# GENERALIZATIONS OF THE COMPLEX-STEP DERIVATIVE APPROXIMATION

by

**Kok-Lam Lai**

September 1, 2006

A dissertation submitted to
the Faculty of the Graduate School of
State University of New York at Buffalo
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical & Aerospace Engineering

Major Professor: ______________________ Ph.D.
Prof. John L. Crassidis

# GENERALIZATIONS OF THE COMPLEX-STEP DERIVATIVE APPROXIMATION

by

**Kok-Lam Lai**

September 1, 2006

A dissertation submitted to
the Faculty of the Graduate School of
State University of New York at Buffalo
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical & Aerospace Engineering

*T*o my parents and Pei-Chien

# Acknowledgment

First of all, I would like to express my gratitude to my advisor, Prof. John L. Crassidis, for his intuition and patience that made this work possible.

Thank you Prof. Tarunraj Singh and Prof. D. Joseph Mook for my educations and training in this field and being my committee members. My gratitude also goes to Prof. Peter Scott whom carefully reviewed the manuscript and served as the outside reader.

Large part of my graduate studies were funded by numerous agencies which I deeply appreciate. They are NASA - Goddard Space Flight Center, University at Buffalo and the US Air Force (via StarVision Technologies).

I would like to thank many individuals that impacted my stay here in a positive way, especially Angela, Bart, Bill, Dawn, Diane, Dirk, Eujin, Frank, Jae-Jun, Jong-Woo, Jongrae, Karl, Matthias, Randy, Takis and Yann. I am very grateful for the tremendous impact you had in my life.

None would be possible without the love, understanding and support of my parents Tak Yau and Lee Ah, and my girlfriend Pei-Chien. Thank you for being my all time best friends.

# Contents

# List of Figures

# Abstract

This dissertation presents a general framework for the complex-step derivative approximation to compute numerical derivatives. For first derivatives the complex-step approach does not suffer subtraction cancellation errors as in standard numerical finite-difference approaches. Therefore, since an arbitrarily small step-size can be chosen, the complex-step method can achieve near analytical accuracy. However, for second derivatives straight implementation of the complex-step approach does suffer from roundoff errors. Therefore, an arbitrarily small step-size cannot be chosen. This dissertation expands upon the standard complex-step approach to provide a wider range of accuracy for both the first and second derivative approximations. Higher accuracy formulations can be obtained by repetitively applying the Richardson extrapolations. The new extensions can allow the use of one step-size to provide optimal accuracy for both derivative approximations. Simulation results are provided to show the performance of the new complex-step approximations on a second-order Kalman filter. The new first and second derivatives are also used to generalize the second-order Divided Difference filter to incorporate the complex-step approach.

# Chapter 1

# Introduction

## 1.1 Complex Number

By *complex* it does not mean complexity (nor simplicity to some), but rather numbers with components in the *imaginary* plane. *Imaginary* has nothing to do with the illusionary state of mind, but rather a name coined for the strenuous acceptance of the idea in the early history of complex number. Thus complex number is antonymous to the *real* number, the domain most choose to deal with. A brief history on the interesting discovery and acceptance of complex number can be found in Ref. [1, pg. 1108-1109].

Using complex numbers for computational purposes is often intentionally avoided because of the nonintuitive nature of this domain. However, this perception should not handicap our ability to seek better solutions to the problems associated with traditional (real-valued) problems. Many physical world phenomena actually have their roots in the complex domain [2]. As an aside we note that some interesting historical notes on the discovery and acceptance of the complex variable can also be found in this reference. This dissertation discusses the use of complex numbers for the step size in a Taylor series expansion.

## 1.2 Motivation

Jacobian and Hessian information find applications across a broad spectrum of science, engineering and mathematical analysis. For example, many spatial field modeling approaches are performed with spherical harmonics [3, 4, 5]. Of the most notable are the Earth's magnetic field [6, 7, 8, 9] and gravity field modeling [10, 11]. Geopotential fields can provide crucial information in navigation and filtering. The magnetic field have been successfully used in many modern spacecraft to estimate orbital parameters and attitude [12, 13, 14, 15, 16]. More accurate modeling requires expansion to higher-other terms in the spherical harmonics, which also increases the nonlinearity. The complexity of spherical harmonics functions renders analytical Jacobian and Hessian information impossible to obtain. Therefore they must be obtained numerically. The ability to obtaining more accurate Jacobian and Hessian expressions in the high nonlinearity regions due to higher expansions of the spherical harmonics terms would be useful.

Multidisciplinary design optimization (MDO) has gained much attention in modern complex engineering endeavors [17, 18]. As a result, NASA Langley Research Center published a realistic MDO performance benchmark problem to facilitate development in the field [19]. More efficient optimization algorithms often require gradient and Hessian information to be readily available. This represents another challenge in finding an optimized solution. Again, accurate Jacobian and Hessian information provides better search directions, which in turn provides better performance characteristics in the MDO algorithms.

Numerical finite differences have been widely used as an easy way to obtain derivative approximations. A Taylor series is perhaps the most common method to represent nonlinear functions. The Taylor series is also used in the derivation of the finite-difference derivative approximations. Theoretically, an infinite Taylor series can exactly represent any nonlinear analytical function. Practical implementation, however, has limited it to the available computational resources, thus restricting the number of terms involved. This leads to a truncation of the series after the first few terms. Moreover, deriving analytical expressions using a Taylor series expansion becomes too cumbersome to tackle, particularly when the Taylor series is used as an intermediate step in deriving a higher-order equation in the hierarchy. For instance, using higher-order terms to approximate first or second derivatives with a finite-difference can be analytically difficult if high accuracy is required. This essentially gives rise to a tradeoff between accuracy, performance and analytical complexity.

Taylor series functions are like a "black box" when used to find derivative approximations, which presents a possible improvement point without affecting the desired output. The simple fact that the square of an unity magnitude imaginary number produces a negative real number, $i^2 = -1$, is the fundamental of this work. For example, at 90° or 270° departure from the positive real axis in the imaginary plane, only the imaginary component exists and at 0° and 180° only the real component exists. This dictates that some portions of the analytical analysis come in and out of the complex domain and could be used to put undesired truncation error in Taylor series expansion into the domain that is not related to the outcome, consequently improving the accuracy of the output.

## 1.3 Background and Dissertation Contribution

The importance of accurate and easily available Jacobian and Hessian information does not need to be stressed further. There have been many attempts to improve finite-difference algorithms to obtain Jacobian and Hessian information. One of these new innovations is the ADIFOR [20, 21], an automatic differentiation generation code with Fortran 77. A few drawbacks with this solution include: 1) ADIFOR works only with Fortran; 2) it requires careful programming of the analytical function; and 3) it can only able obtaining Jacobian information. Other automatic differentiation methods include the forward automatic differentiation (FAD), reverse automatic differentiation (RAD) and source-to-source translation (SST) [22]. Finite difference methods are easy to implement, but are prone to both truncation errors for large step sizes and roundoff errors for small step sizes. Automatic differentiation methods, such as FAD, RAD and SST, involve "differentiating the program" that compute the function value. These methods rely on breaking down the complex function into a finite combination of elementary operations. The advantage of these methods is that exact solutions for the Jacobian and Hessian matrices can be found. However, there are significant disadvantages to these methods. In particular, FAD can be inefficient for large-scale problems, RAD requires *significant* effort for its implementation, and SST requires *major* effort for its implementation.

The complex-step derivative approximation can be used to determine first derivatives in a relatively easy way, while providing near analytic accuracy. Early work on obtaining derivatives via a complex-step approximation in order to improve overall accuracy is shown by Lyness and Moler [23], as well as Lyness [2]. Various recent papers reintroduce the complex-step approach to the engineering community [24, 25, 26, 27, 28, 29, 30]. References [31, 32] demonstrate its application to MDO applications. The advantages of the complex-step approximation approach over a standard finite difference include: 1) the Jacobian approximation is not subject to subtractive cancellations inherent in roundoff errors, 2) it can be used on discontinuous functions, and 3) it is easy to implement in a black-box manner, thereby making it applicable to general nonlinear functions.

The complex-step approximation in the aforementioned papers is derived only for first derivatives. A second-order approximation using the complex-step approach is straightforward to derive; however, this approach is subject to roundoff errors for small step-sizes since difference errors arise, as shown by the classic plot in Figure 1.1. As the step size increases

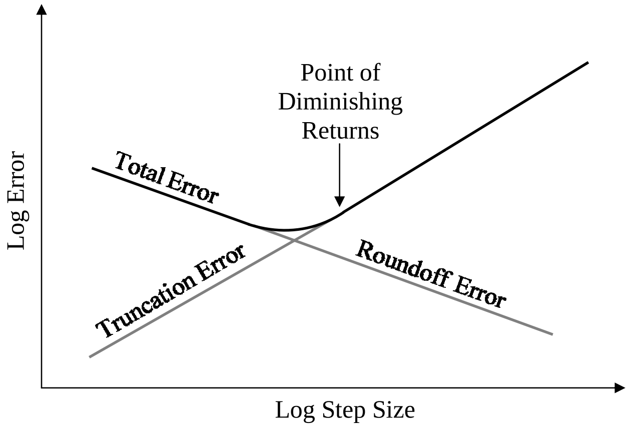


Figure 1.1: Finite Difference Error Versus Step-Size

the accuracy decreases due to truncation errors associated with not adequately approximating the true slope at the point of interest. Decreasing the step size increases the accuracy, but only to an “optimum” point. Any further decrease results in a degradation of the accuracy due to roundoff errors. Hence, a tradeoff between truncation errors and roundoff exists. In fact, through numerous simulations, the complex-step second-derivative approximation is markedly *worse* than a standard finite-difference approach. In this dissertation several extensions of the complex-step approach are derived. These are essentially based on using various complex numbers coupled with Richardson extrapolations [33] to provide further accuracy, instead of the standard purely imaginary approach of the aforementioned papers. As with the standard complex-step approach, all of the new first-derivative approximations are not subject to roundoff errors. However, they all have a wider range of accuracy for larger step sizes than the standard imaginary-only approach. The new second-derivative approximations are more accurate than both the imaginary-only as well as traditional higher-order finite-difference approaches. For example, a new 4-point second-derivative approximation is derived whose accuracy is valid up to tenth-order derivative errors. These new expressions allow a designer to choose one step size in order to provide very accurate approximations, which minimizes the required number of function evaluations. This complex-step derivative approximation (CSDA) is a useful alternative almost everywhere that numerical finite-difference derivative approximations apply. In fact the CSDA is a generalized finite difference that includes complex step sizes, making the CSDA seamlessly applicable to many existing implementations.

A complex-step approach is presented in Ref. [34], which incorporates up to three function variables as quaternion vector components and evaluates the function using quaternion algebra. Using this method, the Jacobian matrix can be obtained with a single function evaluation, which is far less than the one presented here. On the contrary, the method presented in this dissertation has no limitation on number of function variables and does not require programming of quaternion algebra, other than standard complex algebra that is often built into existing mathematical libraries of common programming languages. Consequently, the method presented here is more practical for large multi-variable functions. In addition, the Jacobian matrix is simply a byproduct of the determined Hessian matrix.

This dissertation also extends the first- and second-order CSDA to vector cases to determine the Jacobian and Hessian matrices. This is an attractive way to extend the extended Kalman filter (EKF) to the second-order level to minimize the estimation bias associated with the standard EKF as a result of crude linearization of the nonlinear dynamics. However, like the EKF case, in the second-order Kalman filter (SOKF) the derivative information is obtained at the estimated mean. Unfortunately, the estimated mean often does not match the true mean in presence of high nonlinearity, since derivative information can vary greatly in the vicinity of the true or estimated mean. Interpolation from a Taylor series expansion may yield lower approximation errors of the nonlinear function over the region of interest. The second-order divided difference (DD2) filter [35] relies on this premise and thus often offers significant performance gain and robustness when the estimated mean is far from the truth. In addition, the DD2 filter belongs to a class of filters that linearizes the statistics instead of the nonlinear equations. As a result, Jacobian and Hessian information is not needed. As the DD2 filter was derived from standard first- and second-order finite-difference formulae, this suggests a possible improvement with CSDAs.

## 1.4 Dissertation Outline

As a precursor to the derivations of the CSDA, Chapter 2 reviews the conventional finite-difference derivatives with error analysis. The error analysis include both roundoff (cancellation) error and truncation error. This chapter derives various derivative approximations and presents higher accuracy (less truncation error) by using a Richardson extrapolation [33].

As mentioned before, the nature of complex number will be used as the fundamental theory of this dissertation in application of the Taylor series expansion, thus solving for more accurate first and second derivatives. Chapter 3 derives a general framework for the first and second-order CSDA and even higher accuracy forms using a Richardson approximation. Extension to the vector case will also be presented. A few test cases are presented next to showcase lower numerical error by using the newly derived CSDAs.

Chapter 5 uses the new CSDAs to obtain Jacobian information for a simple EKF, and Jacobian and Hessian information for a SOKF. With deployment of more complicated engineering solutions, the associated underlying dynamics are more complicated and many simple nonlinear filters fail to accurately capture this higher nonlinearity. The widely adopted EKF exhibits significant bias in both the mean and covariance in the face of severe nonlinearity. The second-order Kalman filter offers additional bias correction terms to rectify this problem.

Chapter 4 goes over various forms of state estimation, focusing on nonlinear filtering.

This chapter serves as the precursor to the derivation of the DD2 filter that involves a complex step-size in Chapter 6. Finally, conclusions are drawn in Chapter 7 and some future work is proposed.

# Chapter 2

# Numerical Finite-Difference and Error Analysis

## 2.1 Introduction

This chapter summarizes finite-difference formulae with roundoff error, $E_{\text{round}}(f,h)$, and truncation error, $E_{\text{trunc}}(f,h)$. Focus is mainly placed on first- and second-order derivatives since these most concern our work. Also, concentration will be on a central-difference with abscissas chosen symmetrically from both sides of $x$.

When a nonlinear function is approximated with a Taylor series expansion, most of the time it is difficult to avoid truncation errors. If higher derivatives of a function exist, the Taylor series approximation can be expanded to higher-order to minimize the effects of truncation error. Another way to decrease the effect of truncation error is to use a smaller step size, $h$, however, this would induce another kind of error: roundoff error or subtractive cancellation error. Roundoff error is directly related to the floating point accuracy, which is associated with the limited numerical representation of a computer. The

Table 2.1: IEEE 754 Standard Format Layout for Real Number Representation for Single (32-bit) and Double Precision (64-bit) (the number of bits are shown with bit ranges in square brackets).

| | Sign | Exponent | Significant |
|---|---|---|---|
| Single Precision | 1 [31] | 8 [30-23] | 23 [22-00] |
| Double Precision | 1 [63] | 11 [62-52] | 52 [51-00] |

higher precision capability a computer possess, the lower the roundoff error. Floating point is most commonly represented by the IEEE 754 Standard. In a 64-bit computing environment (or "double" precision in a 32-bit computer) with IEEE 754 Standard, there are 52 significant (also known as mantissa) bits to represent a number. Thus the floating point accuracy (the numerical distance from 1.0 to the next closest floating point number) with this standard is $\varepsilon = \frac{1}{2^{52}} = 2.220446049250313 \times 10^{-016}$. The real number storage layout for the IEEE 754 standard is shown in Table 2.1.

In actual implementation of numerical finite-difference equations, unless an optimal step size is found and used, oftentimes only about half the highest accuracy of the computer capability is retained [36, pg. 343]. This stresses the importance of obtaining an optimal step size whenever possible. Furthermore, accuracy degeneration emerges when dealing with experimental data that often truncates to a few digits. To counter this loss in accuracy, the experimental data could first be run through a smoother or a simple least-squares algorithm before carrying out the numerical differentiation.

## 2.2 Taylor Series Expansions

This section shows various Taylor series expansions at different step sizes that are useful for deriving finite-difference derivatives shown later. For a given function $f(x)$ and its $n^{\text{th}}$ derivative $f^{(n)}$, these are given by

$$\begin{aligned} f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) + \frac{h^5}{5!}f^{(5)}(x) + \frac{h^6}{6!}f^{(6)}(x) \\ + \frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) + \frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{2.1a}$$

$$\begin{aligned} f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) - \frac{h^5}{5!}f^{(5)}(x) + \frac{h^6}{6!}f^{(6)}(x) \\ - \frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) - \frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{2.1b}$$

$$\begin{aligned} f(x+2h) = f(x) + 2hf'(x) + \frac{4h^2}{2}f''(x) + \frac{8h^3}{3!}f^{(3)}(x) + \frac{16h^4}{4!}f^{(4)}(x) + \frac{32h^5}{5!}f^{(5)}(x) + \frac{64h^6}{6!}f^{(6)}(x) \\ + \frac{128h^7}{7!}f^{(7)}(x) + \frac{256h^8}{8!}f^{(8)}(x) + \frac{512h^9}{9!}f^{(9)}(x) \end{aligned} \tag{2.1c}$$

$$f(x-2h) = f(x) - 2hf'(x) + \frac{4h^2}{2}f''(x) - \frac{8h^3}{3!}f^{(3)}(x) + \frac{16h^4}{4!}f^{(4)}(x) - \frac{32h^5}{5!}f^{(5)}(x) + \frac{64h^6}{6!}f^{(6)}(x)$$

$$- \frac{128h^7}{7!} f^{(7)}(x) + \frac{256h^8}{8!} f^{(8)}(x) - \frac{512h^9}{9!} f^{(9)}(x) \quad (2.1\text{d})$$

$$\begin{aligned} f(x+3h) = f(x) + 3hf'(x) + \frac{9h^2}{2} f''(x) + \frac{27h^3}{3!} f^{(3)}(x) + \frac{81h^4}{4!} f^{(4)}(x) + \frac{243h^5}{5!} f^{(5)}(x) + \frac{729h^6}{6!} f^{(6)}(x) \\ + \frac{2187h^7}{7!} f^{(7)}(x) + \frac{6561h^8}{8!} f^{(8)}(x) + \frac{19683h^9}{9!} f^{(9)}(x) \end{aligned} \quad (2.1\text{e})$$

$$\begin{aligned} f(x-3h) = f(x) - 3hf'(x) + \frac{9h^2}{2} f''(x) - \frac{27h^3}{3!} f^{(3)}(x) + \frac{81h^4}{4!} f^{(4)}(x) - \frac{243h^5}{5!} f^{(5)}(x) + \frac{729h^6}{6!} f^{(6)}(x) \\ - \frac{2187h^7}{7!} f^{(7)}(x) + \frac{6561h^8}{8!} f^{(8)}(x) - \frac{19683h^9}{9!} f^{(9)}(x) \end{aligned} \quad (2.1\text{f})$$

$$\begin{aligned} f(x+4h) = f(x) + 4hf'(x) + \frac{16h^2}{2} f''(x) + \frac{64h^3}{3!} f^{(3)}(x) + \frac{256h^4}{4!} f^{(4)}(x) + \frac{1024h^5}{5!} f^{(5)}(x) + \frac{4096h^6}{6!} f^{(6)}(x) \\ + \frac{16384h^7}{7!} f^{(7)}(x) + \frac{65536h^8}{8!} f^{(8)}(x) + \frac{262144h^9}{9!} f^{(9)}(x) \end{aligned} \quad (2.1\text{g})$$

$$\begin{aligned} f(x-4h) = f(x) - 4hf'(x) + \frac{16h^2}{2} f''(x) - \frac{64h^3}{3!} f^{(3)}(x) + \frac{256h^4}{4!} f^{(4)}(x) - \frac{1024h^5}{5!} f^{(5)}(x) + \frac{4096h^6}{6!} f^{(6)}(x) \\ - \frac{16384h^7}{7!} f^{(7)}(x) + \frac{65536h^8}{8!} f^{(8)}(x) - \frac{262144h^9}{9!} f^{(9)}(x) \end{aligned} \quad (2.1\text{h})$$

$$\begin{aligned} f(x+5h) = f(x) + 5hf'(x) + \frac{25h^2}{2}f''(x) + \frac{125h^3}{3!}f^{(3)}(x) + \frac{625h^4}{4!}f^{(4)}(x) + \frac{3125h^5}{5!}f^{(5)}(x) + \frac{15625h^6}{6!}f^{(6)}(x) \\ + \frac{78125h^7}{7!}f^{(7)}(x) + \frac{390625h^8}{8!}f^{(8)}(x) + \frac{1953125h^9}{9!}f^{(9)}(x) \end{aligned} \tag{2.1i}$$

$$\begin{aligned} f(x-5h) = f(x) - 5hf'(x) + \frac{25h^2}{2}f''(x) - \frac{125h^3}{3!}f^{(3)}(x) + \frac{625h^4}{4!}f^{(4)}(x) - \frac{3125h^5}{5!}f^{(5)}(x) + \frac{15625h^6}{6!}f^{(6)}(x) \\ - \frac{78125h^7}{7!}f^{(7)}(x) + \frac{390625h^8}{8!}f^{(8)}(x) - \frac{1953125h^9}{9!}f^{(9)}(x) \end{aligned} \tag{2.1j}$$

The addition and subtraction pairs for the series above are

$$f(x+h) + f(x-h) = 2f(x) + h^2f''(x) + \frac{2h^4}{4!}f^{(4)}(x) + \frac{2h^6}{6!}f^{(6)}(x) + \frac{2h^8}{8!}f^{(8)}(x) \tag{2.2a}$$

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{2h^3}{3!}f^{(3)}(x) + \frac{2h^5}{5!}f^{(5)}(x) + \frac{2h^7}{7!}f^{(7)}(x) + \frac{2h^9}{9!}f^{(9)}(x) \tag{2.2b}$$

$$f(x+2h) + f(x-2h) = 2f(x) + \frac{8h^2}{2}f''(x) + \frac{32h^4}{4!}f^{(4)}(x) + \frac{128h^6}{6!}f^{(6)}(x) + \frac{512h^8}{8!}f^{(8)}(x) \tag{2.2c}$$

$$f(x+2h) - f(x-2h) = 4hf'(x) + \frac{16h^3}{3!}f^{(3)}(x) + \frac{64h^5}{5!}f^{(5)}(x) + \frac{256h^7}{7!}f^{(7)}(x) + \frac{1024h^9}{9!}f^{(9)}(x) \tag{2.2d}$$

$$f(x+3h) + f(x-3h) = 2f(x) + \frac{18h^2}{2}f''(x) + \frac{162h^4}{4!}f^{(4)}(x) + \frac{1458h^6}{6!}f^{(6)}(x) + \frac{13122h^8}{8!}f^{(8)}(x) \tag{2.2e}$$

$$f(x+3h) - f(x-3h) = 6hf'(x) + \frac{54h^3}{3!}f^{(3)}(x) + \frac{486h^5}{5!}f^{(5)}(x) + \frac{4374h^7}{7!}f^{(7)}(x) + \frac{39366h^9}{9!}f^{(9)}(x) \tag{2.2f}$$

$$f(x+4h) + f(x-4h) = 2f(x) + \frac{32h^2}{2}f''(x) + \frac{512h^4}{4!}f^{(4)}(x) + \frac{8192h^6}{6!}f^{(6)}(x) + \frac{131072h^8}{8!}f^{(8)}(x) \tag{2.2g}$$

$$f(x+4h) - f(x-4h) = 8hf'(x) + \frac{128h^3}{3!}f^{(3)}(x) + \frac{2048h^5}{5!}f^{(5)}(x) + \frac{32768h^7}{7!}f^{(7)}(x) + \frac{524288h^9}{9!}f^{(9)}(x) \tag{2.2h}$$

$$f(x+5h)+f(x-5h)=2f(x)+25h^2f''(x)+\frac{625h^4}{12}f^{(4)}(x)+\frac{3125h^6}{72}f^{(6)}(x)+\frac{78125h^8}{4032}f^{(8)}(x) \tag{2.2i}$$

$$f(x+5h)-f(x-5h)=10hf'(x)+\frac{125h^3}{3}f^{(3)}(x)+\frac{625h^5}{12}f^{(5)}(x)+\frac{15625h^7}{504}f^{(7)}(x)+\frac{390625h^9}{36288}f^{(9)}(x) \tag{2.2j}$$

## 2.3 Sample Derivations with Error Analysis

### 2.3.1 $f''(x_0) \approx \frac{f_1-2f_0+f_{-1}}{h^2}$, $\boldsymbol{O}(h^2)$

Assume that $f \in C^3[a,b]$ and that $x-h,\ x,\ x+h \in [a,b]$. From Eqs. (2.1a) and (2.1b)

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f^{(3)}(x) + \frac{h^4}{24}f^{(4)}(c_1) \tag{2.3a}$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f^{(3)}(x) + \frac{h^4}{24}f^{(4)}(c_2) \tag{2.3b}$$

For $O(h^2)$, the terms $f'$ or $f''$ need to be removed, so Eq. (2.3b) and Eq. (2.3a) are added together, and by abbreviating $f(x+nh)$ with $f_n$ we obtain

$$f_1 + f_{-1} = 2f_0 + h^2 f''(x_0) + \frac{h^4}{24}\Big[f^{(4)}(c_1) + f^{(4)}(c_2)\Big] \tag{2.4}$$

