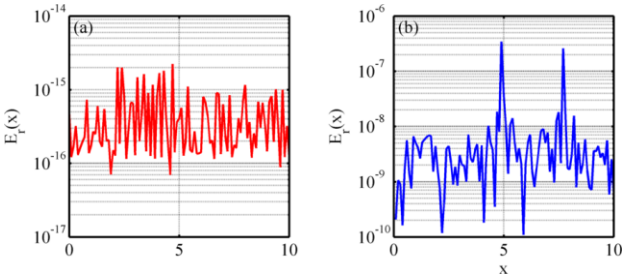


Figure 1: The absolute relative errors of the CP (a), and CFD (b) approximations of $f'(x)$ for the heuristic example using step sizes h = 10^{-25} and 10^{-8} respectively.

3 Computing Jacobians

The generalized CPM formula for vector-valued functions of several variables is:

$$\mathbf{F}(\mathbf{z})=\mathbf{F}(\mathbf{x}+ih\mathbf{e})=\mathbf{F}(\mathbf{x})+ih\mathbf{e}^T\mathbf{J}(\mathbf{x})+O(h^2), \qquad (1)$$

where $\mathbf{F}(\mathbf{x})=\left[f_k(\mathbf{x})\right]_{k=1}^{N}$ $\mathbf{x}=\left[x_k\right]_{k=1}^{M}$, and the $N\times M$ matrix $\mathbf{J}(\mathbf{x})$ is the Jacobian with respect to $\mathbf{x}$, (Abokhodair, 2007). The imaginary part of equation (1) which provides the approximation formula for first derivatives is:

$$\mathbf{e}^T\mathbf{J}(\mathbf{x})=\frac{1}{h}Im\left[\mathbf{F}(\mathbf{z})\right]+O(h^2) \qquad (2)$$

The Jacobian (gradients are special cases) is a first-order derivative structure extensively used in scientific computing in various contexts. Implementation of equation 2 is illustrated for the Jacobian of the environmental model in equation 3, which is a standard test function used for parameters calibration and tuning. The function models a pollutant spill caused by a chemical accident and returns the concentration of the pollutant at the space-time vector $\mathbf{x}=(x,t)$, where $0\le x\le 3$ and $t > 0$ (Bliznouk, et al. 2008)

$$\mathbf{F}(\mathbf{p},\mathbf{x})=\frac{M}{\sqrt{Dt}}\exp\left(\frac{-x^2}{4Dt}\right)+\frac{M}{\sqrt{D(t-\tau)}}\exp\left(\frac{-(x-L)^2}{4D(t-\tau)}\right) \qquad (3)$$

$\mathbf{p}=[M,D,L,\tau]$ is the vector of parameters: M = mass of pollutant; D = diffusion rate; L,τ = location and time of the second spill. Figure 2 depicts the residual errors in the FD-based Jacobian compared to the CP Jacobian. As seen in the figure, the residuals are relatively large (> 10^{-6}) and systematic in character. Accumulation of such errors in iterative procedure can contribute to the ill-conditioning of the computed Jacobian leading at best to biased solutions or at worst to failure of the computational routine. To appreciate the significance of the errors indicated in figure 3, suppose that the computed Jacobian $\tilde{\mathbf{J}}$ is: $\tilde{\mathbf{J}}=\mathbf{J}+\delta\mathbf{J}$, with an error of $\delta\mathbf{J}$, and it is entered into a computation of the form $\mathbf{y}=\mathbf{J}\,\mathbf{x}$, then an upper bound on the relative error in the solution $\mathbf{x}$ is (e.g. Treferthan, and Bau, 1997) :

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|}\le\kappa(\mathbf{J})\frac{\|\delta\mathbf{J}\|}{\|\mathbf{J}\|},$$

which in the present case is not negligible being of the order 10^{-2}.

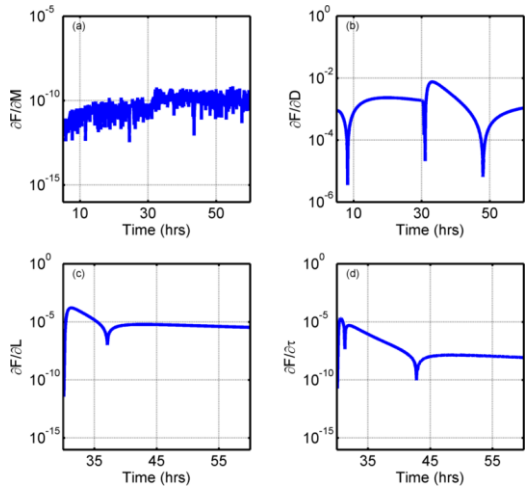


Figure 2: Residual errors in the Jacobian matrix of the environmental function in equation 3