Solving for $f''(x_0)$ gives

$$f''(x_0) = \frac{f_1 - 2f_0 + f_{-1}}{h^2} - \frac{h^2}{24}\Big[f^{(4)}(c_1) + f^{(4)}(c_2)\Big] \tag{2.5}$$

Assuming $f^{(4)}$ has one sign with relatively constant magnitude, then a value $c = c(x) \in [x-h, x+h]$ can be found so that

$$f^{(4)}(c_1) + f^{(4)}(c_2) = 2f^{(4)}(c) \tag{2.6}$$

Substituting this into Eq. (2.5) gives

$$f''(x_0) = \frac{f_1 - 2f_0 + f_{-1}}{h^2} - \frac{h^2}{12}f^{(4)}(c) \tag{2.7}$$

Thus the truncation error is

$$E_{\text{trunc}}(f,h) = -\frac{h^2}{12} f^{(4)}(c) \tag{2.8}$$

✠

### 2.3.2 $f''(x_0) = \frac{-f_2+16f_1-30f_0+16f_{-1}-f_{-2}}{12h^2}$, $\boldsymbol{O}(h^4)$

Assume that $f \in C^5[a,b]$ and that $x-2h$, $x-h$, $x$, $x+h$, $x+2h \in [a,b]$. From Eqs. (2.2a) and (2.2c)

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \frac{h^4}{12} f^{(4)}(x) + \frac{h^6}{360} f^{(6)}(c_1) \tag{2.9a}$$

$$f(x+2h) + f(x-2h) = 2f(x) + 4h^2 f''(x) + \frac{4h^4}{3} f^{(4)}(x) + \frac{8h^6}{45} f^{(6)}(c_2) \tag{2.9b}$$

For $O(h^4)$, the terms $f^{(3)}$ or $f^{(4)}$ need to be removed, so Eq. (2.9a) is multiplied by 16. Subtracting Eq. (2.9b) from it, and by abbreviating $f(x+nh)$ with $f_n$ we obtain

$$-f_2 + 16f_1 + 16f_{-1} - f_{-2} = 30f_0 + 12h^2 f''(x_0) + \frac{h^6}{45}\Big[2f^{(6)}(c_1) - 8f^{(6)}(c_2)\Big] \tag{2.10}$$

Solving for $f''(x_0)$ gives

$$f''(x_0) = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} - \frac{h^4}{270}\Big[f^{(6)}(c_1) - 4f^{(6)}(c_2)\Big] \tag{2.11}$$

Assuming $f^{(6)}$ has one sign with relatively constant magnitude, then a value $c \in [x-2h, x+2h]$ can be found so that

$$f^{(6)}(c_1) - 4f^{(6)}(c_2) = -3f^{(6)}(c) \tag{2.12}$$

Substituting this into Eq. (2.11) gives

$$f''(x_0) = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + \frac{h^4}{90}f^{(6)}(c) \tag{2.13}$$

Thus the truncation error is

$$E_{\text{trunc}}(f, h) = \frac{h^4}{90}f^{(6)}(c) \tag{2.14}$$

✠

### 2.3.3 $f'(x_0) \approx \frac{-f_{-3}+9f_{-2}-45f_{-1}+45f_1-9f_2+f_3}{60h}$, $\boldsymbol{O}(h^6)$

Assume that $f \in C^7[a, b]$ and that $x - 3h$, $x - 2h$, $x - h$, $x$, $x + h$, $x + 2h$, $x + 3h \in [a, b]$. From Eqs. (2.2b), (2.2d) and (2.2f)

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3}f^{(3)}(x) + \frac{h^5}{60}f^{(5)}(x) + \frac{h^7}{2520}f^{(7)}(c_1) \tag{2.15a}$$

$$f(x+2h) - f(x-2h) = 4hf'(x) + \frac{8h^3}{3}f^{(3)}(x) + \frac{8h^5}{15}f^{(5)}(x) + \frac{16h^7}{315}f^{(7)}(c_2) \tag{2.15b}$$

$$f(x+3h) - f(x-3h) = 6hf'(x) + 9h^3f^{(3)}(x) + \frac{81h^5}{20}f^{(5)}(x) + \frac{243h^7}{280}f^{(7)}(c_3) \tag{2.15c}$$

For $O(h^6)$, the terms $f^{(5)}$ or $f^{(6)}$ need to be removed, so Eq. (2.15a) is multiplied by 45 and Eq. (2.15b) is multiplied by $-9$ to obtain

$$45f(x+h) - 45f(x-h) = 90hf'(x) + 15h^3f^{(3)}(x) + \frac{3h^5}{4}f^{(5)}(x) + \frac{h^7}{56}f^{(7)}(c_1) \tag{2.16a}$$

$$-9f(x+2h) + 9f(x-2h) = -36hf'(x) - 24h^3f^{(3)}(x) - \frac{24h^5}{5}f^{(5)}(x) - \frac{16h^7}{35}f^{(7)}(c_2) \tag{2.16b}$$

$$f(x+3h) - f(x-3h) = 6hf'(x) + 9h^3f^{(3)}(x) + \frac{81h^5}{20}f^{(5)}(x) + \frac{243h^7}{280}f^{(7)}(c_3) \tag{2.16c}$$

Adding them together, and by abbreviating $f(x+nh)$ with $f_n$ we obtain

$$\begin{aligned} 45f_1 - 45f_{-1} - 9f_2 + 9f_{-2} + f_3 - f_{-3} = 60hf'(x_0)& \\ + \frac{h^7}{280}\Big[5f^{(7)}(c_1) - 128f^{(7)}(c_2) + 243f^{(7)}(c_3)\Big]& \end{aligned} \tag{2.17}$$

Solving for $f'(x_0)$ gives

$$\begin{aligned} f'(x_0) = \frac{-f_{-3} + 9f_{-2} - 45f_{-1} + 45f_1 - 9f_2 + f_3}{60h}& \\ - \frac{h^6}{16800}\Big[5f^{(7)}(c_1) - 128f^{(7)}(c_2) + 243f^{(7)}(c_3)\Big]& \end{aligned} \tag{2.18}$$

Assuming $f^{(7)}$ has one sign with relatively constant magnitude around the vicinity of $x_0$, then a value $c \in [x-3h, x+3h]$ can be found so that

$$5f^{(7)}(c_1) - 128f^{(7)}(c_2) + 243f^{(7)}(c_3) = 120f^{(7)}(c) \tag{2.19}$$

Substituting this into Eq. (2.18) gives

$$f'(x_0) = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} - \frac{h^6}{140}f^{(7)}(c) \tag{2.20}$$

Thus the truncation error is

$$E_{\text{trunc}}(f,h) = -\frac{h^6}{140}f^{(7)}(c) \tag{2.21}$$

✠

### 2.3.4 $f''(x_0) \approx \frac{2f_{-3}-27f_{-2}+270f_{-1}+490f_0+270f_1-27f_2+2f_3}{180h^2}$, $\boldsymbol{O}(h^6)$

Assume that $f \in C^7[a,b]$ and that $x-3h,\ x-2h,\ x-h,\ x,\ x+h,\ x+2h,\ x+3h \in [a,b]$. From Eqs. (2.2a), (2.2c) and (2.2e)

$$f(x+h)+f(x-h) = 2f(x)+h^2f''(x)+\frac{h^4}{12}f^{(4)}(x) + \frac{h^6}{360}f^{(6)}(x)+\frac{h^8}{20160}f^{(8)}(c_1) \quad (2.22a)$$

$$f(x+2h)+f(x-2h) = 2f(x)+4h^2f''(x)+\frac{4h^4}{3}f^{(4)}(x) + \frac{8h^6}{45}f^{(6)}(x)+\frac{4h^8}{315}f^{(8)}(c_2) \quad (2.22b)$$

$$f(x+3h)+f(x-3h) = 2f(x)+9h^2f''(x)+\frac{27h^4}{4}f^{(4)}(x) + \frac{81h^6}{40}f^{(6)}(x)+\frac{729h^8}{2240}f^{(8)}(c_3) \quad (2.22c)$$

For $O(h^6)$, the terms $f^{(5)}$ or $f^{(6)}$ need to be removed, so Eq. (2.15a) is multiplied by 270, Eq. (2.15b) is multiplied by $-27$ and Eq. (2.15c) is multiplied by 2 to obtain

$$270f(x+h)+270f(x-h) = 540f(x)+270h^2f''(x)+\frac{45h^4}{2}f^{(4)}(x) + \frac{3h^6}{4}f^{(6)}(x)+\frac{3h^8}{224}f^{(8)}(c_1) \quad (2.23a)$$

$$-27f(x+2h)-27f(x-2h) = -54f(x)-108h^2f''(x)-36h^4f^{(4)}(x) - \frac{24h^6}{5}f^{(6)}(x)-\frac{12h^8}{35}f^{(8)}(c_2) \quad (2.23b)$$

$$2f(x+3h)+2f(x-3h)=4f(x)+18h^2f''(x)+\frac{27h^4}{2}f^{(4)}(x)$$
$$+\frac{81h^6}{20}f^{(6)}(x)+\frac{729h^8}{1120}f^{(8)}(c_3) \quad (2.23c)$$

adding them together, and by abbreviating $f(x+nh)$ with $f_n$ we obtain

$$270f_1+270f_{-1}-27f_2-27f_{-2}+2f_3+2f_{-3}=490f_0+180h^2f''(x_0)$$
$$+\frac{h^8}{1120}\Big[15f^{(8)}(c_1)-384f^{(8)}(c_2)+729f^{(8)}(c_3)\Big] \quad (2.24)$$

Solving for $f''(x_0)$ gives

$$f''(x_0)=\frac{2f_{-3}-27f_{-2}+270f_{-1}-490f_0+270f_1-27f_2+2f_3}{180h^2}$$
$$-\frac{h^6}{201600}\Big[15f^{(8)}(c_1)-384f^{(8)}(c_2)+729f^{(8)}(c_3)\Big] \quad (2.25)$$

Assuming $f^{(6)}$ has one sign with relatively constant magnitude, then a value $c\in[x-3h,x+3h]$ can be found so that

$$15f^{(8)}(c_1)-384f^{(8)}(c_2)+729f^{(8)}(c_3)=360f^{(6)}(c) \quad (2.26)$$

Substituting this into Eq. (2.25) gives

$$f''(x_0)=\frac{2f_{-3}-27f_{-2}+270f_{-1}-490f_0+270f_1-27f_2+2f_3}{180h^2}-\frac{h^6}{560}f^{(8)}(c) \quad (2.27)$$

Thus, the truncation error is

$$E_{\text{trunc}}(f,h)=-\frac{h^6}{560}f^{(8)}(c) \quad (2.28)$$

✠

### 2.3.5 Roundoff Error

Let's represent a function evaluation as

$$f(x_0 + nh) = y_n + e_n \tag{2.29}$$

where $y_n$ is the approximation of the function evaluation with $e_n$ denotes the associated roundoff error, for example, $f(x_0 - h) = y_{-1} + e_{-1}$. Taking the 6th-order central finite-difference for the first derivative, Eq. (2.20), as an example gives

$$f'(x_0) = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + E(f,h) \tag{2.30}$$

where the total error term is

$$\begin{aligned} E(f,h) &= E_{\text{round}}(f,h) + E_{\text{trunc}}(f,h) \\ &= \frac{-e_2 + 16e_1 - 30e_0 + 16e_{-1} - e_{-2}}{12h^2} - \frac{h^6}{140} f^{(7)}(c) \end{aligned} \tag{2.31}$$

We can assume that $|e_n| \leq \varepsilon$ for all $n$, and thus the roundoff error is

$$E_{\text{round}}(f,h) = \frac{16\varepsilon}{3h^2} \tag{2.32}$$

Now the total error terms for $f'(x_0)$, $O(h^6)$, are

$$E(f,h) = \frac{16\varepsilon}{3h^2} - \frac{h^6}{140} f^{(7)}(c) \tag{2.33}$$

Table 2.2: Central-Difference Formulae

| Central-Difference | Order | $E_{\text{round}}(f,h)$ | $E_{\text{trunc}}(f,h)$ |
|---|---|---|---|
| $f'(x_0)=\frac{f_1-f_{-1}}{2h}$ | $O(h^2)$ | $\frac{\varepsilon}{h}$ | $-\frac{h^2}{6}f^{(3)}(c)$ |
| $f''(x_0)=\frac{f_1-2f_0+f_{-1}}{h^2}$ | $O(h^2)$ | $\frac{4\varepsilon}{h^2}$ | $-\frac{h^2}{12}f^{(4)}(c)$ |
| $f'(x_0)=\frac{-f_2+8f_1-8f_{-1}+f_{-2}}{12h}$ | $O(h^4)$ | $\frac{2\varepsilon}{3h}$ | $\frac{h^4}{30}f^{(5)}(c)$ |
| $f''(x_0)=\frac{-f_2+16f_1-30f_0+16f_{-1}-f_{-2}}{12h^2}$ | $O(h^4)$ | $\frac{16\varepsilon}{3h^2}$ | $\frac{h^4}{90}f^{(6)}(c)$ |
| $f'(x_0)=\frac{-f_{-3}+9f_{-2}-45f_{-1}+45f_1-9f_2+f_3}{60h}$ | $O(h^6)$ | $\frac{11\varepsilon}{6h}$ | $-\frac{h^6}{140}f^{(7)}(c)$ |
| $f''(x_0)=\frac{2f_{-3}-27f_{-2}+270f_{-1}+490f_0+270f_1-27f_2+2f_3}{180h^2}$ | $O(h^6)$ | $\frac{272\varepsilon}{45h^2}$ | $-\frac{h^6}{560}f^{(8)}(c)$ |

✠

This approach can be applied for other finite-difference formulae as well. The first and second order central finite-difference derivatives with roundoff and truncation errors are given in Table 2.2.

## 2.4 Richardson Extrapolation

Given a derivative approximation of order, let's say, $O(h^k)$, can we do better? Another way to derive higher-order derivative approximations is by linearly combining lower-order approximations, which is called Richardson extrapolation [33]. Let's represent the first derivative approximation with step size $h$ as

$$F(x_0) = D(h) + \underbrace{c_1h^{k_1} + c_2h^{k_2} + c_3h^{k_3} + \cdots}_{E_{\text{trunc}}(h)} \tag{2.34}$$

where $F(x_0)$ is the derivative approximation ($f'(x_0)$ or higher) and $D(h)$ is the finite-difference derivative approximation. Notice that $k_1$ does not necessarily equal 1 or 2, and $k_n$ does not necessarily equal $k_{n-1}+1$. Equation (2.34) becomes a different equation with different $h$; in other words, we can obtain another equation by simply using another value for $h$. The new $F(x_0)$ will be more accurate than before.

Keeping only the first-error term and using $h$ and $h/q$ as step sizes, where $q$ is greater than 1, gives

$$F(x_0) = D(h) + c_1(h)^{k_1} \tag{2.35a}$$

$$F(x_0) = D(h/q) + c_1(h/q)^{k_1} \tag{2.35b}$$

Multiplying Eq. (2.35b) by $q^{k_1}$ and subtracting it from Eq. (2.35a) yields

$$(q^{k_1} - 1)F(x_0) = q^{k_1} D(h/q) - D(h) \tag{2.36}$$

After rearranging,

$$F(x_0) = \frac{q^{k_1} D(h/q) - D(h)}{q^{k_1} - 1} \tag{2.37}$$

with error $O(h^{k_2})$. When $k_1 = 1$, we have $F(x_0) = \frac{1}{q-1}[qD(h/q) - D(h)]$; it is easy to see from here that a heavier weight is put on the approximation with smaller step size, i.e. it is more accurate. This can be expanded to use three different step sizes, $h_1$, $h_2$ and $h_3$, and the truncation error of this new combination will be higher than $O(h_1^{k_3})$, $O(h_2^{k_3})$ or $O(h_3^{k_3})$.

A recursive form can be derived to systematically obtain higher-order derivative approximations. Let the first column be

$$D_{\alpha,1} = D(h/q^{\alpha-1}) \quad \text{for } \alpha = 1, \ldots, n \tag{2.38}$$

and other elements as

$$D_{\alpha,\beta} = \frac{q^{k_{\beta-1}} D_{\alpha,\beta-1} - D_{\alpha-1,\beta-1}}{q^{k_{\beta-1}} - 1} \quad \text{for } \beta = 2, \ldots, n \tag{2.39}$$

Then a higher precision approximation can be found by filling up a lower triangle matrix:

$$\begin{array}{lllll}
D_{1,1} & & & & \\
D_{2,1} & D_{2,2} = \frac{q^{k_1} D_{2,1} - D_{1,1}}{q^{k_1} - 1} & & & \\
D_{3,1} & D_{3,2} = \frac{q^{k_1} D_{3,1} - D_{2,1}}{q^{k_1} - 1} & D_{3,3} = \frac{q^{k_2} D_{3,2} - D_{2,2}}{q^{k_2} - 1} & & \\
\vdots & \vdots & \vdots & \ddots & \\
D_{n,1} & D_{n,2} = \frac{q^{k_1} D_{n,1} - D_{n-1,1}}{q^{k_1} - 1} & D_{n,3} = \frac{q^{k_2} D_{n,2} - D_{n-1,2}}{q^{k_2} - 1} & \cdots & D_{n,n} = \frac{q^{k_{n-1}} D_{n,n-1} - D_{n-1,n-1}}{q^{k_{n-1}} - 1}
\end{array}$$

where the last element, $D_{n,n}$, is the most accurate approximation with error $O(h^{k_n})$.

## 2.5 Sample Derivations with Richardson Approximation

### 2.5.1 $f'(x_0)$, from $O(h^2)$ to $O(h^4)$

To obtain the truncation error as well, we need to expand $f'(x_0)$ from $O(h^2)$ to higher-order,

$$f'(x_0) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f^{(3)}(x) - \frac{h^4}{120} f^{(5)}(x) - \frac{h^6}{5040} f^{(7)}(x) + \cdots \tag{2.40}$$

Thus for this case, $D = \frac{f(x+h)-f(x-h)}{2h} - \frac{h^2}{6} f^{(3)}(x)$ and we wish to eliminate $O(h^2)$ error terms (so $k = 2$) and improve it to the next level, $O(h^4)$. Setting $h \to 2h$ and $q = 2$, using similar notation as before, gives

$$D(h/q \to h) = \frac{f_1 - f_{-1}}{2h} - \frac{h^2}{6} f^{(3)}(c_{2,1}) - \frac{h^4}{120} f^{(5)}(c_{4,1}) \tag{2.41a}$$

$$D(h \to 2h) = \frac{f_2 - f_{-2}}{4h} - \frac{4h^2}{6} f^{(3)}(c_{2,2}) - \frac{16h^4}{120} f^{(5)}(c_{4,2}) \tag{2.41b}$$

Again, it is assumed that $f^{(3)}$ does not change much in the interval of $h$ and $2h$ and thus we can assume $f^{(3)}(c_{2,1}) = f^{(3)}(c_{2,2}) = f^{(3)}(c_2)$ and similarly for $f^{(5)}$. The new approximation using Eq. (2.37) would be ($O(h^2)$ error terms will be omitted since it is known they will cancel out)

$$\begin{aligned} f'(x_0) &= \frac{2^2 \frac{f_1 - f_{-1}}{2h} - \frac{f_2 - f_{-2}}{4h}}{2^2 - 1} + \frac{2^2 \left[ -\frac{h^4}{120} f^{(5)}(c_4) \right] + \frac{2h^4}{15} f^{(5)}(c_4)}{2^2 - 1} \\ &= \underbrace{\frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h}}_{\text{new approximation}} + \underbrace{\frac{h^4}{30} f^{(5)}(c_4)}_{O(h^4)} \end{aligned} \tag{2.42}$$

✠

### 2.5.2 $f''(x_0)$, from $O(h^2)$ to $O(h^4)$

To obtain the truncation error as well, we need to expand $f''(x_0)$ from $O(h^2)$ to higher-order,

$$f''(x_0) = \frac{f_1 - 2f_0 + f_{-1}}{h^2} - \frac{h^2}{12} f^{(4)}(x) - \frac{h^4}{360} f^{(6)}(x) - \frac{h^6}{20160} f^{(8)}(x) + \cdots \tag{2.43}$$

Thus for this case, $D = \frac{f_1 - 2f_0 + f_{-1}}{h^2} - \frac{h^2}{12} f^{(4)}(x)$ and we wish to eliminate $O(h^2)$ error terms (so $k = 2$) and improve it to the next level, $O(h^4)$. Setting $h \rightarrow 2h$ and $q = 2$, using similar notation as before, gives

$$D(h/q \rightarrow h) = \frac{f_1 - 2f_0 + f_{-1}}{h^2} - \frac{h^2}{12} f^{(4)}(c_{2,1}) - \frac{h^4}{360} f^{(6)}(c_{4,1}) \tag{2.44a}$$

$$D(h \rightarrow 2h) = \frac{f_2 - 2f_0 + f_{-2}}{4h^2} - \frac{4h^2}{12} f^{(4)}(c_{2,2}) - \frac{16h^4}{360} f^{(6)}(c_{4,2}) \tag{2.44b}$$

Again, it is assumed that $f^{(4)}$ does not change much in the interval of $h$ and $2h$ and thus we can assume $f^{(4)}(c_{2,1}) = f^{(4)}(c_{2,2}) = f^{(4)}(c_2)$ and similarly for $f^{(6)}$. The new approximation using Eq. (2.37) would be ($O(h^2)$ error terms will be omitted since it is known they will cancel out)

$$\begin{aligned} f''(x_0) &= \frac{2^2 \frac{f_1-2f_0+f_{-1}}{h^2} - \frac{f_2-2f_0+f_{-2}}{4h^2}}{2^2-1} + \frac{2^2\left[-\frac{h^4}{360}f^{(6)}(c_4)\right] + \frac{16h^4}{360}f^{(6)}(c_4)}{2^2-1} \\ &= \underbrace{\frac{-f_2+16f_1-30f_0+16f_{-1}-f_{-2}}{12h^2}}_{\text{new approximation}} + \underbrace{\frac{h^4}{90}f^{(6)}(c_4)}_{O(h^4)} \end{aligned} \qquad (2.45)$$

which is similar to the results from §2.3.2.

✠

## 2.6 Optimal Step-size

Again, the total error terms for the 6th-order central finite-difference first derivative Eq. (2.33)is taken for the sample analysis. Using the notation $M = \max_{a \le c \le b}\{|f^{(7)}(c)|\}$ gives

$$|E(f,h)| \le \frac{16\varepsilon}{3h^2} + \frac{Mh^6}{140} \qquad (2.46)$$

Taking the derivative with respect to $h$ and setting it to zero gives

$$\frac{d}{dh}\left[\frac{16\varepsilon}{3h^2} + \frac{Mh^6}{140}\right] = -\frac{32\varepsilon}{3h^3} + \frac{3Mh^5}{70} = 0 \qquad (2.47)$$

and solving for $h$ yields

$$h = \left(\frac{2240\varepsilon}{3M}\right)^{1/3} \qquad (2.48)$$

Similarly, this applies to other finite-difference formulae as well.

Let us examine an example where $M = 1$ (for example a transcendental function) and $\varepsilon = 2.220446049250313 \times 10^{-016}$ (64-bit precision or double precision in a 32-bit computing

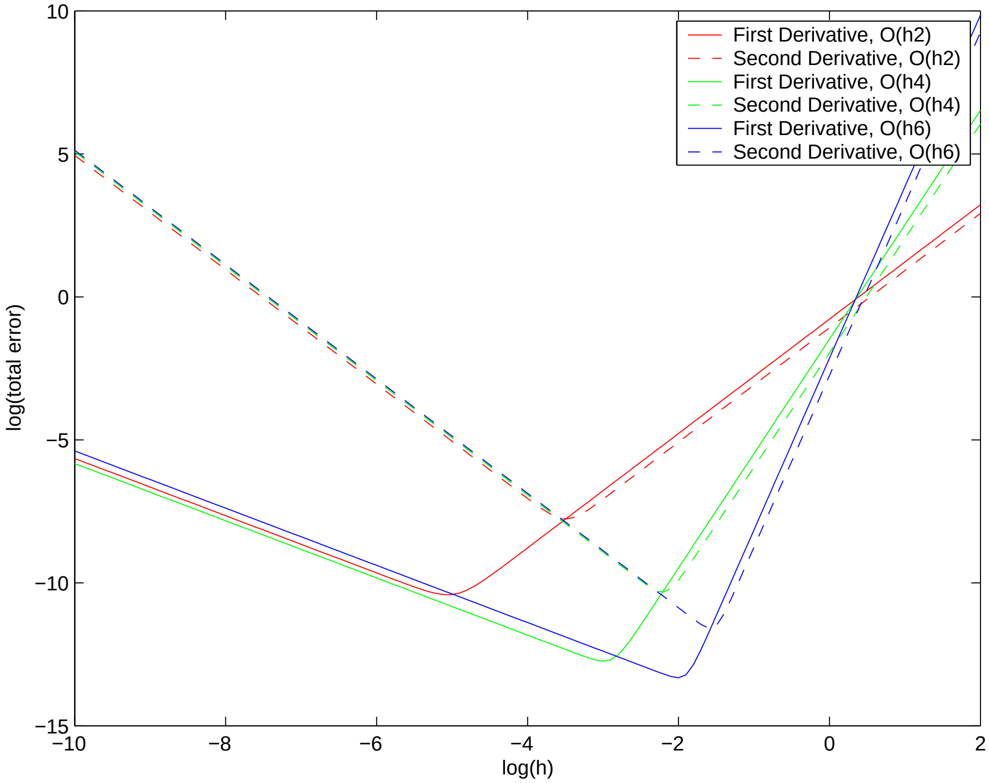


Figure 2.1: Roundoff Error vs. Truncation Error

environment). Figure 2.1 shows the total error of the first and second finite-difference derivatives of various orders. It is obvious that the higher-order approximations have a steeper truncation error slope, in other words, when the step size becomes smaller, they approach the true value faster. Also notice that the higher-order approximations have a lower total error at the optimal step size and that the optimal step sizes are larger than the lower-order approximations too.

Table 2.3: Central-Difference Formulae for Higher Derivatives

| Central-Difference | Order |
|---|---|
| $f^{(3)}(x_0) = \frac{f_2 - 2f_1 + 2f_{-1} - f_{-2}}{2h^3}$ | $O(h^2)$ |
| $f^{(4)}(x_0) = \frac{f_2 - 4f_1 + 6f_0 - 4f_{-1} + f_{-2}}{h^4}$ | $O(h^2)$ |
| $f^{(3)}(x_0) = \frac{-f_3 + 8f_2 - 13f_1 + 13f_{-1} - 8f_{-2} + f_{-3}}{8h^3}$ | $O(h^4)$ |
| $f^{(4)}(x_0) = \frac{-f_3 + 12f_2 - 39f_1 + 56f_0 - 39f_{-1} + 12f_{-2} - f_{-3}}{6h^4}$ | $O(h^4)$ |

Table 2.4: Forward-Difference Formulae of Order $O(h^2)$

| Forward-Difference |
|---|
| $f'(x_0) = \frac{-3f_0 + 4f_1 - f_2}{2h}$ |
| $f''(x_0) = \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2}$ |
| $f^{(3)}(x_0) = \frac{-5f_0 + 18f_1 - 24f_2 + 14f_3 - 3f_4}{2h^3}$ |
| $f^{(4)}(x_0) = \frac{3f_0 - 14f_1 + 26f_2 - 24f_3}{h^4}$ |

## 2.7 Numerical Differentiation Formulae

Table 2.2 summarizes the first and second central finite-difference derivative formulae. Higher-order derivatives are not the focus of the current work and only summarized for reference in Table 2.3. It is possible to obtain finite-difference formulae at abscissas that lie heavier on one side or completely on one side. Abscissas completely on "left" side are included in Table 2.4 and abscissas completely on "right" side are included in Table 2.5. These are included only for reference and further application will not be pursued in the current work.

## 2.8 Conclusion

This chapter summarized first- and second-order derivative approximations central finite-difference methods. More accurate approximations are obtained with truncation of the

Table 2.5: Backward-Difference Formulae of Order $O(h^2)$

| Backward-Difference |
|---|
| $f'(x_0) = \frac{3f_0-4f_{-1}+f_{-2}}{2h}$ |
| $f''(x_0) = \frac{2f_0-5f_{-1}+4f_{-2}-f_{-3}}{h^2}$ |
| $f^{(3)}(x_0) = \frac{5f_0-18f_{-1}+24f_{-2}-14f_{-3}+3f_{-4}}{2h^3}$ |
| $f^{(4)}(x_0) = \frac{3f_0-14f_{-1}+26f_{-2}-24f_{-3}+11f_{-4}-2f_{-5}}{h^4}$ |

Taylor series to higher-order terms or using a smaller step size. Both would produce better local approximations. Derivative approximations with high-order truncation errors can be obtained using a Richardson extrapolation. However, higher-order truncation is valid only when higher derivatives exist for the nonlinear function. Likewise, decreasing step size for better accuracy runs into roundoff errors at some points. The tradeoff between truncation error and roundoff error was shown in this chapter. An optimal step size can be chosen if the maximum magnitude of the truncated series can be quantified. Lastly, tables of forward- and backward- difference formulae were provided as reference.

# Chapter 3

# Complex-Step Derivative Approximation

This chapter presents new numerical derivative formulae using the complex-step derivative approach. First, the complex-step derivative approximation (CSDA) for the first derivative of a scalar function is summarized, followed by the derivation of the second-derivative approximation. Then, the Jacobian and Hessian approximations for multi-variable functions are derived. Next, the generalized CSDA is derived. Finally, some useful cases of the CSDA are shown and tested for both scalar and vector examples with comparison to the finite-difference approximations.

## 3.1 Complex-Step Approximation to the Derivative

In this section the complex-step approximation is shown. First, the derivative approximation of a scalar variable is summarized, followed by an extension to the second derivative. Then, approximations for multi-variable functions are presented for the Jacobian and Hessian matrices.

### 3.1.1 Scalar Case

Numerical finite-difference approximations for any order derivative can be obtained by Cauchy's integral formula [30]

$$f^{(n)}(z) = \frac{n!}{2\pi i} \int\limits_{\Gamma} \frac{f(\xi)}{(\xi - z)^{n+1}} d\xi \tag{3.1}$$

This function can be approximated by

$$f^{(n)}(z) \approx \frac{n!}{mh} \sum_{j=0}^{m-1} \frac{f\left(z + h\, e^{i\frac{2\pi j}{m}}\right)}{e^{i\frac{2\pi j n}{m}}} \tag{3.2}$$

where $h$ is the step-size and $i$ is the imaginary unit, $\sqrt{-1}$. For example, when $n = 1$, $m = 2$

$$f^{'}(z) \approx \frac{1}{2\,h}\Big[f(z+h) - f(z-h)\Big] \tag{3.3}$$

We can see that this formula involves a subtraction that would introduce near cancellation errors when the step size becomes too small.

#### First Derivative

The derivation of the complex-step derivative approximation is accomplished by approximating a nonlinear function with a complex variable using a Taylor's series expansion [27]:

$$f(x + ih) = f(x) + ihf^{'}(x) - \frac{h^2}{2!} f^{''}(x) - i\frac{h^3}{3!} f^{(3)}(x) + \frac{h^4}{4!} f^{(4)}(x) + \cdots \tag{3.4}$$

Taking only the imaginary parts of both sides gives

$$\Im\Big\{f(x+ih)\Big\} = hf^{'}(x) - \frac{h^3}{3!}f^{(3)}(x) + \cdots \tag{3.5}$$

Dividing by $h$ and rearranging yields

$$f^{'}(x) = \frac{\Im\Big\{f(x+ih)\Big\}}{h} + \cancelto{O(h^2) \approx 0}{\frac{h^2}{3!}f^{(3)}(x) + \cdots} \tag{3.6}$$

Terms with order $h^2$ or higher can be ignored since the interval $h$ can be chosen up to machine precision. Thus, to within first-order the complex-step derivative approximation is given by

$$f^{'}(x) = \frac{\Im\Big\{f(x+ih)\Big\}}{h} \quad , \quad E_{\text{trunc}}(h) = \frac{h^2}{6}f^{(3)}(x) \tag{3.7}$$

Note that this solution is not a function of differences, which ultimately provides better accuracy than a standard finite difference.

**Second Derivative**

In order to derive a second-derivative approximation, the real components of Eq. (3.4) are taken, which gives

$$\Re\left\{\frac{h^2}{2!}f^{''}(x)\right\} = f(x) - \Re\Big\{f(x+ih)\Big\} + \frac{h^4}{4!}f^{(4)}(x) + \cdots \tag{3.8}$$

Solving for $f^{''}(x)$ yields

$$f^{''}(x) = \frac{2!}{h^2}\Big[f(x) - \Re\Big\{f(x+ih)\Big\}\Big] + \frac{2!h^2}{4!}f^{(4)}(x) + \cdots \tag{3.9}$$

Analogous to the approach shown before, we truncate up to the second-order approximation to obtain

$$f^{''}(x) = \frac{2}{h^2}\Big[f(x) - \Re\Big\{f(x+ih)\Big\}\Big] \quad , \quad E_{\text{trunc}}(h) = \frac{h^2}{12} f^{(4)}(x) \tag{3.10}$$

As with Cauchy's formula, we can see that this formula involves a subtraction that may introduce machine cancellation errors when the step size is too small.

### 3.1.2 Vector Case

The scalar case is now expanded to include vector functions. This case involves a vector $\mathbf{f}(\mathbf{x})$ of order $m$ function equations and order $n$ variables with $\mathbf{x} = [x_1, \ x_2, \ \cdots, \ x_n]^T$.

#### First Derivative

The Jacobian of a vector function is a simple extension of the scalar case. This Jacobian is defined by

$$F_x \triangleq \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_p} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_p} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_q(\mathbf{x})}{\partial x_1} & \frac{\partial f_q(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_q(\mathbf{x})}{\partial x_p} & \cdots & \frac{\partial f_q(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \frac{\partial f_m(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_p} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix} \tag{3.11}$$

The complex approximation is obtained by

$$F_x = \frac{1}{h}\Im\begin{bmatrix} f_1(\mathbf{x}+ih\mathbf{e}_1) & f_1(\mathbf{x}+ih\mathbf{e}_2) & \cdots & f_1(\mathbf{x}+ih\mathbf{e}_p) & \dots & f_1(\mathbf{x}+ih\mathbf{e}_n) \\ f_2(\mathbf{x}+ih\mathbf{e}_1) & f_2(\mathbf{x}+ih\mathbf{e}_2) & \cdots & f_2(\mathbf{x}+ih\mathbf{e}_p) & \dots & f_2(\mathbf{x}+ih\mathbf{e}_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_q(\mathbf{x}+ih\mathbf{e}_1) & f_q(\mathbf{x}+ih\mathbf{e}_2) & \cdots & f_q(\mathbf{x}+ih\mathbf{e}_p) & \dots & f_q(\mathbf{x}+ih\mathbf{e}_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_m(\mathbf{x}+ih\mathbf{e}_1) & f_m(\mathbf{x}+ih\mathbf{e}_2) & \cdots & f_m(\mathbf{x}+ih\mathbf{e}_p) & \dots & f_m(\mathbf{x}+ih\mathbf{e}_n) \end{bmatrix} \tag{3.12}$$

where $\mathbf{e}_p$ is the $p^{\text{th}}$ column of an $n^{\text{th}}$-order identity matrix and $f_q$ is the $q^{\text{th}}$ equation of $\mathbf{f}(\mathbf{x})$.

**Second Derivative**

The procedure to obtain the Hessian matrix is more involved than the Jacobian case. The Hessian matrix for the $q^{\text{th}}$ equation of $\mathbf{f}(\mathbf{x})$ is defined by

$$F_{xx}^q \triangleq \begin{bmatrix} \frac{\partial^2 f_q(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_1 \partial x_p} & \cdots & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f_q(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_2 \partial x_p} & \cdots & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f_q(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_n \partial x_p} & \cdots & \frac{\partial^2 f_q(\mathbf{x})}{\partial x_n^2} \end{bmatrix} \tag{3.13}$$

The complex approximation is defined by

$$F_{xx}^q \equiv \begin{bmatrix} F_{xx}^q(1,1) & F_{xx}^q(1,2) & \cdots & F_{xx}^q(1,p) & \cdots & F_{xx}^q(1,n) \\ F_{xx}^q(2,1) & F_{xx}^q(2,2) & \cdots & F_{xx}^q(2,p) & \cdots & F_{xx}^q(2,n) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ F_{xx}^q(n,1) & F_{xx}^q(n,2) & \cdots & F_{xx}^q(n,p) & \cdots & F_{xx}^q(n,n) \end{bmatrix} \tag{3.14}$$

where $F^q_{xx}(i,j)$ is obtained by using Eq. (3.10). The easiest way to describe this procedure is by showing pseudocode, given by

```
$F_{xx} = \mathbf{0}_{n\times n\times m}$
**for** $\xi = 1$ **to** $m$
  $\texttt{out1} = \mathbf{f}(\mathbf{x})$
  **for** $\kappa = 1$ **to** $n$
    $\mathbf{small} = \mathbf{0}_{n\times 1}$
    $\mathbf{small}(\kappa) = 1$
    $\texttt{out2} = \mathbf{f}(\mathbf{x} + i * h * \mathbf{small})$
    $F_{xx}(\kappa,\kappa,\xi) = \frac{2}{h^2}\Big[\texttt{out1}(\xi) - \Re\big\{\texttt{out2}(\xi)\big\}\Big]$
  **end**
  $\lambda = 1$
  $\kappa = n - 1$
  **while** $\kappa > 0$
    **for** $\phi = 1$ **to** $\kappa$
      $\mathbf{img\_vec} = \mathbf{0}_{n\times 1}$
      $\mathbf{img\_vec}(\phi \ldots \phi + \lambda, 1) = 1$
      $\texttt{out2} = \mathbf{f}(\mathbf{x} + i * h * \mathbf{img\_vec})$
      $F_{xx}(\phi,\phi+\lambda,\xi) = \left[\frac{2}{h^2}\Big[\texttt{out1}(\xi) - \Re\big\{\texttt{out2}(\xi)\big\}\Big] - \sum_{\alpha=\phi}^{\phi+\lambda}\sum_{\beta=\phi}^{\phi+\lambda} F_{xx}(\alpha,\beta,\xi)\right]/2$
      $F_{xx}(\phi+\lambda,\phi,\xi) = F_{xx}(\phi,\phi+\lambda,\xi)$
    **end**
    $\kappa = \kappa - 1$
```

```
            $\lambda = \lambda + 1$
        end
    end
```

where $\Re\{\cdot\}$ denotes the real value operator. The first part of this code computes the diagonal elements and the second part computes the off-diagonal elements. The Hessian matrix is a symmetric matrix, so only the upper or lower triangular elements need to be computed.

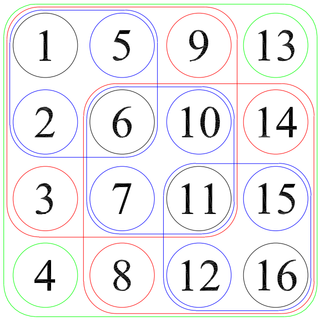


Figure 3.1: Graphical illustration of steps in finding the Hessian matrix.

The graphical illustration of the steps in obtaining the Hessian matrix for $n = 4$ is presented in Fig. 3.1. First, we find the second derivative of all diagonal terms, which are circled in black with index $\kappa = 1, \ldots, n = 4$. Second, we find the $2 \times 2$ off-diagonal terms circled in blue with index $\lambda = 1, \kappa = 3, \phi = 1, 2, 3$. Similarly, we then proceed to the next higher dimension circled in red the with index $\lambda = 2, \kappa = 2, \phi = 1, 2$. And finally the off-diagonal terms circled in green with index $\lambda = 3, \kappa = 1, \phi = 1$.

## 3.2 New Complex-Step Approximations

It can easily be seen from Eq. (3.4) that deriving second-derivative approximations without some sort of difference is difficult, if not intractable. With any complex number $I$ that has $|I| = 1$, it's impossible for $I^2 \perp 1$ and $I^2 \perp I$. But, it may be possible to obtain better approximations than Eq. (3.10). Let us again list down the Taylor series expansions with complex step sizes that would assist us in the forthcoming derivation and analysis:

$$i^{\frac{0}{6}} = 1 \qquad \begin{aligned} f(x+h) = {} & f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) + \frac{h^5}{5!}f^{(5)}(x) \\ & + \frac{h^6}{6!}f^{(6)}(x) + \frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) + \frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15a}$$

$$i^{\frac{1}{6}} \qquad \begin{aligned} f(x+i^{\frac{1}{6}}h) = {} & f(x) + i^{\frac{1}{6}}hf'(x) + i^{\frac{1}{3}}\frac{h^2}{2}f''(x) + i^{\frac{1}{2}}\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{2}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{5}{6}}\frac{h^5}{5!}f^{(5)}(x) \\ & + i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{7}{6}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{4}{3}}\frac{h^8}{8!}f^{(8)}(x) + i^{\frac{3}{2}}\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15b}$$

$$i^{\frac{2}{6}} = i^{\frac{1}{3}} \qquad \begin{aligned} f(x+i^{\frac{1}{3}}h) = {} & f(x) + i^{\frac{1}{3}}hf'(x) + i^{\frac{2}{3}}\frac{h^2}{2}f''(x) + i\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{4}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{5}{3}}\frac{h^5}{5!}f^{(5)}(x) \\ & - \frac{h^6}{6!}f^{(6)}(x) + i^{\frac{7}{3}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{8}{3}}\frac{h^8}{8!}f^{(8)}(x) - i\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15c}$$

$$i^{\frac{3}{6}} = i^{\frac{1}{2}} \qquad \begin{aligned} f(x+i^{\frac{1}{2}}h) = {} & f(x) + i^{\frac{1}{2}}hf'(x) + i\frac{h^2}{2}f''(x) + i^{\frac{3}{2}}\frac{h^3}{3!}f^{(3)}(x) - \frac{h^4}{4!}f^{(4)}(x) + i^{\frac{5}{2}}\frac{h^5}{5!}f^{(5)}(x) \\ & - i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{7}{2}}\frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) + i^{\frac{9}{2}}\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15d}$$

$$i^{\frac{4}{6}} = i^{\frac{2}{3}} \qquad \begin{aligned} f(x+i^{\frac{2}{3}}h) = {} & f(x) + i^{\frac{2}{3}}hf'(x) + i^{\frac{4}{3}}\frac{h^2}{2}f''(x) - \frac{h^3}{3!}f^{(3)}(x) + i^{\frac{8}{3}}\frac{h^4}{4}f^{(4)}(x) + i^{\frac{10}{3}}\frac{h^5}{5}f^{(5)}(x) \\ & + \frac{h^6}{6}f^{(6)}(x) + i^{\frac{14}{3}}\frac{h^7}{7}f^{(7)}(x) + i^{\frac{16}{3}}\frac{h^8}{8}f^{(8)}(x) - \frac{h^9}{9}f^{(9)}(x) \end{aligned} \tag{3.15e}$$

$$i^{\frac{5}{6}} \qquad f(x+i^{\frac{5}{6}}h) = f(x) + i^{\frac{5}{6}}hf'(x) + i^{\frac{5}{3}}\frac{h^2}{2}f''(x) + i^{\frac{5}{2}}\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{10}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{25}{6}}\frac{h^5}{5!}f^{(5)}(x)$$

$$+ i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{35}{6}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{20}{3}}\frac{h^8}{8!}f^{(8)}(x) + i^{\frac{15}{2}}\frac{h^9}{9!}f^{(9)}(x) \tag{3.15f}$$

$$i^{\frac{6}{6}} = i \qquad \begin{aligned} f(x+ih) = {} & f(x) + ihf'(x) - \frac{h^2}{2}f''(x) - i\frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) + i\frac{h^5}{5!}f^{(5)}(x) \\ & - \frac{h^6}{6!}f^{(6)}(x) - i\frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) + i\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15g}$$

$$i^{\frac{7}{6}} \qquad \begin{aligned} f(x+i^{\frac{7}{6}}h) = {} & f(x) + i^{\frac{7}{6}}hf'(x) + i^{\frac{7}{3}}\frac{h^2}{2}f''(x) + i^{\frac{7}{2}}\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{14}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{35}{6}}\frac{h^5}{5!}f^{(5)}(x) \\ & - i\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{49}{6}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{28}{3}}\frac{h^8}{8!}f^{(8)}(x) + i^{\frac{21}{2}}\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15h}$$

$$i^{\frac{8}{6}} = i^{\frac{4}{3}} \qquad \begin{aligned} f(x+i^{\frac{4}{3}}h) = {} & f(x) + i^{\frac{4}{3}}hf'(x) + i^{\frac{8}{3}}\frac{h^2}{2}f''(x) + \frac{h^3}{3!}f^{(3)}(x) + i^{\frac{16}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{20}{3}}\frac{h^5}{5!}f^{(5)}(x) \\ & + \frac{h^6}{6!}f^{(6)}(x) + i^{\frac{28}{3}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{32}{3}}\frac{h^8}{8!}f^{(8)}(x) + \frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15i}$$

$$i^{\frac{9}{6}} = i^{\frac{3}{2}} \qquad \begin{aligned} f(x+i^{\frac{3}{2}}h) = {} & f(x) + i^{\frac{3}{2}}hf'(x) - i\frac{h^2}{2}f''(x) + i^{\frac{1}{2}}\frac{h^3}{3!}f^{(3)}(x) - \frac{h^4}{4!}f^{(4)}(x) + i^{\frac{15}{2}}\frac{h^5}{5!}f^{(5)}(x) \\ & + i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{21}{2}}\frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) + i^{\frac{27}{2}}\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15j}$$

$$i^{\frac{10}{6}} = i^{\frac{5}{3}} \qquad \begin{aligned} f(x+i^{\frac{5}{3}}h) = {} & f(x) + i^{\frac{5}{3}}hf'(x) + i^{\frac{10}{3}}\frac{h^2}{2}f''(x) + i\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{20}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{25}{3}}\frac{h^5}{5!}f^{(5)}(x) \\ & - \frac{h^6}{6!}f^{(6)}(x) + i^{\frac{35}{3}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{40}{3}}\frac{h^8}{8!}f^{(8)}(x) - i\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15k}$$

$$i^{\frac{11}{6}} \qquad \begin{aligned} f(x+i^{\frac{11}{6}}h) = {} & f(x) + i^{\frac{11}{6}}hf'(x) + i^{\frac{11}{3}}\frac{h^2}{2}f''(x) + i^{\frac{11}{2}}\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{22}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{55}{6}}\frac{h^5}{5!}f^{(5)}(x) \\ & - i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{77}{6}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{44}{3}}\frac{h^8}{8!}f^{(8)}(x) + i^{\frac{33}{2}}\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15l}$$

$$i^{\frac{12}{6}} = -1 \qquad \begin{aligned} f(x-h) = {} & f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) - \frac{h^5}{5!}f^{(5)}(x) \\ & + \frac{h^6}{6!}f^{(6)}(x) - \frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) - \frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.15m}$$

$$i^{\frac{13}{6}} \qquad f(x+i^{\frac{13}{6}}h) = f(x) + i^{\frac{13}{6}}hf'(x) + i^{\frac{13}{3}}\frac{h^2}{2}f''(x) + i^{\frac{13}{2}}\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{26}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{65}{6}}\frac{h^5}{5!}f^{(5)}(x)$$

$$+ i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{91}{6}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{52}{3}}\frac{h^8}{8!}f^{(8)}(x) + i^{\frac{39}{2}}\frac{h^9}{9!}f^{(9)}(x) \tag{3.15n}$$

$$i^{\frac{14}{6}} = i^{\frac{7}{3}} \qquad f(x + i^{\frac{7}{3}}h) = f(x) + i^{\frac{7}{3}}hf'(x) + i^{\frac{14}{3}}\frac{h^2}{2}f''(x) - i\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{28}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{35}{3}}\frac{h^5}{5!}f^{(5)}(x)$$

$$- \frac{h^6}{6!}f^{(6)}(x) + i^{\frac{49}{3}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{56}{3}}\frac{h^8}{8!}f^{(8)}(x) + i\frac{h^9}{9!}f^{(9)}(x) \tag{3.15o}$$

$$i^{\frac{15}{6}} = i^{\frac{5}{2}} \qquad f(x + i^{\frac{5}{2}}h) = f(x) + i^{\frac{5}{2}}hf'(x) + i\frac{h^2}{2}f''(x) + i^{\frac{15}{2}}\frac{h^3}{3!}f^{(3)}(x) - \frac{h^4}{4!}f^{(4)}(x) + i^{\frac{25}{2}}\frac{h^5}{5!}f^{(5)}(x)$$

$$- i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{35}{2}}\frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) + i^{\frac{45}{2}}\frac{h^9}{9!}f^{(9)}(x) \tag{3.15p}$$

$$i^{\frac{16}{6}} = i^{\frac{8}{3}} \qquad f(x + i^{\frac{8}{3}}h) = f(x) + i^{\frac{8}{3}}hf'(x) + i^{\frac{16}{3}}\frac{h^2}{2}f''(x) + \frac{h^3}{3!}f^{(3)}(x) + i^{\frac{32}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{40}{3}}\frac{h^5}{5!}f^{(5)}(x)$$

$$+ \frac{h^6}{6!}f^{(6)}(x) + i^{\frac{56}{3}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{64}{3}}\frac{h^8}{8!}f^{(8)}(x) + \frac{h^9}{9!}f^{(9)}(x) \tag{3.15q}$$