4 Computing Hessians

The approximation formula for first order derivative (equation 1) is the only subtraction-free formula that can be obtained from complex variable theory. However, at the expense of a small lose in accuracy this formula can be extended to second-order derivatives by any ordinary finite difference scheme. This may be accomplished by perturbing the target variable say *x*, by a full complex step $\delta z = \delta x + ih$ and then expanding in Taylor series. The approximation formula so obtained using a centered difference (CFD) scheme is (Abokhodair, 2007):

$$\mathbf{e}^T \mathbf{H}\, \mathbf{e} = \frac{1}{2h\delta x} \mathrm{Im}\left[\Delta f(\mathbf{z})\right] + O(h^2) \qquad (4)$$

where $\mathbf{H}$ is the hessian matrix.

The hessian matrix, a second-order derivative structure, plays a crucial role in optimization algorithms allowing for the quadratic approximation of the objective function locally. To demonstrate application of equation 4, the Colville function is used, which is a standard optimization test function defined as (Surjanovic and Bingham, 2013):

$$f(\mathbf{x}) = g_1(\mathbf{x}) + g_2(\mathbf{x})$$
$$g_1(\mathbf{x}) = 100\left(x_1^2 + x_2\right)^2 + \left(x_1 - 1\right)^2 + \left(x_3 - 1\right)^2 + 90\left(x_3^2 + x_4\right)^2$$
$$g_2(\mathbf{x}) = 10.1\left[\left(x_2 - 1\right)^2 + \left(x_4 - 1\right)^2\right] + 19.8\left(x_2 - 1\right)\left(x_4 - 1\right)$$

The function is usually evaluated on the hypercube: $[x_i]_{i=1}^4 \in [-10, 10]$, and has a global minimum: $f(\hat{\mathbf{x}}) = 0$, at $\hat{\mathbf{x}} = (1,1,1,1)$. Table 1 compares the element-wise relative errors in the Hessians of the objective near its minimum value computed by the CPM and CFD schemes. As seen in the table, the CPM-based Hessian is 7 orders of magnitude more accurate than that of the CFD-based approximation. In terms of norms, the relative errors are of the orders: 10^{-12} and 10^{-6}, respectively a substantial gain in accuracy by the CPM scheme (Eqution 4) despite the fact that it involves a differencing operation.

Table 1: Relative errors in the Hessian of Colville test function (Eqution 5)

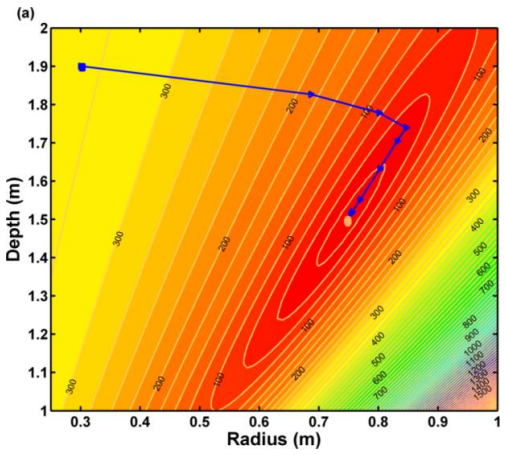
CPM Hessian × 10^{-10}			
0.0655	0.0655	0	0
0.0655	0.1874	0	0.0655
0	0	0.1902	0.0655
0	0.0655	0.0655	0.0655

CFD Hessian × 10^{-3}			
0	0	0.2032	0.0508
0	0	0.4064	0
0.2032	0.4064	0	0
0.5080	0	0	0

5 Application Examples

Use of the CP method in computational problems is demonstrated with an optimization example. In near-surface geophysical applications (e.g. archaeology and environmental studies), ground magnetics is often the method of choice for locating small buried metallic objects such as archaeological artifacts, drums and unexploded ordnances (UXOs). For this purpose, an isolated dipole or, equivalently, a uniformly magnetized spherical source is an adequate model of the anomaly source. Assuming induced magnetization, the total-field magnetic anomaly of a sphere of radius R and magnetization J buried at depth zo is given by (e.g. Blakely, 1995):

$$\Delta B(\mathbf{p}, x) = \frac{KR^3}{r^5}\left(A\,x^2 + z_o B\,x + C\,z_o^2\right) \qquad (5)$$
$$A = 3\cos^2(I) - 1 \quad B = -3\sin(2I) \quad C = 3\sin^2(I) - 1$$