$$i^{\frac{17}{6}} \qquad f(x + i^{\frac{17}{6}}h) = f(x) + i^{\frac{17}{6}}hf'(x) + i^{\frac{17}{3}}\frac{h^2}{2}f''(x) + i^{\frac{34}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{85}{6}}\frac{h^5}{5!}f^{(5)}(x)$$

$$+ i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{119}{6}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{68}{3}}\frac{h^8}{8!}f^{(8)}(x) + i^{\frac{51}{2}}\frac{h^9}{9!}f^{(9)}(x) \tag{3.15r}$$

$$i^{\frac{18}{6}} = -i \qquad f(x - ih) = f(x) - ihf'(x) - \frac{h^2}{2}f''(x) + i\frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) - i\frac{h^5}{5!}f^{(5)}(x)$$

$$- \frac{h^6}{6!}f^{(6)}(x) + i\frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) - i\frac{h^9}{9!}f^{(9)}(x) \tag{3.15s}$$

$$i^{\frac{19}{6}} \qquad f(x + i^{\frac{19}{6}}h) = f(x) + i^{\frac{19}{6}}hf'(x) + i^{\frac{19}{3}}\frac{h^2}{2}f''(x) + i^{\frac{19}{2}}\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{38}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{95}{6}}\frac{h^5}{5!}f^{(5)}(x)$$

$$- i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{133}{6}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{76}{3}}\frac{h^8}{8!}f^{(8)}(x) + i^{\frac{57}{2}}\frac{h^9}{9!}f^{(9)}(x) \tag{3.15t}$$

$$i^{\frac{20}{6}} = i^{\frac{10}{3}} \qquad f(x + i^{\frac{10}{3}}h) = f(x) + i^{\frac{10}{3}}hf'(x) + i^{\frac{20}{3}}\frac{h^2}{2}f''(x) - \frac{h^3}{3!}f^{(3)}(x) + i^{\frac{40}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{50}{3}}\frac{h^5}{5!}f^{(5)}(x)$$

$$+ \frac{h^6}{6!}f^{(6)}(x) + i^{\frac{70}{3}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{80}{3}}\frac{h^8}{8!}f^{(8)}(x) - \frac{h^9}{9!}f^{(9)}(x) \tag{3.15u}$$

$$i^{\frac{21}{6}} = i^{\frac{7}{2}} \qquad f(x + i^{\frac{7}{2}}h) = f(x) + i^{\frac{7}{2}}hf'(x) - i\frac{h^2}{2}f''(x) + i^{\frac{21}{2}}\frac{h^3}{3!}f^{(3)}(x) - \frac{h^4}{4!}f^{(4)}(x) + i^{\frac{35}{2}}\frac{h^5}{5!}f^{(5)}(x)$$

$$+ i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{49}{2}}\frac{h^7}{7!}f^{(7)}(x) + \frac{h^8}{8!}f^{(8)}(x) + i^{\frac{63}{2}}\frac{h^9}{9!}f^{(9)}(x) \tag{3.15v}$$

$$i^{\frac{22}{6}} = i^{\frac{11}{3}} \qquad f(x + i^{\frac{11}{3}}h) = f(x) + i^{\frac{11}{3}}hf'(x) + i^{\frac{22}{3}}\frac{h^2}{2}f''(x) - i\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{44}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{55}{3}}\frac{h^5}{5!}f^{(5)}(x)$$

$$- \frac{h^6}{6!}f^{(6)}(x) + i^{\frac{77}{3}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{88}{3}}\frac{h^8}{8!}f^{(8)}(x) + i\frac{h^9}{9!}f^{(9)}(x) \tag{3.15w}$$

$$i^{\frac{23}{6}} \qquad f(x + i^{\frac{23}{6}}h) = f(x) + i^{\frac{23}{6}}hf'(x) + i^{\frac{23}{3}}\frac{h^2}{2}f''(x) + i^{\frac{23}{2}}\frac{h^3}{3!}f^{(3)}(x) + i^{\frac{46}{3}}\frac{h^4}{4!}f^{(4)}(x) + i^{\frac{115}{6}}\frac{h^5}{5!}f^{(5)}(x)$$

$$- i\frac{h^6}{6!}f^{(6)}(x) + i^{\frac{161}{6}}\frac{h^7}{7!}f^{(7)}(x) + i^{\frac{92}{3}}\frac{h^8}{8!}f^{(8)}(x) + i^{\frac{69}{2}}\frac{h^9}{9!}f^{(9)}(x) \tag{3.15x}$$

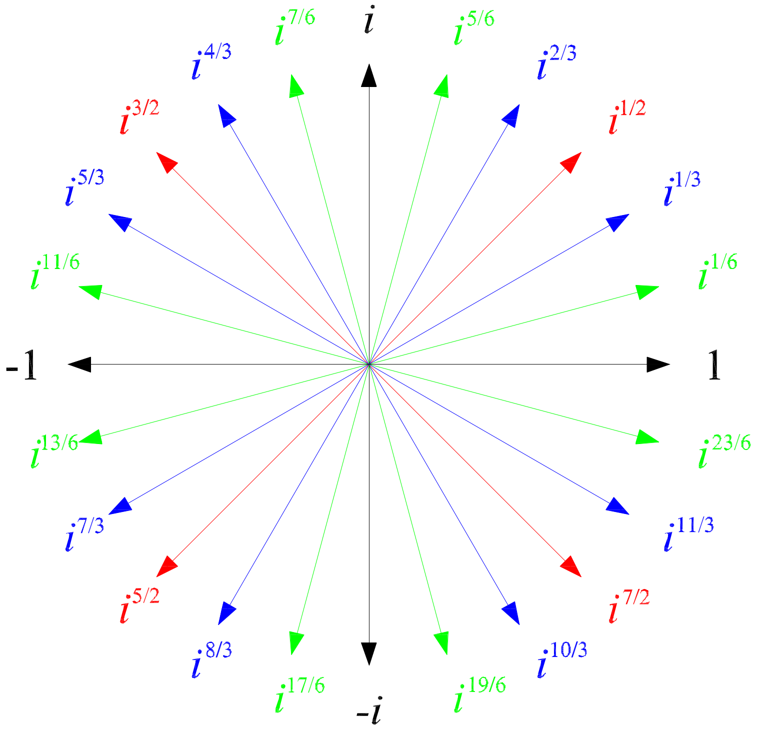
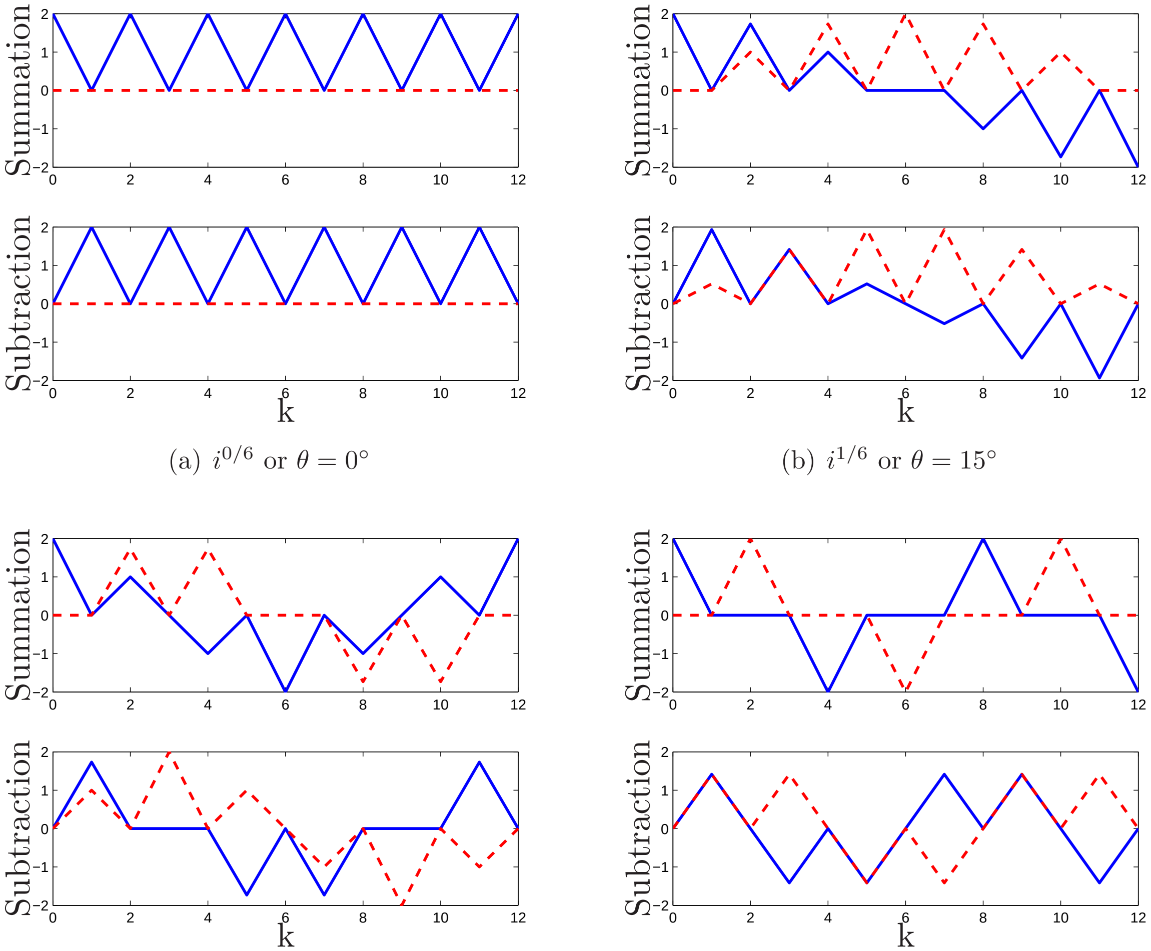


Figure 3.2: Various complex numbers.

Figure 3.2 shows the unity magnitude complex number raised to various rational number powers with common denominator of 6, i.e. multiple of 15°. It may be more convenient to represent the complex number in another way. With identities from Appendix B.1.1 can be derived that $i^{p/q} = e^{i\theta}$ with phase angle $\theta = \frac{p}{q}90^\circ = \frac{p}{2q}\pi$ rad. The Taylor series expansion



(a) $i^{0/6}$ or $\theta = 0°$

(b) $i^{1/6}$ or $\theta = 15°$

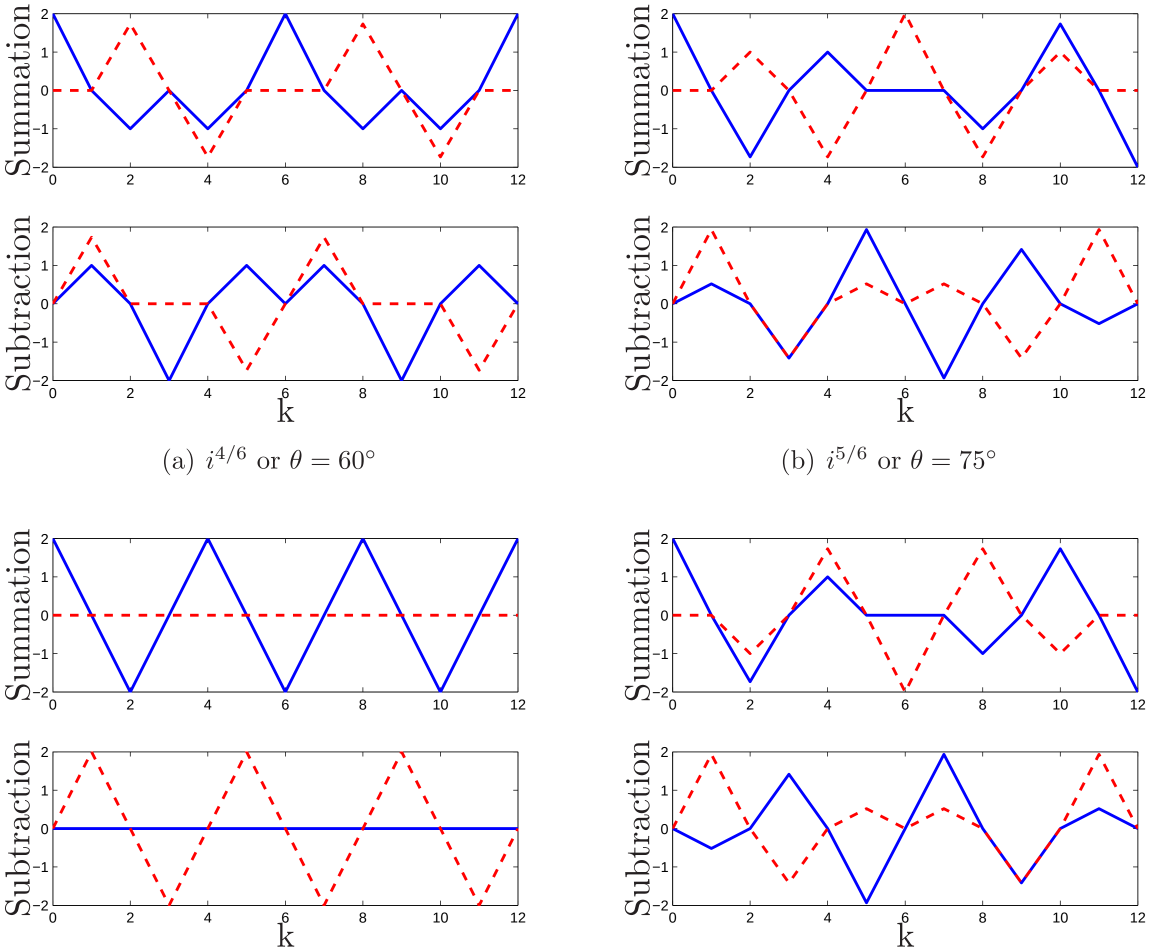
(c) $i^{2/6}$ or $\theta = 30°$

(d) $i^{3/6}$ or $\theta = 45°$

Figure 3.3: for $\theta$ from 0° to 45° (Solid Lines = Real, Dotted Lines = Imaginary)

pair with complex step sizes can then be rewritten as

$$\begin{aligned} f(x+e^{i\theta}h) = f(x) &+ e^{i\theta}hf'(x) + e^{2i\theta}\frac{h^2}{2}f''(x) + e^{3i\theta}\frac{h^3}{3!}f^{(3)}(x) \\ &+ e^{4i\theta}\frac{h^4}{4!}f^{(4)}(x) + e^{5i\theta}\frac{h^5}{5!}f^{(5)}(x) + e^{6i\theta}\frac{h^6}{6!}f^{(6)}(x) \\ &+ e^{7i\theta}\frac{h^7}{7!}f^{(7)}(x) + e^{8i\theta}\frac{h^8}{8!}f^{(8)}(x) + e^{9i\theta}\frac{h^9}{9!}f^{(9)}(x) \\ &+ e^{10i\theta}\frac{h^{10}}{10!}f^{(10)}(x) \end{aligned} \tag{3.16a}$$



(a) $i^{4/6}$ or $\theta = 60°$

(b) $i^{5/6}$ or $\theta = 75°$

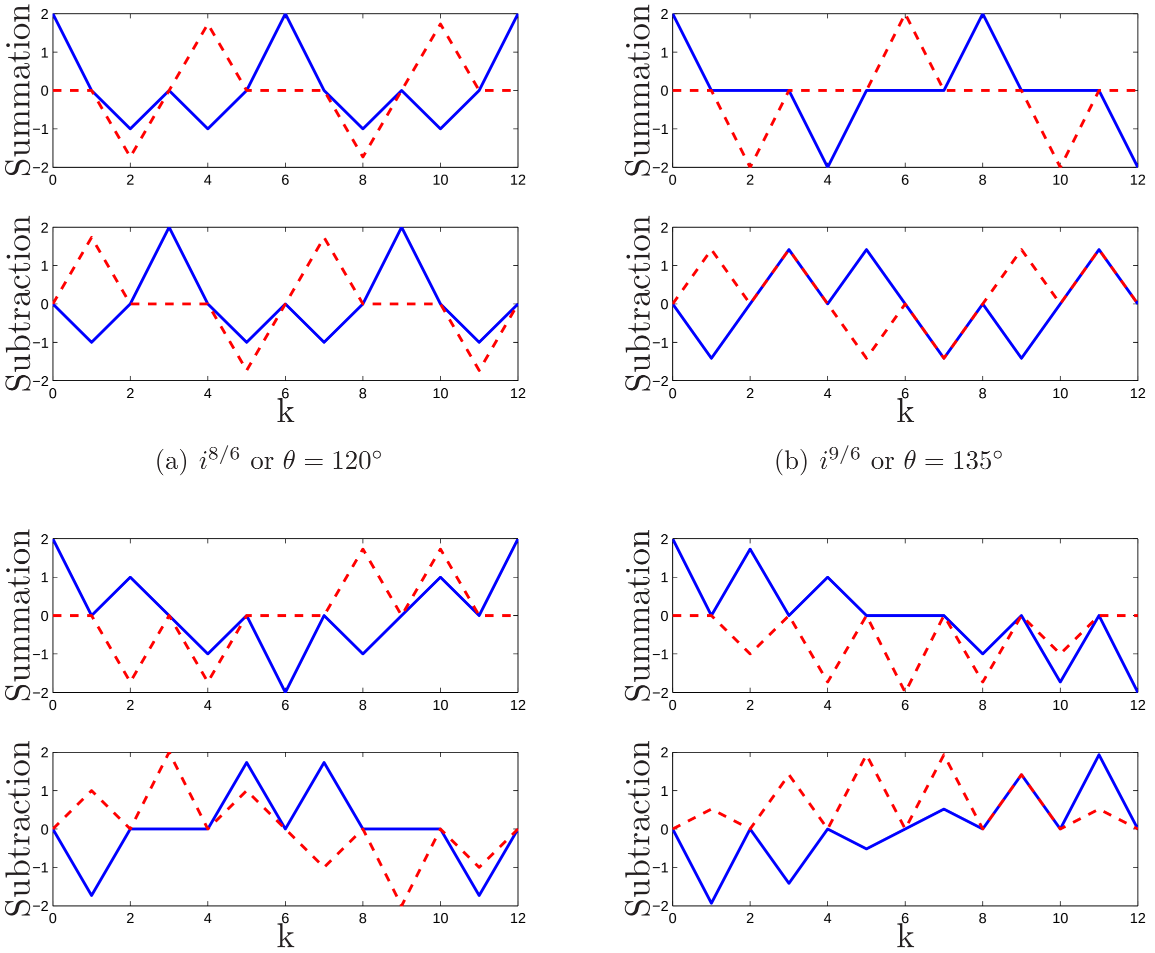
(c) $i^{6/6}$ or $\theta = 90°$

(d) $i^{7/6}$ or $\theta = 105°$

Figure 3.4: for $\theta$ from 60° to 105° (Solid Lines = Real, Dotted Lines = Imaginary)

$$\begin{aligned} f(x + e^{i(\theta+\pi)}h) = f(x) &+ e^{i(\theta+\pi)}hf'(x) + e^{2i(\theta+\pi)}\frac{h^2}{2}f''(x) + e^{3i(\theta+\pi)}\frac{h^3}{3!}f^{(3)}(x) \\ &+ e^{4i(\theta+\pi)}\frac{h^4}{4!}f^{(4)}(x) + e^{5i(\theta+\pi)}\frac{h^5}{5!}f^{(5)}(x) + e^{6i(\theta+\pi)}\frac{h^6}{6!}f^{(6)}(x) \\ &+ e^{7i(\theta+\pi)}\frac{h^7}{7!}f^{(7)}(x) + e^{8i(\theta+\pi)}\frac{h^8}{8!}f^{(8)}(x) + e^{9i(\theta+\pi)}\frac{h^9}{9!}f^{(9)}(x) \\ &+ e^{10i(\theta+\pi)}\frac{h^{10}}{10!}f^{(10)}(x) \end{aligned} \tag{3.16b}$$

Instead of representing the complex step with the imaginary number $i$ raised to power, it can be represented using trigonometry with Euler's relation, $e^{i\theta} = \cos\theta + i\sin\theta$. Euler's



(a) $i^{8/6}$ or $\theta = 120°$

(b) $i^{9/6}$ or $\theta = 135°$

(c) $i^{10/6}$ or $\theta = 150°$

(d) $i^{11/6}$ or $\theta = 165°$

Figure 3.5: for $\theta$ from 120° to 165° (Solid Lines = Real, Dotted Lines = Imaginary)

relation bridges the field of algebra with geometry. Therefore, the addition and subtraction of Eqs. (3.16) can be written as

$$\begin{aligned} f(x+e^{i\theta}h)+f(x+e^{i(\theta+\pi)}h) = 2f(x) &+ 2[\cos 2\theta + i\sin 2\theta]\frac{h^2}{2}f''(x) \\ &+ 2[\cos 4\theta + i\sin 4\theta]\frac{h^4}{4!}f^{(4)}(x) \\ &+ 2[\cos 6\theta + i\sin 6\theta]\frac{h^6}{6!}f^{(6)}(x) \\ &+ 2[\cos 8\theta + i\sin 8\theta]\frac{h^8}{8!}f^{(8)}(x) \\ &+ 2[\cos 10\theta + i\sin 10\theta]\frac{h^{10}}{10!}f^{(10)}(x) \end{aligned} \tag{3.17a}$$

$$\begin{aligned} f(x+e^{i\theta}h)-f(x+e^{i(\theta+\pi)}h) = 2[\cos\theta + i\sin\theta]hf'(x) &+ 2[\cos 3\theta + i\sin 3\theta]\frac{h^3}{3!}f^{(3)}(x) \\ &+ 2[\cos 5\theta + i\sin 5\theta]\frac{h^5}{5!}f^{(5)}(x) \\ &+ 2[\cos 7\theta + i\sin 7\theta]\frac{h^7}{7!}f^{(7)}(x) \\ &+ 2[\cos 9\theta + i\sin 9\theta]\frac{h^9}{9!}f^{(9)}(x) \end{aligned} \tag{3.17b}$$

Solving these two equations for $f''(x)$ and $f'(x)$ yields

$$\begin{aligned} f''(x) = &\frac{f(x+e^{i\theta}h)-2f(x)+f(x+e^{i(\theta+pi)}h)}{[\cos 2\theta + i\sin 2\theta]h^2} - \frac{2[\cos 4\theta + i\sin 4\theta]}{\cos 2\theta + i\sin 2\theta}\frac{h^2}{4!}f^{(4)}(x) \\ &- \frac{2[\cos 6\theta + i\sin 6\theta]}{\cos 2\theta + i\sin 2\theta}\frac{h^4}{6!}f^{(6)}(x) - \frac{2[\cos 8\theta + i\sin 8\theta]}{\cos 2\theta + i\sin 2\theta}\frac{h^6}{8!}f^{(8)}(x) \\ &- \frac{2[\cos 10\theta + i\sin 10\theta]}{\cos 2\theta + i\sin 2\theta}\frac{h^8}{10!}f^{(10)}(x) \end{aligned} \tag{3.18a}$$

$$\begin{aligned}f'(x) = {} & \frac{f(x+e^{i\theta}h) - f(x+e^{i(\theta+\pi)}h)}{2[\cos\theta + i\sin\theta]h} - \frac{\cos 3\theta + i\sin 3\theta}{\cos\theta + i\sin\theta}\frac{h^2}{3!}f^{(3)}(x) \\ & - \frac{\cos 5\theta + i\sin 5\theta}{\cos\theta + i\sin\theta}\frac{h^4}{5!}f^{(5)}(x) - \frac{\cos 7\theta + i\sin 7\theta}{\cos\theta + i\sin\theta}\frac{h^6}{7!}f^{(7)}(x) \\ & - \frac{\cos 9\theta + i\sin 9\theta}{\cos\theta + i\sin\theta}\frac{h^8}{9!}f^{(9)}(x)\end{aligned} \tag{3.18b}$$

### 3.2.1 General Form

Now the pattern is obvious from previous derivation and we can generalize the complex Taylor series expansion pair from the previous section as

$$f(x+e^{i\theta}h) = f(x) + \sum_{n=1}^{\infty} e^{ni\theta}\frac{h^n}{n!}f^{(n)}(x) \tag{3.19a}$$

$$f(x+e^{i(\theta+\pi)}h) = f(x) + \sum_{n=1}^{\infty} e^{ni(\theta+\pi)}\frac{h^n}{n!}f^{(n)}(x) \tag{3.19b}$$

and their summation and subtraction pair

$$f(x+e^{i\theta}h) + f(x+e^{i(\theta+\pi)}h) = 2f(x) + 2\sum_{n=1}^{\infty}\left[\cos 2n\theta + i\sin 2n\theta\right]\frac{h^{2n}}{(2n)!}f^{(2n)}(x) \tag{3.20a}$$

$$\begin{aligned}f(x+e^{i\theta}h) - f(x+e^{i(\theta+\pi)}h) = 2\sum_{n=1}^{\infty}\Bigg[ & \cos[(2n-1)\theta] \\ & + i\sin[(2n-1)\theta]\Bigg]\frac{h^{2n-1}}{(2n-1)!}f^{(2n-1)}(x)\end{aligned} \tag{3.20b}$$

Finally solving for $f''(x)$ and $f'(x)$ yields

$$\begin{aligned}f''(x) = {} & \frac{f(x+e^{i\theta}h) - 2f(x) + f(x+e^{i(\theta+\pi)}h)}{[\cos 2\theta + i\sin 2\theta]h^2} \\ & - \frac{2}{\cos 2\theta + i\sin 2\theta}\sum_{n=2}^{\infty}\left[\cos 2n\theta + i\sin 2n\theta\right]\frac{h^{2n-2}}{(2n)!}f^{(2n)}(x)\end{aligned} \tag{3.21a}$$

$$f'(x) = \frac{f(x+e^{i\theta}h) - f(x+e^{i(\theta+\pi)}h)}{2[\cos\theta + i\sin\theta]h}$$

$$- \frac{1}{\cos\theta + i\sin\theta} \sum_{n=2}^{\infty} \left[ \cos[(2n-1)\theta] + i\sin[(2n-1)\theta] \right] \frac{h^{2n-2}}{(2n-1)!} f^{(2n-1)}(x) \quad (3.21b)$$

or simply

$$\begin{aligned} f''(x) &= \frac{f(x+e^{i\theta}h) - 2f(x) + f(x+e^{i(\theta+\pi)}h)}{[\cos 2\theta + i\sin 2\theta]h^2} \\ &\quad - 2\sum_{n=2}^{\infty} \left[ \cos[(2n-2)\theta] + i\sin[(2n-2)\theta] \right] \frac{h^{2n-2}}{(2n)!} f^{(2n)}(x) \quad (3.22a) \\ f'(x) &= \frac{f(x+e^{i\theta}h) - f(x+e^{i(\theta+\pi)}h)}{2[\cos\theta + i\sin\theta]h} \\ &\quad - \sum_{n=2}^{\infty} \left[ \cos[(2n-2)\theta] + i\sin[(2n-2)\theta] \right] \frac{h^{2n-2}}{(2n-1)!} f^{(2n-1)}(x) \quad (3.22b) \end{aligned}$$

Instead of raising $i$ to some power, Eqs. (3.22) clearly separate the real and imaginary components. If the separation is not necessary, they can be expressed in a simpler form

$$f''(x) = \frac{f(x+e^{i\theta}h) - 2f(x) + f(x-e^{i\theta}h)}{(e^{i\theta}h)^2} - 2\sum_{n=2}^{\infty} \frac{(e^{i\theta}h)^{2n-2}}{(2n)!} f^{(2n)}(x) \quad (3.23a)$$

$$f'(x) = \frac{f(x+e^{i\theta}h) - f(x-e^{i\theta}h)}{2e^{i\theta}h} - \sum_{n=2}^{\infty} \frac{(e^{i\theta}h)^{2n-2}}{(2n-1)!} f^{(2n-1)}(x) \quad (3.23b)$$

This generalization also works for pure real-valued finite differences as in Chapter 2 by simply using $\theta = 0$. The Richardson extrapolation from §2.4 applies here too. The extension of all the aforementioned approximations to multi-variables for the Jacobian and Hessian matrices is straightforward, which follow along similar lines as the previous section.

### 3.2.2 Useful Cases

The first and second order CSDA have so far been generalized for any angle of the complex-step. However, a suitable angle $\theta$ is needed to unlock the full potential of CSDA. Figures

3.3, 3.4 and 3.5 show the summation and subtraction pairs of Taylor series expansion with complex step sizes that are 180° apart. These figures give some intuitive perception for the following derivation of generalized CSDA. The implications of these figures will be revisited in §3.2.2. Observing Figs. 3.3 to 3.5, there are several *interesting* cases where certain elements (real or imaginary component) of the series annihilate. This is exactly the desirable phenomenon to be taken advantage of to increase the convergence rate of the Taylor series approximation towards the original nonlinear function. In fact, this is the main goal of evaluating functions with a complex step size.

The summation plots (refer to Eq. (3.20a)) are associated with the second-order derivative and the subtraction plots (refer to Eq. (3.20b)) are associated with the first-order derivative. Most cases have few or no "flat lines" where annihilation occurs. A flat line or annihilation occurs when the transcendental function sine or cosine evaluates to zero. This obviously has to occur at 90° or 270° for cosine and 0° or 180° for sine. From Euler's relation, cosine is coupled to the real component and sine to the imaginary component. Therefore, the CSDA angle needs to be related to these four angles to produce the greatest numbers of *flat lines*. Thus, it is not surprising to see 45° produces the greatest number of *flat lines* for the summation cases and 60° produces most *flat lines* for the subtraction cases. In addition, it is desired to have more *flat lines* at the lower $k$ number, as $k$ links to the order of derivative and canceling of these terms enhances the derivative approximation accuracy with higher-order truncation error.

However, there is a tradeoff between first-derivative accuracy and second-derivative accuracy. For example 45° may work best for second-order accuracy but offers no CSDA benefit for the first derivative. On the other hand, 60° offers better first-order derivative accuracy at the expense of accuracy in the second-order derivative. These two unit angles will be further discussed below.

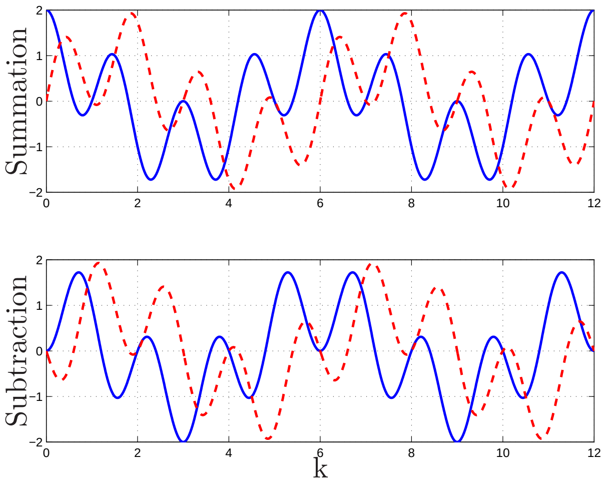


Figure 3.6: $i^{4/6}$ or $\theta = 60°$ (Solid Lines = Real, Dotted Lines = Imaginary)

Figures 3.3 to 3.5 are generated at integer $k$ values. Using the $\theta = 60°$ case as example, Fig. 3.6 shows the information in between two successive integer $k$ values. This figure is not important in our current derivation of the first- and second-order derivative approximations but may pave ways for generalization of "partial" or fractional derivatives [37, 38] in the future. Also, the periodic or sinusoidal patterns hint to a possible integration of the angle $\theta$ which associated with the *annihilation* points into the generalizations.

$\underline{\theta = 45°}$

From Eqs. (3.23a) with $\theta = 45°$, taking only the imaginary components gives

$$f^{''}(x) = \frac{\Im\big\{f(x+i^{1/2}h) + f(x+i^{5/2}h)\big\}}{h^2} \quad , \quad E_{\text{trunc}}(h) = \frac{h^4}{360} f^{(6)}(x) \tag{3.24}$$

Note that when $n = 2$, the imaginary component $\sin 2n\theta = 0$, thus the first non-zero value occurs when $n = 3$, corresponding to $O(h^4)$, which is the main goal of CSDA. This approximation is still subject to difference errors, but the truncation error associated with this approximation is $h^4\, f^{(6)}(x)/360$ whereas the error associated with Eq. (3.10) is $h^2\, f^{(4)}(x)/12$.

It will also be shown through simulation that Eq. (3.24) is less sensitive to roundoff errors than Eq. (3.10).

Unfortunately, to obtain the first and second derivatives using Eqs. (3.7) and (3.24) requires function evaluations of $f(x+ih)$, $f(x+i^{1/2}h)$ and $f(x+i^{5/2}h)$. To obtain a first-derivative expression that involves only $f(x+i^{1/2}h)$ and $f(x+i^{5/2}h)$, we substitute $\theta = 45^\circ$ into Eq. (3.23b) and again take the imaginary components to give

$$f^{'}(x) = \frac{f(x+i^{1/2}h) - f(x+i^{5/2}h)}{\sqrt{2}\,(i+1)\,h} \tag{3.25}$$

Actually either the imaginary or real parts of Eq. (3.25) can be taken to determine $f^{'}(x)$; however, it's better to use the imaginary parts since no differences exist (they are actually *additions* of imaginary numbers) since $f(x+i^{1/2}h)-f(x+i^{5/2}h) = f(x+i^{1/2}h)-f(x-i^{1/2}h)$. This yields

$$f^{'}(x) = \frac{\Im\big\{f(x+i^{1/2}h) - f(x+i^{5/2}h)\big\}}{h\sqrt{2}} \quad , \quad E_{\text{trunc}}(h) = -\frac{h^2}{6} f^{(3)}(x) \tag{3.26}$$

The approximation in Eq. (3.26) has errors equal to Eq. (3.7). Hence, both forms yield identical answers; however, Eq. (3.26) uses the same function evaluations as Eq. (3.24).

Further refinements can be made by applying the Richardson extrapolation approach from §2.4 [33]. From Eq. (2.37) with $q = 2$ and $k_1 = 4$

$$\begin{aligned} f^{''}(x) &= \frac{2^4 \frac{\Im\left\{f\left(x+i^{1/2}\frac{h}{2}\right)+f\left(x+i^{5/2}\frac{h}{2}\right)\right\}}{h^2/4} - \frac{\Im\left\{f(x+i^{1/2}h)+f(x+i^{5/2}h)\right\}}{h^2}}{2^4-1} \\ &= \frac{\Im\big\{64\big[f(x+i^{1/2}\frac{h}{2}) + f(x+i^{5/2}\frac{h}{2})\big] - \big[f(x+i^{1/2}h) + f(x+i^{5/2}h)\big]\big\}}{15h^2} \quad , \\ E_{\text{trunc}}(h) &= -\frac{h^8}{1,814,400} f^{(10)}(x) \end{aligned} \tag{3.27}$$

This approach can be continued *ad nauseam* using the next value of $k$. However, the next highest-order derivative-difference past $O(h^8)$ that has imaginary parts is $O(h^{12})$. This error is given by $\frac{h^{12}}{4.35891456\times 10^{10}} f^{(14)}(x)$. Hence, it seems unlikely that the accuracy will improve much by using more terms. The same approach can be applied to the first derivative as well. Applying the Richardson extrapolation with $q = 2$, $k_1 = 2$ to Eq. (3.26) yields

$$f'(x) = \Im\left\{8\left[f\left(x + i^{1/2}\frac{h}{2}\right) - f\left(x + i^{5/2}\frac{h}{2}\right)\right] - \left[f\left(x + i^{1/2}h\right) - f\left(x + i^{5/2}h\right)\right]\right\} / (3\sqrt{2}\,h)$$
$$E_{\text{trunc}}(h) = \frac{h^4}{120} f^{(5)}(x) \tag{3.28}$$

Performing a Richardson extrapolation again would cancel fifth-order derivative errors, which leads to the following approximation:

$$f'(x) = \Im\left\{4096\left[f\left(x + i^{1/2}\frac{h}{4}\right) - f\left(x + i^{5/2}\frac{h}{4}\right)\right] - 640\left[f\left(x + i^{1/2}\frac{h}{2}\right) - f\left(x + i^{5/2}\frac{h}{2}\right)\right]\right.$$
$$\left. + 16\left[f(x + i^{1/2}h) - f(x + i^{5/2}h)\right]\right\} / (720\sqrt{2}\,h) \quad , \quad E_{\text{trunc}}(h) = \frac{h^6}{5040} f^{(7)}(x) \tag{3.29}$$

As with Eq. (3.26), the approximations in Eq. (3.28) and Eq. (3.29) are not subject to roundoff errors, so an arbitrarily small value of $h$ can be chosen.

$\underline{\theta = 60^\circ}$

Using $\theta = 60^\circ$ in Eqs. (3.23) gives

$$f'(x) = \frac{\Im\{f(x + i^{2/3}h) - f(x + i^{8/3}h)\}}{\sqrt{3}h} \quad , \quad E_{\text{trunc}}(h) = \frac{h^4}{120} f^{(5)}(x) \tag{3.30a}$$

$$f''(x) = \frac{\Im\{f(x + i^{2/3}h) + f(x + i^{8/3}h)\}}{\sqrt{3}h} \quad , \quad E_{\text{trunc}}(h) = \frac{h^2}{24} f^{(4)}(x) \tag{3.30b}$$

Performing a Richardson extrapolation once on each of these equations yields

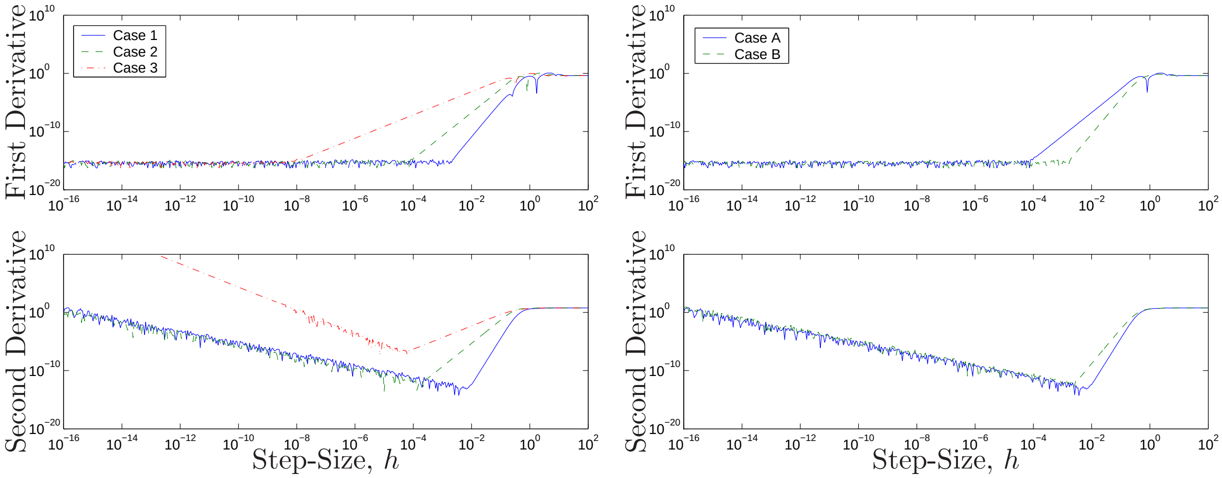
$$f^{'}(x) = \frac{\Im\Big\{32\big[f\left(x+i^{2/3}\frac{h}{2}\right) - f\left(x+i^{8/3}\frac{h}{2}\right)\big] - \big[f(x+i^{2/3}h) - f(x+i^{8/3}h)\big]\Big\}}{15\sqrt{3}\,h} \quad ,$$
$$E_{\text{trunc}}(h) = -\frac{h^6}{5040}f^{(7)}(x) \quad (3.31a)$$

$$f^{''}(x) = 2\frac{\Im\Big\{\big[f(x+i^{2/3}h) + f(x+i^{8/3}h)\big] - 16\,\big[f(x+i^{2/3}\frac{h}{2}) + f(x+i^{8/3}\frac{h}{2})\big]\Big\}}{3\sqrt{3}\,h^2} \quad ,$$
$$E_{\text{trunc}}(h) = -\frac{h^6}{40320}f^{(8)}(x) \quad (3.31b)$$

These solutions have the same order of accuracy as Eq. (3.29), but involves less function evaluations. Using $i^{2/3}$ instead of $i^{1/2}$ for the second-derivative approximation yields worse results than Eq. (3.27) since the approximation has errors on the order of $h^6\,f^{(8)}(x)$ instead of $h^8\,f^{(10)}(x)$. Hence, a tradeoff between the first-derivative and second-derivative accuracy will always exist if using the same function evaluations for both is desired. Higher-order versions of Eqs. (3.31) are given by

$$\begin{aligned} f^{'}(x) = \Im\Big\{&3072\,\big[f(x+i^{2/3}\frac{h}{4}) - f(x+i^{8/3}\frac{h}{4})\big] - 256\,\big[f(x+i^{2/3}\frac{h}{2}) - f(x+i^{8/3}\frac{h}{2})\big] \\ &+ 5\,\big[f(x+i^{2/3}h) - f(x+i^{8/3}h)\big]\Big\}/(645\sqrt{3}\,h) \quad , \end{aligned}$$
$$E_{\text{trunc}}(h) = \frac{h^{10}}{39916800}f^{(11)}(x) \quad (3.32a)$$

$$\begin{aligned} f^{''}(x) = 2\,\Im\Big\{&15\,\big[f(x+i^{2/3}h) + f(x+i^{8/3}h)\big] + 16\,\big[f(x+i^{2/3}\frac{h}{2}) + f(x+i^{8/3}\frac{h}{2})\big] \\ &- 4096\,\big[f(x+i^{2/3}\frac{h}{4}) + f(x+i^{8/3}\frac{h}{4})\big]\Big\}/(237\sqrt{3}\,h^2) \quad , \end{aligned}$$
$$E_{\text{trunc}}(h) = \frac{h^8}{3628800}f^{(10)}(x) \quad (3.32b)$$



(a) First and Second Derivative Errors (b) First and Second Derivative Errors

Figure 3.7: Comparisons of the Various Complex-Derivative Approaches

### 3.2.3 Simple Examples

Consider the following highly nonlinear function:

$$f(x) = \frac{e^x}{\sqrt{\sin^3(x) + \cos^3(x)}} \tag{3.33}$$

evaluated at $x = -0.5$. Error results for the first and second derivative approximations are shown in Figures 3.7(a). Case 1 shows results using Eqs. (3.29) and (3.27) for the first and second order derivatives, respectively. Case 2 shows results using Eqs. (3.28) and (3.24) for the first and second order derivatives, respectively. Case 3 shows results using Eqs. (3.7) and (3.10) for the first and second order derivatives, respectively. We again note that using Eq. (3.26) produces the same results as using Eq. (3.7). Using Eqs. (3.29) and (3.27) for the approximations allows one to use only one step size for all function evaluations. For this example, setting $h = 0.024750$ gives a first derivative error on the order of $10^{-16}$ and a second derivative error on the order of $10^{-15}$. Figure 3.7(b) shows results using Eqs. (3.28) and (3.27), Case A, versus results using Eqs. (3.31a) and (3.31b), Case B, for the first and

Table 3.1: Iteration Results of $x$ Using $h = 1 \times 10^{-8}$ for the Complex-Step and Finite-Difference Approaches

| Iteration | Complex-Step | Finite-Difference |
|---|---|---|
| 0 | 5.0000 | 5.0000 |
| 1 | 4.5246 | 4.4628 |
| 2 | 3.8886 | 5.1509 |
| 3 | 3.4971 | 2.6087 |
| 4 | 3.0442 | 3.2539 |
| 5 | 2.4493 | 2.5059 |
| 6 | 2.0207 | 3.2198 |
| 7 | 1.6061 | 5.2075 |
| 8 | 1.0975 | $1.3786 \times 10^{1}$ |
| 9 | $5.9467 \times 10^{-1}$ | $1.3753 \times 10^{1}$ |
| 10 | $2.9241 \times 10^{-1}$ | $1.3395 \times 10^{1}$ |
| 11 | $6.6074 \times 10^{-2}$ | $1.2549 \times 10^{1}$ |
| 12 | $1.2732 \times 10^{-3}$ | $1.2061 \times 10^{1}$ |
| 13 | $1.0464 \times 10^{-8}$ | $1.1628 \times 10^{1}$ |
| 14 | $-3.6753 \times 10^{-17}$ | $1.1583 \times 10^{1}$ |
| 15 | $-3.6753 \times 10^{-17}$ | $1.1016 \times 10^{1}$ |

second derivatives, respectively. For this example using Eqs. (3.31a) and (3.31b) provides the best overall accuracy with the least number of function evaluations for both derivatives.

Another example is given by using Halley's method for root finding. The iteration function is given by

$$x_{n+1} = x_n - \frac{2\, f(x_n)\, f'(x_n)}{2\, [f'(x_n)]^2 - f(x_n)\, f''(x_n)} \tag{3.34}$$

The following function is tested:

$$f(x) = \frac{(1 - e^x)\, e^{3x}}{\sqrt{\sin^4(x) + \cos^4(x)}} \tag{3.35}$$

which has a root at $x = 0$. Equation (3.34) is used to determine the root with a starting value of $x_0 = 5$. Equations (3.28) and (3.27) are used for the complex-step approximations. For comparison purposes the derivatives are also determined using a symmetric 4-point approximation for the first derivative and a 5-point approximation for the second derivative:

$$f^{'}(x) = \frac{f(x-2\,h) - 8\,f(x-h) + 8\,f(x+h) - f(x+2\,h)}{12h} \quad , \qquad E_{\text{trunc}}(h) = \frac{h^4}{30} f^{(5)}(x) \qquad (3.36a)$$

$$f^{''}(x) = \frac{-f(x-2\,h) + 16\,f(x-h) - 30\,f(x) + 16\,f(x+h) - f(x+2\,h)}{12\,h^2} \quad , \qquad E_{\text{trunc}}(h) = \frac{h^6}{90} f^{(6)}(x) \qquad (3.36b)$$

MATLAB® is used to perform the numerical computations. Various values of $h$ are tested in decreasing magnitude (by one order each time), starting at $h = 0.1$ and going down to $h = 1 \times 10^{-16}$. For values of $h = 0.1$ to $h = 1 \times 10^{-7}$ both methods converge, but the complex-step approach convergence is faster or (at worst) equal to the standard finite-difference approach. For values less than $1 \times 10^{-7}$, e.g. when $h = 1 \times 10^{-8}$, the finite-difference approach becomes severally degraded. Table 3.1 shows the iterations for both approaches using $1 \times 10^{-8}$. For $h$ values from $1 \times 10^{-8}$ down to $1 \times 10^{-15}$, the complex-step approach always converges in less than 15 iterations. When $h = 1 \times 10^{-16}$ the finite-difference approach produces a zero-valued correction for all iterations, while the complex-step approach converges in about 40 iterations.

### 3.2.4 Multi-Variable Numerical Example

A multi-variable example is now shown to assess the performance of the complex-step approximations. The infinity norm[‖] is used to access the accuracy of the numerical finite-difference and complex-step approximation solutions. The relationship between the magnitude of the various solutions and step-size is also discussed. The function to be tested is given by two equations with four variables:

$$\mathbf{f} \triangleq \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} x_1^2x_2x_3x_4^2 + x_2^2x_3^3x_4 \\ x_1^2x_2x_3^2x_4 + x_1x_2^3x_4^2 \end{bmatrix} \tag{3.37}$$

The Jacobian is given by

$$F_x = \begin{bmatrix} 2x_1x_2x_3x_4^2 & x_1^2x_3x_4^2 + 2x_2x_3^3x_4 & x_1^2x_2x_4^2 + 3x_2^2x_3^2x_4 & 2x_1^2x_2x_3x_4 + x_2^2x_3^3 \\ 2x_1x_2x_3^2x_4 + x_2^3x_4^2 & x_1^2x_3^2x_4 + 3x_1x_2^2x_4^2 & 2x_1^2x_2x_3x_4 & x_1^2x_2x_3^2 + 2x_1x_2^3x_4 \end{bmatrix} \tag{3.38}$$

The two Hessian matrices are given by

$$F_{xx}^1 = \begin{bmatrix} 2x_2x_3x_4^2 & 2x_1x_3x_4^2 & 2x_1x_2x_4^2 & 4x_1x_2x_3x_4 \\ 2x_1x_3x_4^2 & 2x_3^3x_4 & x_1^2x_4^2 + 6x_2x_3^2x_4 & 2x_1^2x_3x_4 + 2x_2x_3^3 \\ 2x_1x_2x_4^2 & x_1^2x_4^2 + 6x_2x_3^2x_4 & 6x_2^2x_3x_4 & 2x_1^2x_2x_4 + 3x_2^2x_3^2 \\ 4x_1x_2x_3x_4 & 2x_1^2x_3x_4 + 2x_2x_3^3 & 2x_1^2x_2x_4 + 3x_2^2x_3^2 & 2x_1^2x_2x_3 \end{bmatrix} \tag{3.39a}$$

$$F_{xx}^2 = \begin{bmatrix} 2x_2x_3^2x_4 & 2x_1x_3^2x_4 + 3x_2^2x_4^2 & 4x_1x_2x_3x_4 & 2x_1x_2x_3^2 + 2x_2^3x_4 \\ 2x_1x_3^2x_4 + 3x_2^2x_4^2 & 6x_1x_2x_4^2 & 2x_1^2x_3x_4 & x_1^2x_3^2 + 6x_1x_2^2x_4 \\ 4x_1x_2x_3x_4 & 2x_1^2x_3x_4 & 2x_1^2x_2x_4 & 2x_1^2x_2x_3 \\ 2x_1x_2x_3^2 + 2x_2^3x_4 & x_1^2x_3^2 + 6x_1x_2^2x_4 & 2x_1^2x_2x_3 & 2x_1x_2^3 \end{bmatrix} \tag{3.39b}$$

[‖] The largest row sum of a matrix $A$, $|A|_\infty = \max\{\sum |A^T|\}$.