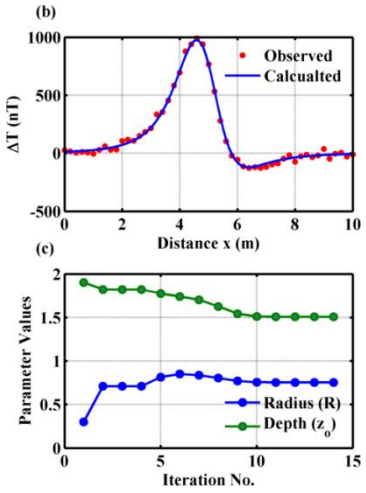


Figure 3: Results of the inversion example - (a) contours of the objective function and superimposed convergence path (blue); (b) observed data and solution-based model response; (c) iteration paths of the two parameters.

In equation 5, $\mathbf{p} = [R, z_o]$ is the vector of parameters, I is the geomagnetic inclination and K is a constant. For this didactic example, synthetic data was generate from equation 6 for a source of radius R = 0.75 m at depth z_o = 1.5 m. A normal random error of zero mean and a standard deviation 2.5% of the peak anomaly was added to simulate observational data. The inverse problem solved consisted in estimating the parameters R and z_o from the noisy data. The inversion results (figure 3) are: R = 0.75002 ± 0.00071 and z_o = 1.50137 ± 0.0038. The inversion algorithm used is based on Newton method with a Levenberg-Marquardt type damping and requires at each iteration step updated versions of the Hessian and gradient of the objective function and associated constraints. The derivative structures were supplied by a Matlab function that was integrated into the optimizing routine.

6 Summary

The CP method introduced in this paper is a hybrid between finite difference (FD) and true automatic differentiation (AD). Computationally, it is simply FD in the complex plane, but without the step-size dilemma. And, in terms of performance, it is competitive with AD. The examples presented here demonstrate some of the key advantages of the CP technique: namely, implementation simplicity, accuracy, and numerical stability throughout a wide range of step-sizes down to 10^{-50}. Because of these performance features, the method provides a better alternative to ordinary FD schemes as on-the-fly automatic differentiation tool already built into most common computational environments.

7 Acknowledgements

This research work was conducted at King Fahd University of Petroleum and Minerals, (KFUPM), Department of Earth Sciences, Dhahran, Saudi Arabia. Support of KFUPM is gratefully acknowledged. I also thank Professor A. Dogan and Dr. K. Al-Ramadan for reviewing the manuscript and for their valuable suggestions.

8 References

[1] Abokhodair, A. A.: Numerical tools for geoscience computations: Semiautomatic differentiation—SD. Comput. Geosci. 11, 283-296 (2007).

[2] Burg, C. O. E, and Newman, J.C.: Computationally efficient, numerically exact design space derivatives via the complex Taylor's series expansion method. Computers & Fluids 32, 373-383 (2003).

[3] Cerviño, L. I. and Bewley,T. R.: On the extension of the complex-step derivative technique to pseudospectral algorithms, J. Comp. Phys., 187, 544-549 (2003).

[4] Mark, H. and Workman Jr., J.: Derivatives in Spectroscopy: Part I – The Behavior of the derivative, Spectroscopy 18(4), 32-37 (2003).

[5] Martin, J. R. R. A., Kroo, I. M. and Alonso, J. j.: An Automated method for sensitivity analysis using complex variables, Paper AIAA 2001-689 (2001).

[6] Squire, W. and Trapp, G. Using complex variables to estimate derivatives of real functions, SIAM Rev., 40(1), 110-112 (1998).

[7] Vatsa, V.: Computation of sensitivity derivatives of Navier-Stokes equations using complex variables. Advances in Engineering Software 31, 655–659 (2000).

[8] Wang, B. P.: Complex variable method for sensitivity analysis in structural optimization. 4th World Congress on Computational Mechanics, in conjunction with APCOM (2004).

[9] DePauw, D. J. W. and Vanrolleghem, P. A.: Avoiding the finite difference sensitivity analysis deathtrap by using the complex-step derivative approximation technique. 3rd biennial meeting of the International Environmental Modeling and Software Society (iEMSs), Burlington (2006).

[10] Orfanidis, S.J.: Introduction to Signal Processing. Prentice Hall, Englewood Cliffs, NJ, (1996).

[11] Surjanovic, S. & Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets. Retrieved April 5, 2014, from http://www.sfu.ca/~ssurjano.