Given $\mathbf{x} = [5,\ 3,\ 6,\ 4]^T$ the following analytical solutions are obtained:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} 14976 \\ 12960 \end{bmatrix} \tag{3.40a}$$

$$F_x = \begin{bmatrix} 2880 & 7584 & 5088 & 5544 \\ 4752 & 5760 & 3600 & 3780 \end{bmatrix} \tag{3.40b}$$

$$F_{xx}^1 = \begin{bmatrix} 576 & 960 & 480 & 1440 \\ 960 & 1728 & 2992 & 2496 \\ 480 & 2992 & 1296 & 1572 \\ 1440 & 2496 & 1572 & 900 \end{bmatrix} \tag{3.40c}$$

$$F_{xx}^2 = \begin{bmatrix} 864 & 1872 & 1440 & 1296 \\ 1872 & 1440 & 1200 & 1980 \\ 1440 & 1200 & 600 & 900 \\ 1296 & 1980 & 900 & 270 \end{bmatrix} \tag{3.40d}$$

### Numerical Solutions

The step-size for the Jacobian and Hessian calculations (both for complex-step approximation and numerical finite difference) is $1 \times 10^{-4}$. The absolute Jacobian error between the true and complex-step solutions, and true and numerical finite-difference solutions, respectively, are

$$|\Delta^c F_x| = \begin{bmatrix} 0.0000 & 0.0000 & 0.3600 & 0.0000 \\ 0.0000 & 0.8000 & 0.0000 & 0.0000 \end{bmatrix} \times 10^{-8} \tag{3.41a}$$

$$|\Delta^n F_x| = \begin{bmatrix} 0.2414 & 0.3348 & 0.0485 & 0.1074 \\ 0.1051 & 0.4460 & 0.0327 & 0.0298 \end{bmatrix} \times 10^{-7} \tag{3.41b}$$

The infinity norms of Eq. (3.41) are $8.0008 \times 10^{-9}$ and $7.3217 \times 10^{-8}$, respectively, which means that the complex-step solution is more accurate than the finite-difference one. The absolute Hessian error between the true solutions and the complex-step and numerical finite-difference solutions, respectively, are

$$|\Delta^c F_{xx}^1| = \begin{bmatrix} 0.0000 & 0.0011 & 0.0040 & 0.0016 \\ 0.0011 & 0.0010 & 0.0011 & 0.0009 \\ 0.0040 & 0.0011 & 0.0019 & 0.0021 \\ 0.0016 & 0.0009 & 0.0021 & 0.0004 \end{bmatrix} \tag{3.42a}$$

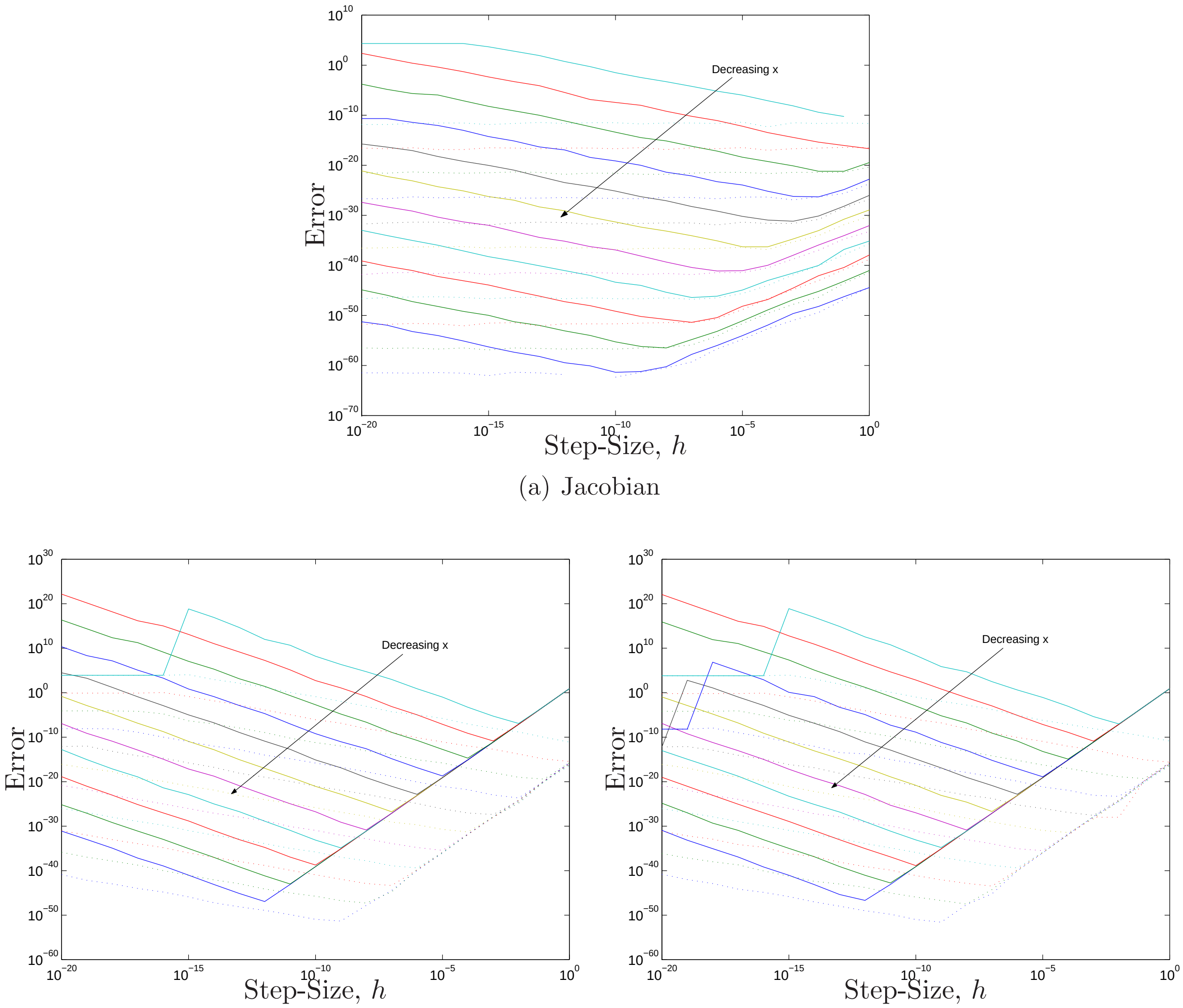
$$|\Delta^n F_{xx}^1| = \begin{bmatrix} 0.0002 & 0.0010 & 0.0041 & 0.0017 \\ 0.0010 & 0.0009 & 0.0011 & 0.0011 \\ 0.0041 & 0.0011 & 0.0021 & 0.0019 \\ 0.0017 & 0.0011 & 0.0019 & 0.0003 \end{bmatrix} \tag{3.42b}$$

and

$$|\Delta^c F_{xx}^2| = \begin{bmatrix} 0.0018 & 0.0007 & 0.0030 & 0.0064 \\ 0.0007 & 0.0016 & 0.0010 & 0.0018 \\ 0.0030 & 0.0010 & 0.0018 & 0.0004 \\ 0.0064 & 0.0018 & 0.0004 & 0.0029 \end{bmatrix} \tag{3.43a}$$

$$|\Delta^n F_{xx}^2| = \begin{bmatrix} 0.0018 & 0.0007 & 0.0031 & 0.0065 \\ 0.0007 & 0.0015 & 0.0008 & 0.0021 \\ 0.0031 & 0.0008 & 0.0018 & 0.0006 \\ 0.0065 & 0.0021 & 0.0006 & 0.0025 \end{bmatrix} \tag{3.43b}$$

The infinity norms of Eq. (3.42) are $9.0738 \times 10^{-3}$ and $9.1858 \times 10^{-3}$, respectively, and the infinity norms of Eq. (3.43) are $1.1865 \times 10^{-3}$ and $1.2103 \times 10^{-3}$, respectively. As with the Jacobian, the complex-step Hessian approximation solutions are more accurate than the finite difference solutions.



(a) Jacobian

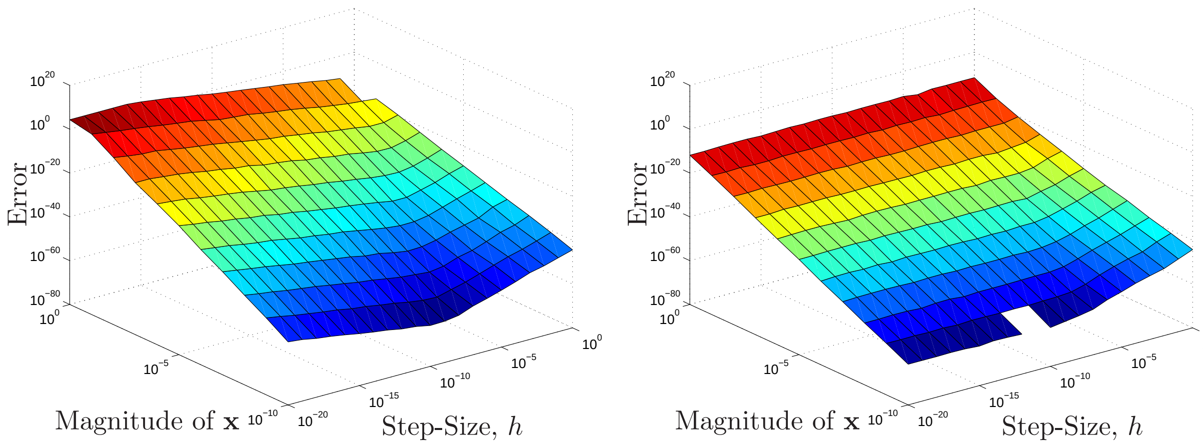
(b) Hessian 1

(c) Hessian 2

Figure 3.8: Infinity Norm of the Error Matrix for Different Magnitudes (Solid Lines = Finite Difference, Dotted Lines = Complex-Step)

## Performance Evaluation

The performance of the complex-step approach in comparison to the numerical finite-difference approach is examined further here using the same function. Tables 3.2 and 3.3 shows the



(a) Jacobian - Finite-Difference

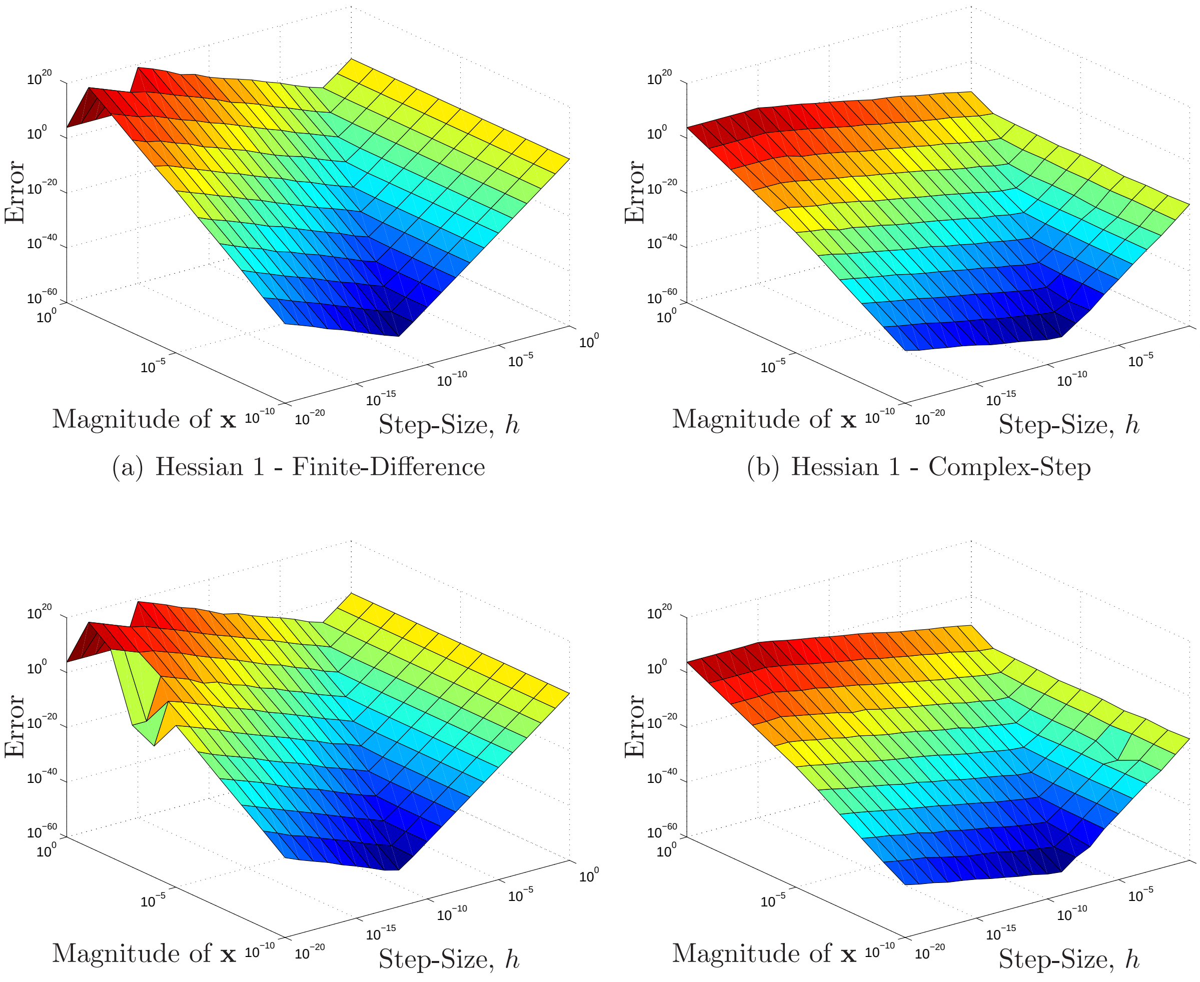
(b) Jacobian - Complex-Step

Figure 3.9: Infinity Norm of the Jacobian Error Matrix for Different Magnitudes and Step-Sizes

infinity norm of the error between the true and the approximated solutions. The difference between the finite difference solution and the complex-step solution is also included in the last three rows, where positive values indicate the complex-step solution is more accurate. In most cases, the complex-step approach performs either comparable or better than the finite-difference approach. The complex-step approach provides accurate solutions for $h$ values from 0.1 down to $1 \times 10^{-9}$. However, the range of accurate solutions for the finite-difference approach is significantly smaller than that of complex-step approach. Clearly, the complex-step approach is much more robust than the numerical finite-difference approach.

Figure 3.8 shows plots of the infinity norm of the Jacobian and Hessian errors obtained using a numerical finite-difference and the complex-step approximation. The function is evaluated at different magnitudes by multiplying the nominal values with a scale factor from 1 down to $1 \times 10^{-10}$. The direction of the arrow shows the solutions for decreasing $\mathbf{x}$. The solutions for the complex-step and finite-difference approximation using the same $\mathbf{x}$ value are plotted with the same color within a plot.

For the case of the finite-difference Jacobian, shown in Figure 3.8(a), at some cer-



(a) Hessian 1 - Finite-Difference

(b) Hessian 1 - Complex-Step

(c) Hessian 2 - Finite-Difference

(d) Hessian 2 - Complex-Step

Figure 3.10: Infinity Norm of the Hessian Error Matrix for Different Magnitudes and Step-Sizes

tain point of decreasing step-size, as mentioned before, the subtraction cancellation error dominates, which decreases the accuracy. The complex-step solution does not exhibit this phenomenon and the accuracy continues to increase with decreasing step-size up to machine precision. As a higher-order complex-step approximation is used, Eq. (3.31a) instead of Eq. (3.7), the truncation errors for the complex-step Jacobian at larger step-sizes are also greatly reduced to the extent that the truncation errors are almost unnoticeable, even at large $\mathbf{x}$ values. The complex-step approximation for the Hessian case also benefits from the

Table 3.2: Infinity Norm of the Difference from Truth for Larger Step-Sizes, $h$

| $h$ | $1 \times 10^{0}$ | $1 \times 10^{-1}$ | $1 \times 10^{-2}$ | $1 \times 10^{-3}$ | $1 \times 10^{-4}$ |
|---|---|---|---|---|---|
| $\lvert\Delta^n F_x\rvert$ | $8.0004 \times 10^{-9}$ | $8.0554 \times 10^{-9}$ | $8.2664 \times 10^{-9}$ | $9.6984 \times 10^{-9}$ | $7.3218 \times 10^{-8}$ |
| $\lvert\Delta^c F_x\rvert$ | $8.0026 \times 10^{-9}$ | $8.0004 \times 10^{-9}$ | $8.0013 \times 10^{-9}$ | $8.0026 \times 10^{-9}$ | $8.0008 \times 10^{-9}$ |
| $\lvert\Delta^n F^1_{xx}\rvert$ | $8.0000$ | $9.1000 \times 10^{-3}$ | $9.1000 \times 10^{-3}$ | $9.1000 \times 10^{-3}$ | $9.2000 \times 10^{-3}$ |
| $\lvert\Delta^c F^1_{xx}\rvert$ | $9.1000 \times 10^{-3}$ | $9.1000 \times 10^{-3}$ | $9.1000 \times 10^{-3}$ | $9.1000 \times 10^{-3}$ | $9.1000 \times 10^{-3}$ |
| $\lvert\Delta^n F^2_{xx}\rvert$ | $7.9990$ | $1.1100 \times 10^{-2}$ | $1.1900 \times 10^{-2}$ | $1.1900 \times 10^{-2}$ | $1.2100 \times 10^{-2}$ |
| $\lvert\Delta^c F^2_{xx}\rvert$ | $1.1900 \times 10^{-2}$ | $1.1900 \times 10^{-2}$ | $1.1900 \times 10^{-2}$ | $1.1900 \times 10^{-2}$ | $1.1900 \times 10^{-2}$ |
| $\lvert\Delta^n F_x\rvert - \lvert\Delta^c F_x\rvert$ | $-2.2737 \times 10^{-12}$ | $5.5024 \times 10^{-11}$ | $2.6512 \times 10^{-10}$ | $1.6958 \times 10^{-9}$ | $6.5217 \times 10^{-8}$ |
| $\lvert\Delta^n F^1_{xx}\rvert - \lvert\Delta^c F^1_{xx}\rvert$ | $7.9909$ | $-5.0477 \times 10^{-11}$ | $3.5698 \times 10^{-10}$ | $1.5272 \times 10^{-6}$ | $1.1200 \times 10^{-4}$ |
| $\lvert\Delta^n F^2_{xx}\rvert - \lvert\Delta^c F^2_{xx}\rvert$ | $7.9871$ | $-8.0000 \times 10^{-4}$ | $-6.5184 \times 10^{-8}$ | $-5.3940 \times 10^{-8}$ | $2.3823 \times 10^{-4}$ |

higher-order approximation, as shown in Figures 3.8(b) and 3.8(c). The complex-step Hessian approximation used to generate these results is given by Eq. (3.31b). One observation is that there is always only one (global) optimum of specific step-size with respect to the error.

Figures 3.9 and 3.10 represent the same information in more intuitive looking three-dimensional plots. The "depth" of the error in log scale is represented as a color scale with dark red being the highest and dark blue being the lowest. A groove is clearly seen in most of the plots (except the complex-step Jacobian), which corresponds to the optimum step-size. The "empty surface" in Figure 3.9 corresponds to when the difference between the complex-step solution and the truth is below machine precision. This is shown as "missing line" in Figure 3.8(a). Clearly, the complex-step approximation solutions are comparable or

Table 3.3: Infinity Norm of the Difference from Truth for Smaller Step-Sizes, $h$

| $h$ | $1 \times 10^{-5}$ | $1 \times 10^{-6}$ | $1 \times 10^{-7}$ |
|---|---|---|---|
| $\lvert\Delta^n F_x\rvert$ | $1.0133 \times 10^{-6}$ | $6.4648 \times 10^{-6}$ | $5.8634 \times 10^{-5}$ |
| $\lvert\Delta^c F_x\rvert$ | $8.0026 \times 10^{-9}$ | $8.0004 \times 10^{-9}$ | $8.0026 \times 10^{-9}$ |
| $\lvert\Delta^n F^1_{xx}\rvert$ | $1.0160 \times 10^{-1}$ | $7.6989$ | $9.5627 \times 10^{2}$ |
| $\lvert\Delta^c F^1_{xx}\rvert$ | $9.1000 \times 10^{-3}$ | $9.1000 \times 10^{-3}$ | $9.1000 \times 10^{-3}$ |
| $\lvert\Delta^n F^2_{xx}\rvert$ | $7.3500 \times 10^{-2}$ | $4.2094$ | $3.1084 \times 10^{2}$ |
| $\lvert\Delta^c F^2_{xx}\rvert$ | $1.1900 \times 10^{-2}$ | $1.1900 \times 10^{-2}$ | $1.1700 \times 10^{-2}$ |
| $\lvert\Delta^n F_x\rvert - \lvert\Delta^c F_x\rvert$ | $1.0053 \times 10^{-6}$ | $6.4568 \times 10^{-6}$ | $5.8626 \times 10^{-5}$ |
| $\lvert\Delta^n F^1_{xx}\rvert - \lvert\Delta^c F^1_{xx}\rvert$ | $9.2500 \times 10^{-2}$ | $7.6898$ | $9.5626 \times 10^{2}$ |
| $\lvert\Delta^n F^2_{xx}\rvert - \lvert\Delta^c F^2_{xx}\rvert$ | $6.1600 \times 10^{-2}$ | $4.1976$ | $3.1082 \times 10^{2}$ |

| $h$ | $1 \times 10^{-8}$ | $1 \times 10^{-9}$ | $1 \times 10^{-10}$ |
|---|---|---|---|
| $\lvert\Delta^n F_x\rvert$ | $5.0732 \times 10^{-4}$ | $3.5000 \times 10^{-3}$ | $3.1200 \times -2$ |
| $\lvert\Delta^c F_x\rvert$ | $8.0013 \times 10^{-9}$ | $7.9995 \times 10^{-9}$ | $7.9999 \times -9$ |
| $\lvert\Delta^n F^1_{xx}\rvert$ | $5.2882 \times 10^{4}$ | $2.2007 \times 10^{6}$ | $1.5916 \times 8$ |
| $\lvert\Delta^c F^1_{xx}\rvert$ | $9.1000 \times 10^{-3}$ | $1.4800 \times 10^{-2}$ | $1.2730 \times -1$ |
| $\lvert\Delta^n F^2_{xx}\rvert$ | $4.9658 \times 10^{4}$ | $7.6182 \times 10^{5}$ | $2.4253 \times 8$ |
| $\lvert\Delta^c F^2_{xx}\rvert$ | $1.3500 \times 10^{-2}$ | $8.8000 \times 10^{-3}$ | $9.9700 \times -2$ |
| $\lvert\Delta^n F_x\rvert - \lvert\Delta^c F_x\rvert$ | $5.0731 \times 10^{-4}$ | $3.5000 \times 10^{-3}$ | $3.1200 \times -2$ |
| $\lvert\Delta^n F^1_{xx}\rvert - \lvert\Delta^c F^1_{xx}\rvert$ | $5.2882 \times 10^{4}$ | $2.2007 \times 10^{6}$ | $1.5916 \times 8$ |
| $\lvert\Delta^n F^2_{xx}\rvert - \lvert\Delta^c F^2_{xx}\rvert$ | $4.9658 \times 10^{4}$ | $7.6182 \times 10^{5}$ | $2.4253 \times 8$ |

more accurate than the finite-difference solutions.

## 3.3 Conclusion

This chapter demonstrated the ability of numerically obtaining derivative information via a complex-step approximations. For the Jacobian case, unlike standard derivative approaches, more control in the accuracy of the standard complex-step approximation is provided since it does not succumb to roundoff errors for small step sizes. For the Hessian case, however, an arbitrarily small step size cannot be chosen due to roundoff errors. Also, using the standard complex-step approach to approximate second derivatives was found to be less accurate than the numerical finite-difference obtained one. The accuracy was improved by deriving a number of new complex-step approximations for both the first and second derivatives. These new approximations allow for high accuracy results in both the Jacobian and Hessian approximations by using the same function evaluations and step sizes for both. The main advantage of the approach presented in this dissertation is that a “black box” can be employed to obtain the Jacobian or Hessian matrices for any vector function.

# Chapter 4

# State Estimation

## 4.1 Introduction

This chapter briefly discusses various state estimators. Estimation is defined as extraction of useful information from single/multiple noisy sensor measurements with reference to an assumed mathematical plant model. These filters can be derived in various ways but only one of each will be presented here. We begin with the celebrated Kalman filter (KF). After the derivation of the KF in 1960 [39] Apollo scientists quickly realized its potential. However, most of the real world dynamics are essentially nonlinear in nature, including the highly nonlinear attitude and orbital dynamics. Thus a nonlinear version of KF was developed shortly after. A brief history on the extended Kalman filter (EKF) can be found in Ref. [40]. We also present a new approach called the Unscented filter. The Unscented filter promises more accurate estimates with a lower expected error covariance than the EKF. It linearizes to higher order than the standard EKF.

State estimation is needed to “filter” out or to minimize the effect of sensor noise. It is the fusion of various noisy measurement data to obtain (estimate) information about the system states with the help from an assumed system model. Also, as in most of the time, the sensor does not measure all the state components or does not measure the state components directly. Thus an estimator is needed to obtain the missing information that is often important from control point of view.

If the system and sensor/measurement models are “valid” (high correlationship between the model and the “truth”), with adequate observability and appropriate assumptions (for example the dominance of first order series expansion to the EKF), the filter should converge towards the true states with decreasing estimated error covariance. However, there is a limit to this filter until a balance is reached between gaining new information from new measurements and loss of information from propagation.

## 4.2 Kalman Filter

### 4.2.1 Discrete Kalman Filter

The truth model is given by

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \Upsilon_k \mathbf{w}_k \tag{4.1a}$$

$$\tilde{\mathbf{y}}_k = H_k \mathbf{x}_k + \mathbf{v}_k \tag{4.1b}$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero-mean Gaussian white-noise (with no memory, uncorrelated, with the past and future noise) with covariances $Q_k$ and $R_k$ respectively,

$$\mathrm{E}\{\mathbf{w}_k\} = \mathbf{0} \quad , \qquad \mathrm{E}\{\mathbf{w}_j \mathbf{w}_k^T\} = \delta_{jk} Q_k \tag{4.2a}$$

$$\mathrm{E}\{\mathbf{v}_k\} = \mathbf{0} \quad , \qquad \mathrm{E}\{\mathbf{v}_j \mathbf{v}_k^T\} = \delta_{jk} R_k \tag{4.2b}$$

where E{·} denotes the expectation operator and $\delta_{jk}$ is the Kronecker function

$$\delta_{jk} = \begin{cases} 0 & \text{when } j \neq k \\ 1 & \text{when } j = k \end{cases} \tag{4.3}$$

The cross-correlation covariance is assumed to follow

$$\mathrm{E}\{\mathbf{w}_j \mathbf{v}_k^T\} = \mathbf{0} \quad , \qquad\qquad \forall j, k \tag{4.4}$$

Thus, they are uncorrelated with each other. The vector $\mathbf{w}_k$ is also known as the process noise, it indicates how accurately we know the assumed model or how much the model changes with time. Unfortunately, this usually is not known precisely and some "tuning" effort is required to obtain the desired performance. The vector $\mathbf{v}_k$ is the measurement noise. It indicates the accuracy of measurements from the sensor. This is usually easily determined or specified by the sensor manufacturer. Let the new state estimate be a linear combination of the propagated values and the sensors measurements:

$$\hat{\mathbf{x}}_k^+ = K_k' \hat{\mathbf{x}}_k^- + K_k \tilde{\mathbf{y}}_k \tag{4.5}$$

where the superscript $-$ indicates propagated/pre-update value and $+$ indicates updated value. Define the error states as

$$\tilde{\mathbf{x}}_k^- = \hat{\mathbf{x}}_k^- - \mathbf{x}_k \tag{4.6a}$$

$$\tilde{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^+ - \mathbf{x}_k \tag{4.6b}$$

note that a tilde sign over $\mathbf{y}$ denotes measurement but a tilde over $\mathbf{x}$ denotes error. From Eq. (4.6b) and with the substitution of Eqs. (4.5), (4.1b) and (4.6a) for $\hat{\mathbf{x}}_k^+$, we have

$$\tilde{\mathbf{x}}_k^+ = (K_k' + K_k H_k - I)\mathbf{x}_k + K_k \mathbf{v}_k + K_k' \tilde{\mathbf{x}}_k^- \tag{4.7}$$

Taking the expectation of the above equation and setting the expectation of the error states to zero for an unbiased estimator (note that $\mathbf{x}_k$ is not a random variable) leads to $K_k' + K_k H_k - I = 0$ or

$$K_k' = I - K_k H_k \tag{4.8}$$

Substituting this result into Eq. (4.5) gives

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\tilde{\mathbf{y}}_k - H_k \hat{\mathbf{x}}_k^-) \tag{4.9}$$

Substituting Eqs. (4.9) and (4.1b) into Eq. (4.6b) gives

$$\begin{aligned}
\tilde{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + K_k(H_k \mathbf{x}_k + \mathbf{v}_k - H_k \hat{\mathbf{x}}_k^-) - \mathbf{x}_k \\
&= (I - K_k H_k)\hat{\mathbf{x}}_k^- - (I - K_k H_k)\mathbf{x}_k + K_k \mathbf{v}_k \\
&= (I - K_k H_k)\tilde{\mathbf{x}}_k^- + K_k \mathbf{v}_k \qquad (4.10)
\end{aligned}$$

Now let's divert our attention to the error covariance. The state propagation equation is simply given by taking the expectation of Eq. (4.1a):

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ + \Gamma_k \mathbf{u}_k \tag{4.11}$$

Substituting Eqs. (4.11) and (4.1a) into the one time-step ahead of Eq. (4.6a) gives

$$\begin{aligned}\tilde{\mathbf{x}}_{k+1}^- &= \Phi_k\hat{\mathbf{x}}_k^+ - \Phi_k\mathbf{x}_k - \Gamma_k\mathbf{w}_k \\ &= \Phi_k\tilde{\mathbf{x}}_k^+ - \Gamma_k\mathbf{w}_k \end{aligned} \tag{4.12}$$

note that Eq. (4.12) is not a function of the control input $\mathbf{u}_k$, it is assumed to be a known or a deterministic property instead of a random variable. Now define the pre-update and updated error covariances:

$$P_k^- \equiv \mathrm{E}\{\tilde{\mathbf{x}}_k^-\tilde{\mathbf{x}}_k^{-T}\} \tag{4.13a}$$

$$P_k^+ \equiv \mathrm{E}\{\tilde{\mathbf{x}}_k^+\tilde{\mathbf{x}}_k^{+T}\} \tag{4.13b}$$

Substituting Eq. (4.12) into the one time-step ahead of Eq. (4.13a) and assuming the statistical independence of $\mathbf{w}_k$ and $\tilde{\mathbf{x}}_k^+$, $\mathrm{E}\{\mathbf{w}_k\tilde{\mathbf{x}}_k^{+T}\} = \mathrm{E}\{\tilde{\mathbf{x}}_k^+\mathbf{w}_k^T\} = 0$ (this is due to the fact that $\mathbf{w}_k$ is correlated with $\tilde{\mathbf{x}}_{k+1}$ instead of $\tilde{\mathbf{x}}_k$ as obvious from Eq. (4.1a)) leads to

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T \tag{4.14}$$

We then apply Eq. (4.10) into Eq. (4.13b) and again with statistical independence assumption between $\mathbf{v}_k$ and $\tilde{\mathbf{x}}_k^-$, $\mathrm{E}\{\mathbf{v}_k\tilde{\mathbf{x}}_k^{-T}\} = \mathrm{E}\{\tilde{\mathbf{x}}_k^-\mathbf{v}_k^T\} = 0$ (as obvious from Eq. (4.7) that $\mathbf{v}_k$ is directly correlated with $\tilde{\mathbf{x}}_k^+$ instead of $\tilde{\mathbf{x}}_k^-$) yields

$$P_k^+ = (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \tag{4.15}$$

There are numerous ways to capture the overall deviation of the filter estimates from the truth. The diagonal terms of the error covariance correspond to the error covariance of each state, thus, one way is to minimize the sum of the diagonal terms, or trace, of the error covariance. Let's define the cost function as

$$J(K_k) = \mathrm{Tr}[P_k^+] \tag{4.16}$$

We take the derivative of this cost function and set it to zero to find the extremal. Also, $\frac{\partial}{\partial A}[Tr(ABA^T)] = 2AB$ for symmetric $B$. Thus, the partial derivative of Eq. (4.16) (note that the error covariance matrices are symmetric) is

$$\frac{\partial}{\partial K_k} J(K_k) = 0 = -2(I - K_k H_k) P_k^- H_k^T + 2K_k R_k \tag{4.17}$$

Solving the above equation for $K_k$ gives

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \tag{4.18}$$

Expanding Eq. (4.13b) leads to

$$P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k [H_k P_k^- H_k^T + R_k] K_k^T \tag{4.19}$$

Substituting Eq. (4.18) for the $K_k$ before the square bracket yields

$$\begin{aligned} P_k^+ &= P_k^- - K_k H_k P_k^- \qquad (4.20) \\ &= [I - K_k H_k] P_k^- \qquad (4.21) \end{aligned}$$

with initial condition given by $P_0^+ = \mathrm{E}\{\tilde{\mathbf{x}}_0^+ \tilde{\mathbf{x}}_0^{+T}\}$. As long as $R_k$ is positive definite and $Q_k$ semi-definite, the linear KF is always stable. A summary of the discrete KF algorithm is given in Table 4.1

Table 4.1: Discrete Kalman Filter

| | |
|---|---|
| Model | $\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \Upsilon_k \mathbf{w}_k$<br>$\tilde{\mathbf{y}}_k = H_k \mathbf{x}_k + \mathbf{v}_k$ |
| Initialize | $\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$<br>$P_0 = \mathrm{E}\{\tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T\}$ |
| Gain | $K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$ |
| Update | $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\tilde{\mathbf{y}}_k - H_k \hat{\mathbf{x}}_k^-)$<br>$P_k^+ = [I - K_k H_k] P_k^-$ |
| Propagation | $\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ + \Gamma_k \mathbf{u}_k$<br>$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T$ |

## 4.3 Extended Kalman Filter

### 4.3.1 Continuous-Discrete Extended Kalman Filter

The continuous nonlinear model equation with discrete-time measurements is

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + G(t)\mathbf{w}(t) \tag{4.22a}$$

$$\tilde{\mathbf{y}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \tag{4.22b}$$

The noise terms are again

$$\mathrm{E}\{\mathbf{w}(t)\} = \mathbf{0} \quad , \qquad \mathrm{E}\{\mathbf{w}(\tau)\mathbf{w}(t)^T\} = Q(t)\delta(t-\tau) \tag{4.23a}$$

$$\mathrm{E}\{\mathbf{v}_k\} = \mathbf{0} \quad , \qquad \mathrm{E}\{\mathbf{v}_j \mathbf{v}_k^T\} = \delta_{jk} R_k \tag{4.23b}$$

where $Q(t)$ is the power spectral density matrix of the process noise and $\delta(t-\tau)$ is the Dirac delta function. Applying a Taylor's series expansion on the nonlinear terms in Eq. (4.22) yields

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} (\mathbf{x} - \hat{\mathbf{x}}) + \text{h.o.t.} \tag{4.24a}$$

$$\mathbf{h}_k(\mathbf{x}_k) = \mathbf{h}_k(\hat{\mathbf{x}}_k^-) + \left.\frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \text{h.o.t.} \tag{4.24b}$$

where h.o.t. abbreviates "higher order terms". It is assumed that if the estimate is close to the truth, then the first order term of the Taylor's series expansion dominates. Assuming that the partial derivatives exist, we define

$$F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \equiv \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad , \qquad H_k(\hat{\mathbf{x}}_k^-) \equiv \left.\frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \tag{4.25}$$

Note that $F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)$ is expanded about the current estimate (conditional mean) while $H_k(\hat{\mathbf{x}}_k^-, t)$ is expanded about the propagated states. Neglecting the h.o.t., then Eq. (4.24) becomes

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \simeq \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)(\mathbf{x} - \hat{\mathbf{x}}) \tag{4.26a}$$

$$\mathbf{h}_k(\mathbf{x}_k) \simeq \mathbf{h}_k(\hat{\mathbf{x}}_k^-) + H_k(\hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \tag{4.26b}$$

Taking the expectation of Eq. (4.26a) yields

$$\hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t), t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \tag{4.27}$$

Taking the expectation of Eq. (4.22a) and with Eq. (4.27) leads to the propagation equation for the state vector between measurement times, i.e. $t_{k-1} \leq t < t_k$

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \tag{4.28}$$

The continuous error covariance propagation equation is given by

$$\dot{P}(t) = F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)P(t) + P(t)F^T(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + G(t)Q(t)G^T(t) \tag{4.29}$$

Now let's consider the update equations for the state vector and error covariance. Motivated by the linear update equation of Eq. (4.5) we again have

$$\hat{\mathbf{x}}_k^+ = K_k'\hat{\mathbf{x}}_k^- + K_k\tilde{\mathbf{y}}_k \tag{4.30}$$

and the definition of the error states of Eq. (4.6)

$$\tilde{\mathbf{x}}_k^- = \hat{\mathbf{x}}_k^- - \mathbf{x}_k \tag{4.31a}$$

$$\tilde{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^+ - \mathbf{x}_k \tag{4.31b}$$

Substituting Eq. (4.22b) into Eq. (4.30) and then the resultant into Eq. (4.31b) and with Eq. (4.31a) for $-\mathbf{x}_k$ leads to

$$\tilde{\mathbf{x}}_k^+ = K_k'\hat{\mathbf{x}}_k^- + K_k\mathbf{h}_k(\mathbf{x}_k) + K_k\mathbf{v}_k + \tilde{\mathbf{x}}_k^- - \hat{\mathbf{x}}_k^- \tag{4.32}$$

Taking the expectation of Eq. (4.32) with $\mathrm{E}\{\tilde{\mathbf{x}}_k^+\} = \mathrm{E}\{\tilde{\mathbf{x}}_k^-\} = \mathrm{E}\{\mathbf{v}_k\} = 0$ and solving for $K_k'\tilde{\mathbf{x}}_k^-$ yields

$$K_k'\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_k^- - K_k\hat{\mathbf{h}}_k(\mathbf{x}_k) \tag{4.33}$$

Substituting this into Eq. (4.30) gives

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - \hat{\mathbf{h}}_k(\mathbf{x}_k)] \tag{4.34}$$

Substituting Eq. (4.33) into Eq. (4.32) and using the approximation of Eq. (4.26b) yields

$$\begin{aligned}\tilde{\mathbf{x}}_k^+ &= \tilde{\mathbf{x}}_k^- + K_k[\mathbf{h}_k(\mathbf{x}_k) - \hat{\mathbf{h}}_k(\mathbf{x}_k)] + K_k\mathbf{v}_k \\ &= \tilde{\mathbf{x}}_k^- + K_k[\mathbf{h}_k(\hat{\mathbf{x}}_k^-) + H_k(\hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \hat{\mathbf{h}}_k(\mathbf{x}_k)] + K_k\mathbf{v}_k \\ &= \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)] \qquad (4.35)\end{aligned}$$

The Kalman gain $K_k$ is determined in the same fashion as before. Again, we choose to minimize the mean-square-error (MSE) by minimizing the trace of the error covariance matrix. Again, the definition of error covariances are

$$P_k^- \equiv \mathrm{E}\{\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{-T}\} \tag{4.36a}$$

$$P_k^+ \equiv \mathrm{E}\{\tilde{\mathbf{x}}_k^+ \tilde{\mathbf{x}}_k^{+T}\} \tag{4.36b}$$

Using Eq. (4.35) in Eq. (4.36b) with the assumption that $P_k^+$ and $\tilde{\mathbf{y}}_k$ are independent from each other gives

$$\begin{aligned}P_k^+ &= P_k^- + K_k\mathrm{E}\{[\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)][\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]^T\}K_k^T \\ &\quad + \mathrm{E}\{\tilde{\mathbf{x}}_k^-[\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]^T\} + K_k\mathrm{E}\{[\mathbf{h}_k(\mathbf{x}_k) \qquad (4.37) \\ &\quad - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]\tilde{\mathbf{x}}_k^{-T}\} + K_k R_k K_k^T\end{aligned}$$

The cost function is to minimize the trace of $P_k^+$:

$$J(K_k) = \mathrm{Tr}[P_k^+] \tag{4.38}$$

Taking the partial derivative gives

$$\frac{\partial}{\partial K_k} J(K_k) = 0 \tag{4.39}$$

and solving for $K_k$ with approximation of Eq. (4.26b) yields

$$\begin{aligned} K_k &= -\mathrm{E}\{\tilde{\mathbf{x}}_k^-[\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]^T\} \\ &\quad \times \left[\mathrm{E}\{[\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)][\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]^T\} + R_k\right]^{-1} \\ &= P_k^- H_k^T(\hat{\mathbf{x}}_k^-)[H_k(\hat{\mathbf{x}}_k^-)P_k^- H_k^T(\hat{\mathbf{x}}_k^-) + R_k]^{-1} \end{aligned} \tag{4.40}$$

Substituting Eq. (4.40) into Eq. (4.37) with approximation of Eq. (4.26b) *mutatis mutandis* leads to

$$\begin{aligned} P_k^+ &= P_k^- + K_k \mathrm{E}\{[\mathbf{h}_k(\mathbf{x}_k) - \tilde{\mathbf{h}}_k(\mathbf{x}_k)]\tilde{\mathbf{x}}_k^{-T}\} \\ &= [I - K_k H_k(\hat{\mathbf{x}}_k^-)]P_k^- \end{aligned} \tag{4.41}$$

A Summary of the Continuous-Discrete EKF is presented in Table 4.2. The Discrete EKF is given in Table 4.3. Unlike the linear KF which can be proven to be globally stable by using Lyapunov's direct method [41], the stability of EKF is not guaranteed. Care must be exercised in initializing the filter as large deviations from the truth could potentially diverge the filter. This is due to the violation of the first-order linearization assumption when the estimation error becomes large.

Table 4.2: Continuous-Discrete Extended Kalman Filter

| | |
|---|---|
| Model | $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + G(t)\mathbf{w}(t)$<br>$\tilde{\mathbf{y}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$ |
| Initialize | $\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$<br>$P_0 = \mathrm{E}\{\tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T\}$ |
| Gain | $K_k = P_k^- H_k^T(\hat{\mathbf{x}}_k^-)[H_k(\hat{\mathbf{x}}_k^-) P_k^- H_k^T(\hat{\mathbf{x}}_k^-) + R_k]^{-1}$<br>$H_k(\hat{\mathbf{x}}_k^-) \equiv \left. \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \right\vert_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$ |
| Update | $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k)]$<br>$P_k^+ = [I - K_k H_k(\hat{\mathbf{x}}_k^-)]P_k^-$ |
| Propagation | $\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)$<br>$\dot{P}(t) = F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)P(t) + P(t)F^T(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + G(t)Q(t)G^T(t)$<br>$F(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \equiv \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right\vert_{\mathbf{x}=\hat{\mathbf{x}}}$ |

## 4.4 Unscented Filter

In this section the filter first proposed in [42] is presented. This new filter, which they called the *Unscented filter* (UF) or *Unscented Kalman filter* (UKF), works on the premise that it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function given a fixed number of parameters [42].

The EKF basically linearizes the system and measurement models to first order with Taylor's series expansion, then simplies update them with a linear KF. This "first-order" assumption usually works well when the sampling time is small, and when the system behaves in a quasi-linear manner. However, this linearized propagation could easily cause an instability in the filter, which leads to divergence of the filter. The nonlinear transformation, the heart of the UF, called the *unscented transformation* (UT), propagates the states with a

Table 4.3: Discrete Extended Kalman Filter

| | |
|---|---|
| Model | $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) + G_k \mathbf{w}_k$<br>$\tilde{\mathbf{y}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$ |
| Initialize | $\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$<br>$P_0 = \mathrm{E}\{\tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T\}$ |
| Gain | $K_k = P_k^- H_k^T(\hat{\mathbf{x}}_k^-)[H_k(\hat{\mathbf{x}}_k^-) P_k^- H_k^T(\hat{\mathbf{x}}_k^-) + R_k]^{-1}$<br>$H_k(\hat{\mathbf{x}}_k^-) \equiv \left. \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \right\vert_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$ |
| Update | $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k)]$<br>$P_k^+ = [I - K_k h_k(\hat{\mathbf{x}}_k^-)] P_k^-$ |
| Propagation | $\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k, \mathbf{u}_k, k)$<br>$P_{k+1}^- = \Phi_k P_k^+ \Phi_k + G_k Q_k G_k^T$ |

minimal set of carefully selected sample points through a nonlinear transformation that captures the posterior mean and covariance accurately to higher order Taylor series expansion [43].

The discrete-time system and measurement models with additive noises are assumed to be

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + G_k \mathbf{w}_k \tag{4.42a}$$

$$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k \tag{4.42b}$$

where $\mathbf{x}_k$ is the $n \times 1$ state vector; $\tilde{\mathbf{y}}_k$ is the $m \times 1$ measurement vector; $\mathbf{w}_k$ is the process noise and $\mathbf{v}_k$ is the measurement noise. As before, we further assume both the process noise and the measurement noise are zero-mean Gaussian noise processes with covariances $Q_k$ and $R_k$ respectively, as given by Eqs. (4.2) and (4.4). The KF update scheme is first rewritten as

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \boldsymbol{\upsilon}_k \tag{4.43a}$$

$$P_k^+ = P_k^- - K_k P_k^{\upsilon\upsilon} K_k^T \tag{4.43b}$$

where $\boldsymbol{\upsilon}_k$ is the *innovation* given by

$$\boldsymbol{\upsilon}_k \equiv \tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^- = \tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-,\, k) \tag{4.44}$$

and $P_k^{\upsilon\upsilon}$ is the covariance of $\boldsymbol{\upsilon}_k$. The Kalman gain $K_k$ in this case is then given by

$$K_k = P_k^{xy}(P_k^{\upsilon\upsilon})^{-1} \tag{4.45}$$

where $P_k^{xy}$ is the cross-correlation matrix between $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{y}}_k^-$.

As mentioned before, the core of the UF is its different propagation that more accurately captures the mean and covariance during the propagation phase with the UT. Instead of a Monte-Carlo style approach to retain the true mean and covariance, the UT generates a discrete distribution with minimum and carefully selected *sigma points* that have the same first and second and possible higher moments. Reference [44] shows that these finite points estimate the true mean and covariance better than the EKF without a computational-burdensome Monte-Carlo run. With an $n$-dimensional Gaussian distribution with error covariance matrix $P$, first generate a set of $2n$ sigma points

$$\boldsymbol{\sigma}_k \leftarrow 2n \text{ columns from } \pm\sqrt{(n+\lambda)[P_k^+ + Q_k]} \tag{4.46a}$$

$$\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k^+ \tag{4.46b}$$

$$\boldsymbol{\chi}_k(i) = \boldsymbol{\sigma}_k(i) + \hat{\mathbf{x}}_k^+ \tag{4.46c}$$

that have the same covariance from the columns or rows. These sigma points has zero-mean. If the Gaussian distribution has mean $\mu$, simply adding $\mu$ to each of these points would yield the desired mean and covariance [42]. The symmetric nature of the sigma points set guarantee its first three moments to be the same as the original Gaussian distribution. Also, the odd moments are all zero due to this symmetry. For any choice of the scalar $\lambda$, the first three moments remain unchanged, however, it is a useful parameter for exploiting knowledge about higher moments. For the Gaussian distribution, choosing $\lambda$ such that $n + \lambda = 3$ minimizes the MSE up to fourth order [42]. One efficient method to compute the matrix square-root is the Cholesky decomposition. These sigma points are then propagated with

$$\boldsymbol{\chi}_{k+1}(i) = \mathbf{f}[\boldsymbol{\chi}_k(i), k] \tag{4.47}$$

The predicted mean is given by

$$\hat{\mathbf{x}}_{k+1}^{-} = \frac{1}{n+\lambda}\left\{\lambda\,\boldsymbol{\chi}_{k+1}(0) + \frac{1}{2}\sum_{i=1}^{2n}\boldsymbol{\chi}_{k+1}(i)\right\} \tag{4.48}$$

The predicted error covariance is given by

$$\begin{aligned} P_{k+1}^{-} = \frac{1}{n+\lambda}\Big\{&\lambda\,[\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^{-}]\,[\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^{-}]^T \\ &+\frac{1}{2}\sum_{i=1}^{2n}[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^{-}]\,[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^{-}]^T\Big\} \end{aligned} \tag{4.49}$$

The mean predicted observation is given by

$$\hat{\mathbf{y}}_{k+1}^{-} = \frac{1}{n+\lambda}\left\{\lambda\,\boldsymbol{\gamma}_{k+1}(0) + \frac{1}{2}\sum_{i=1}^{2n}\boldsymbol{\gamma}_{k+1}(i)\right\} \tag{4.50}$$

where

$$\boldsymbol{\gamma}_{k+1}(i) = \mathbf{h}[\boldsymbol{\chi}_{k+1}(i), k] \tag{4.51}$$

The output covariance is given by

$$\begin{aligned} P_{k+1}^{yy} = \frac{1}{n+\lambda}\Bigg\{&\lambda\,[\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^{-}]\,[\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^{-}]^T \\ &+\frac{1}{2}\sum_{i=1}^{2n}[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^{-}]\,[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^{-}]^T\Bigg\} \end{aligned} \tag{4.52}$$

Then, the innovation covariance is simply given by

$$P_{k+1}^{vv} = P_{k+1}^{yy} + R_{k+1} \tag{4.53}$$

Finally, the cross-correlation matrix is determined using

$$\begin{aligned} P_{k+1}^{xy} = \frac{1}{n+\lambda}\Bigg\{&\lambda\,[\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^{-}]\,[\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^{-}]^T \\ &+\frac{1}{2}\sum_{i=1}^{2n}[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^{-}]\,[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^{-}]^T\Bigg\} \end{aligned} \tag{4.54}$$

Then the Kalman gain is computed with Eq. (4.45), and the state vector can now be updated using Eq. (4.43). A summary of the UF algorithm is presented in Table 4.4.

Notice that the predicted error covariance, Eq. (4.49), has the potential to become non-positive semi-definite when $\lambda$ is negative. This could potential cause difficulty in Eq. (4.46a). If this is the case, numerous approaches can be used to guarantee its positive semi-definiteness. One way is to neglect the $\lambda$ term from Eq. (4.49), Eq. (4.54) and Eq. (4.52) [41], which leads to

$$P_{k+1}^{-} = \frac{1}{2(n+\lambda)}\left\{\sum_{i=1}^{2n}[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^{-}]\,[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^{-}]^T\right\} \tag{4.55}$$

Table 4.4: Unscented Filter

| | |
|---|---|
| Model | $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + G_k \mathbf{w}_k$<br>$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k$ |
| Initialize | $\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$<br>$P_0 = \mathrm{E}\{\tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T\}$ |
| Gain | $K_k = P_k^{xy} (P_k^{\upsilon\upsilon})^{-1}$ |
| Update | $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \boldsymbol{\upsilon}_k$<br>$P_k^+ = P_k^- - K_k P_k^{\upsilon\upsilon} K_k^T$<br>$\boldsymbol{\upsilon}_k \equiv \tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^- = \tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-,\, k)$ |
| Propagation | $\boldsymbol{\sigma}_k \leftarrow 2n \text{ columns from } \pm\sqrt{(n+\lambda)[P_k^+ + Q_k]}$<br>$\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k^+$<br>$\boldsymbol{\chi}_k(i) = \boldsymbol{\sigma}_k(i) + \hat{\mathbf{x}}_k^+$<br>$\boldsymbol{\chi}_{k+1}(i) = \mathbf{f}[\boldsymbol{\chi}_k(i), k]$<br>$\hat{\mathbf{x}}_{k+1}^- = \frac{1}{n+\lambda}\left\{\lambda\, \boldsymbol{\chi}_{k+1}(0) + \frac{1}{2}\sum_{i=1}^{2n} \boldsymbol{\chi}_{k+1}(i)\right\}$<br>$P_{k+1}^- = \frac{1}{2(n+\lambda)}\left\{\sum_{i=1}^{2n}[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]\,[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]^T\right\}$<br>$\hat{\mathbf{y}}_{k+1}^- = \frac{1}{n+\lambda}\left\{\lambda\, \boldsymbol{\gamma}_{k+1}(0) + \frac{1}{2}\sum_{i=1}^{2n} \boldsymbol{\gamma}_{k+1}(i)\right\}$<br>$\boldsymbol{\gamma}_{k+1}(i) = \mathbf{h}[\boldsymbol{\chi}_{k+1}(i), k]$<br>$P_{k+1}^{yy} = \frac{1}{2(n+\lambda)}\left\{\sum_{i=1}^{2n}[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]\,[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T\right\}$<br>$P_{k+1}^{\upsilon\upsilon} = P_{k+1}^{yy} + R_{k+1}$<br>$P_{k+1}^{xy} = \frac{1}{2(n+\lambda)}\left\{\sum_{i=1}^{2n}[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]\,[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T\right\}$ |

$$P_{k+1}^{yy} = \frac{1}{2(n+\lambda)}\left\{\sum_{i=1}^{2n}[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]\,[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-]^T\right\} \tag{4.56}$$

$$P_{k+1}^{xy} = \frac{1}{2(n+\lambda)} \left\{ \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^{-}]\,[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^{-}]^T \right\} \tag{4.57}$$

For other methods to guarantee the semi-positiveness of the error covariance, refer to Refs. [44] and [45].

The UF generally requires more computational power from the $2n+1$ propagations, however, the computational requirement could be comparable to the EKF especially when numerical Jacobians need to be evaluated. Another possible performance improvement of the UF is through parallel processing of the propagation equations. When noises are not additive in the system or measurement model, another method that incorporates these noises into the system states and error covariance is presented in Ref. [44]. This method promises possible higher performance (lower estimated error covariance or better stability), but vastly increase the computational load.

## 4.5 Summary

Various sequential filters are discussed in this chapter: the KF, EKF and UF. The traditional KF applies only for linear systems, however, most of the real world problems are nonlinear in nature. The EKF is the nonlinear extension of the traditional KF. It linearizes the system models so that linear KF equations apply. Instead of linearizing the system models, the UF propagates the Gaussian distribution directly, this results in a lower expected error covariance which leads to higher estimation accuracy. The UF linearizes to higher order than the EKF and offers better treatment of the statistics, hence it is more robust than the EKF when higher nonlinearity is a concern. Also, no Jacobian derivations are needed in the UF formulation, thus the UF is far easier to implement on nonlinear systems.

# Chapter 5

# Applications of Complex-Step Derivative Approximation

## 5.1 Introduction

Modern day applications involving complicated nonlinear systems or model equations often make obtaining gradient information analytically from these equations difficult, if not impossible. The goal of this research is to derive a new second-order Kalman filter for nonlinear systems that does not require one to take analytical Jacobian or Hessian of the nonlinear models. Instead, the first- and second-order information is obtained via the complex-step approximation, which has been shown in Chapter 3 to offer better accuracy than standard finite differences.

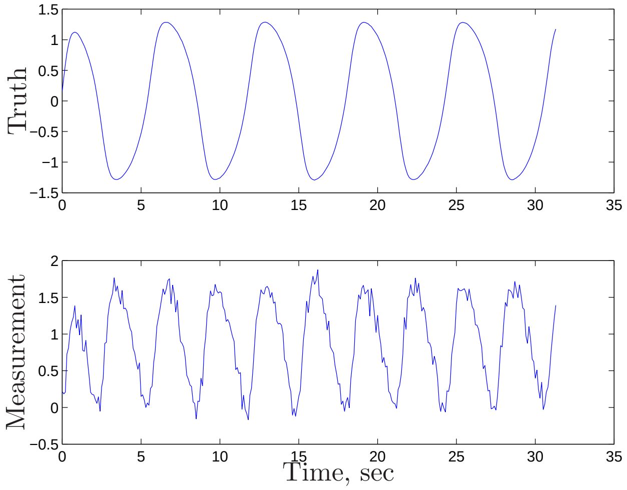
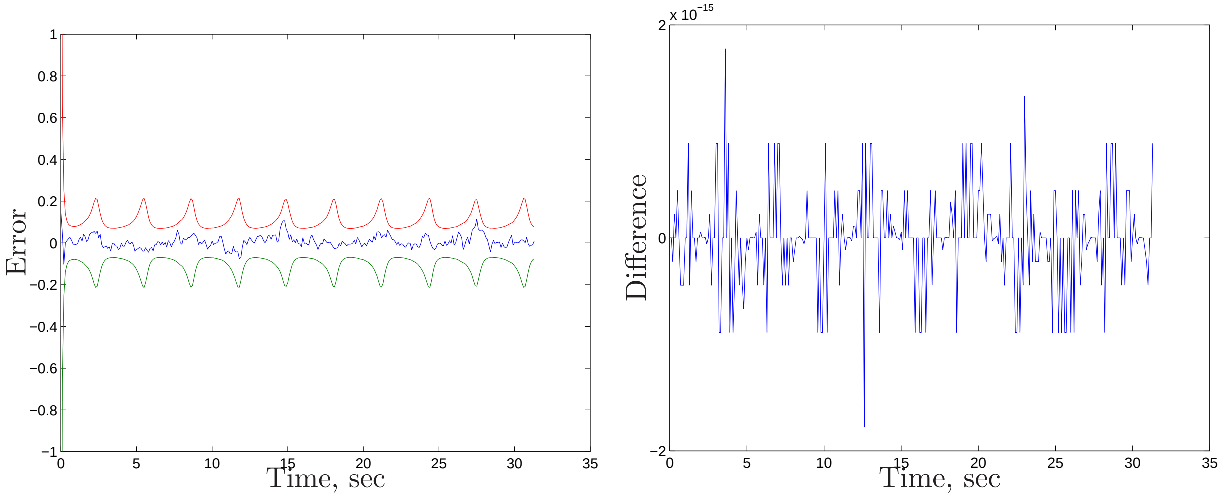


Figure 5.1: Truth and Measurement



(a) State Estimation Error with 3$\sigma$ Bounds

(b) Difference between Analytical and Complex-Step Approximation Solutions

Figure 5.2: Performance Evaluation

## 5.2 First-Order Example

The main purpose of this example is to show the validity of the complex-step derivative in replacement of the analytic or numerical gradient information. For this purpose, an arbitrary system and observation model is used. The system model is given by

$$\dot{x}(t) = -e^{-\gamma x(t)} x^3(t) + 2.3\cos(t) + w(t) \tag{5.1}$$

where $\gamma = 5 \times 10^{-5}$ and $w(t)$ is the process noise with power spectral density $Q = 0.1^2$. The discrete-time observation model

$$y_k = x_k^2 + v_k \tag{5.2}$$

where $v_k$ is the observation noise with variance $R = 0.1^2$. The "truth" is propagated at time-step of 0.01 sec using a fourth-order Runge-Kutta numerical integration. The sampling and filtering time-step is 0.1 sec. The step size for complex-step approximation is set to $h = 1 \times 10^{-11}$.

Figure 5.1 shows the truth and measurement. Figure 5.2 shows the performance of the filter; the estimation error with $3\sigma$ bounds, Fig. 5.2(a), and the difference between analytic derivative and complex-step approximation, Fig. 5.2(b). It is obvious from the scale in Fig. 5.2(b) that the complex-step derivative is very accurate, rivaling that of the analytical solution.

## 5.3 Second-Order Kalman Filter

The extended Kalman filter (EKF) is undeniably the most widely used algorithm for nonlinear state estimation. The EKF is actual a "pseudo-linear" filter since it retains the linear update of the linear Kalman filter, as well as the linear covariance propagation. It only uses the original nonlinear function for the state propagation and definition of the output vector to form the residual [46, 41]. The heart of the EKF lies in a first-order Taylor series expansion of the state and output models. Two approaches can be used for this expansion. The

Table 5.1: Discrete Second-Order Kalman Filter

| | |
|---|---|
| Model | $\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k, \quad \mathbf{w}_k \sim N(\mathbf{0}, Q_k)$<br>$\tilde{\mathbf{y}}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, R_k)$ |
| Initialize | $\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$<br>$P_0 = E\{\tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T\}$ |
| Propagation | $\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}_k(\hat{\mathbf{x}}_k^+) + \frac{1}{2}\sum_{i=1}^n \mathbf{e}_i \mathrm{Tr}\left\{F_{xx,k}^i P_k^+\right\}$<br>$P_{k+1}^- = F_{x,k} P_k^+ F_{x,k}^T + \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \mathbf{e}_i \mathbf{e}_j^T \mathrm{Tr}\left\{F_{xx,k}^i P_k^+ F_{xx,k}^j P_k^+\right\} + Q_k$<br>$\hat{\mathbf{y}}_{k+1} = \mathbf{h}_{k+1}(\hat{\mathbf{x}}_{k+1}^-) + \frac{1}{2}\sum_{i=1}^m \mathbf{e}_i \mathrm{Tr}\left\{H_{xx,k+1}^i P_{k+1}^-\right\}$ |
| Gain | $P_{k+1}^{xy} = H_{x,k+1} P_{k+1}^- H_{x,k+1}^T$<br>$+\frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \mathbf{e}_i \mathbf{e}_j^T \mathrm{Tr}\left\{H_{xx,k+1}^i P_{k+1}^- H_{xx,k+1}^j P_{k+1}^-\right\} + R_{k+1}$<br>$K_{k+1} = P_{k+1}^- H_{x,k+1}^T [P_{k+1}^{xy}]^{-1}$ |
| Update | $\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + K_{k+1}[\tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1}]$<br>$P_{k+1}^+ = P_{k+1}^- - K_{k+1} P_{k+1}^{xy} K_{k+1}^T$ |

first expands a nonlinear function about a nominal (prescribed) trajectory, while the second expands about the current estimate. The advantage of the first approach is the filter gain can be computed offline. However, since the nominal trajectory is usually not as close to the truth as the current estimate in most applications, and with the advent of fast processors in modern-day computers, the second approach is mostly used in practice over the first. Even though the EKF is an approximate approach (at best), its use has found many applications, e.g. in inertial navigation [47], and it does remarkably well.

Even with its wide acceptance, we still must remember that the EKF is merely a linearized approach. Many state estimation designers have fallen into the fallacy that it can work well for any application encountered. But even early-day examples have shown that the first-order expansion approach in the EKF may not produce adequate state estimation results [48]. One obvious extension of the EKF involves a second-order expansion [49], which provides improved performance at the expense of an increased computational burden due to the calculation of second derivatives. Other approaches are shown in Ref. [49] as well, such as the iterated EKF and a statistically linearized filter. Yet another approach that is rapidly gaining attention is the Unscented Kalman filter (UKF), described in Chapter 4. Advantages of the UKF over the EKF are 1) the expected error is lower than the EKF, 2) the new filter can be applied to non-differentiable functions, 3) the new filter avoids the derivation of Jacobian matrices, and 4) the new filter is valid to higher-order expansions than the standard EKF. The UKF works on the premise that with a fixed number of parameters it should be easier to approximate a Gaussian distribution than to approximate an arbitrary nonlinear function. Also, the UKF uses the standard Kalman form in the post-update, but uses a different propagation of the covariance and pre measurement update with no local iterations.

The UKF performance is generally equal to the performance of the second-order Kalman filter (SOKF) since its accuracy is good up to fourth-order moments [44]. The main advantage of the UKF over a SOKF is that partials need not be computed. For simple problems this poses no difficulties, however for large scale problems, such as determining the position of a vehicle from magnetometer measurements [16], these partials are generally analytically intractable. One approach to compute these partials is to use a simple numerical derivative. This approach only works well when these numerical derivatives are nearly as accurate as the analytical derivatives. The new complex-step derivative approximations are used in the SOKF in order to numerically compute these derivatives.

The second-order Kalman filter used here is called the *modified Gaussian second-order filter*. The algorithm is summarized in Table 5.1 for the discrete models, where $\mathbf{e}_i$ represents the $i^{\text{th}}$ basis vector from the identity matrix of appropriate dimension, $F_x$ and $H_x$

are the Jacobian matrices of $\mathbf{f}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$, respectively, and $F^i_{xx}$ and $H^i_{xx}$ are the $i^{\text{th}}$ Hessian matrices of $\mathbf{f}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$, respectively. All Jacobian and Hessian matrices are evaluated at the current state estimates. Notice the extra terms associated with these equations over the standard EKF. These are correction terms to compensate for "biases" that emerge from the nonlinearity in the models. If these biases are insignificant and negligible, the filter reduces to the standard EKF. The SOKF is especially attractive when the process and measurement noise are small compared to the bias correction terms. The only setback of this filter is the requirement of Hessian information, which is often challenging to analytically calculate, if not impossible, for many of today's complicated systems. These calculations are replaced with the complex-step Jacobian and hybrid Hessian approximations.

## 5.4 Classical Falling Object Example

The example presented in this section was first proposed in Ref. [48] and has since become a standard performance evaluation example for various other nonlinear estimators [41, pg. 314] [50]. In Ref. [50] a simpler implementation of this problem is proposed by using a coordinate transformation to reduce the computation load and implementation complexity.

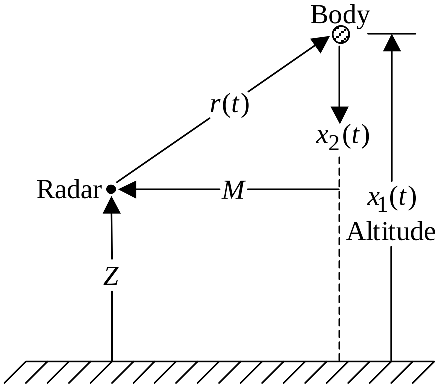


Figure 5.3: Vertically Falling Body Example

However, this coordinate transformation is problem specific and may not apply well to other nonlinear systems. In this section the original formulation is used and applied on a SOKF with the Jacobian and Hessian matrices obtained via both the numerical finite-difference and complex-step approaches. The performance is compared with the EKF, which uses the analytical Jacobian. The equations of motion of the system are given by

$$\dot{x}_1(t) = -x_2(t) \tag{5.3a}$$

$$\dot{x}_2(t) = -e^{-\alpha x_1(t)} x_2^2(t) x_3(t) \tag{5.3b}$$

$$\dot{x}_3(t) = 0 \tag{5.3c}$$

where $x_1(t)$ is the altitude, $x_2(t)$ is the downward velocity, $x_3(t)$ is the constant ballistic coefficient and $\alpha = 5 \times 10^{-5}$ is a constant that's relates air density with altitude. The range

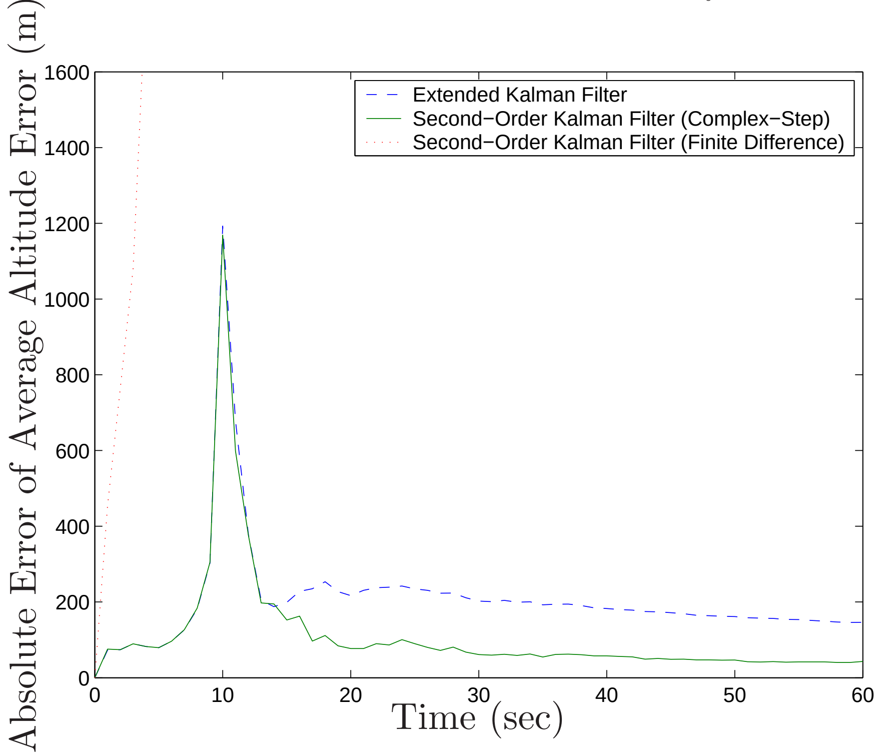


Figure 5.4: Absolute Mean Position Error

observation model is given by

$$\tilde{y}_k = \sqrt{M^2 + (x_{1,k} - Z)^2} + \nu_k \tag{5.4}$$

where $\nu_k$ is the observation noise, and $M$ and $Z$ are constants. These parameters are given by $M = 1 \times 10^5$ and $Z = 1 \times 10^5$. The variance of $\nu_k$ is given by $1 \times 10^4$. Measurements are sampled at 1-second intervals.

The true state and initial estimates are given by

$$x_1(0) = 3 \times 10^5 \quad , \qquad \hat{x}_1(0) = 3 \times 10^5 \tag{5.5}$$

$$x_2(0) = 2 \times 10^4 \quad , \qquad \hat{x}_2(0) = 2 \times 10^4 \tag{5.6}$$

$$x_3(0) = 1 \times 10^{-3} \quad , \qquad \hat{x}_3(0) = 3 \times 10^{-5} \tag{5.7}$$

Clearly, an error is present in the ballistic coefficient value. Physically this corresponds to assuming that the body is “heavy” whereas in reality the body is “light.” The initial covariance for all filters is given by

$$P(0) = \begin{bmatrix} 1 \times 10^6 & 0 & 0 \\ 0 & 4 \times 10^6 & 0 \\ 0 & 0 & 1 \times 10^{-4} \end{bmatrix} \tag{5.8}$$

The goal is to substitute analytical or numerical finite difference with complex-step derivative approximation. Thus it is useful in this example that the analytical Jacobian and Hessian information is available. We use this as a comparison to the ones obtain via complex-step approximation. The analytical Jacobian of the system and measurement model is obtained as

$$Jf = e^{-\alpha\hat{x}_1}\begin{bmatrix} 0 & -e^{\alpha\hat{x}_1} & 0 \\ \alpha\hat{x}_2^2\hat{x}_3 & -2\hat{x}_2\hat{x}_3 & -\hat{x}_2^2 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.9a}$$

$$Jh = \begin{bmatrix} \frac{\hat{x}_1 - Z}{\sqrt{M^2 + (\hat{x}_1 - Z)^2}} & 0 & 0 \end{bmatrix} \tag{5.9b}$$

The analytical Hessian of the system and measurement model

$$Hf_{\hat{x}_1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.10a}$$

$$Hf_{\hat{x}_2} = e^{-\alpha\hat{x}_1}\begin{bmatrix} -\alpha^2\hat{x}_2^2\hat{x}_3 & 2\alpha\hat{x}_2\hat{x}_3 & \alpha\hat{x}_2^2 \\ 2\alpha\hat{x}_2\hat{x}_3 & -2\hat{x}_3 & -2\hat{x}_2 \\ \alpha\hat{x}_2^2 & -2\hat{x}_2 & 0 \end{bmatrix} \tag{5.10b}$$

$$Hf_{\hat{x}_3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.10c}$$

$$Hh_{\hat{x}_1} = \begin{bmatrix} M^2[M^2 + (\hat{x}_1 - Z)^2]^{-3/2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.10d}$$

Figure 5.4 shows the average position error, using a Monte-Carlo simulation of 10 runs, of the EKF with analytical derivatives, the SOKF with complex-step derivatives, and SOKF with finite-difference derivatives. The step size, $h$, for all finite-difference and complex-step operations is set to $1 \times 10^{-4}$. For all Monte-Carlo simulations the SOKF with finite-difference derivatives diverges using this step-size. For other step sizes the SOKF with finite-difference derivatives does converge, but only within a narrow region of step sizes (in this case only between 1.0 and $1 \times 10^{-3}$). The performance is never better than using the complex-step approach. From Figure 5.4 there is little difference among the EKF and SOKF with complex-step derivatives approaches before the first 12 seconds when the altitude is high. When the drag becomes significant at about 9 seconds, then these two filters exhibit large errors in position estimation. This coincides with the time when the falling body is on the same level as the radar, so the system becomes nearly unobservable. Eventually, the two filters demonstrate convergence with the EKF being the slowest. This is due to the deficiency of the EKF to capture the high nonlinearities present in the system. The SOKF with complex-step derivatives performs clearly better than the EKF. It should be noted that the SOKF with the complex-step derivative approximation performs equally well as using the analytical Jacobian and Hessian matrices for all values of $h$ discussed here.

## 5.5 Conclusion

The usefulness of complex-step approximation in obtaining derivative information of an arbitrary nonlinear function has been shown. The accuracy of this approximation is very close to the solution obtained from analytical methods, so close that the difference between them is often within machine precision. It is reasonable for us to believe that application to second-order Kalman filter would greatly reduce the filter implementation time by eliminating the needs to painfully derive the analytical Jacobian and Hessian. The performance of this new filter is expected to be of the same order or better than many other nonlinear filters. Also, unlike some variants of the linear regression Kalman filter (LRKF), such as the UKF or sigma point Kalman filter (SPKF), the new filter does not require the “tuning” of filter parameters, other than the process noise covariance.

# Chapter 6

# Square-Root Filtering with Complex-Step Derivative Approximation

## 6.1 Introduction

The complex-step derivative approximations derived in Chapter 3 were applied to finding the Jacobian and Hessian information of a function and applied to first- and second-order extended Kalman filters in Chapter 5. In this chapter a second-order filtering scheme running at the "square root" level is explored. A new class of filters that do not require linearizations of the state (dynamics) propagation and measurement model equations has become popular in recent years [51, 42, 35]. Reference [52] further classifies these filters collectively as "Linear Regression Kalman filters," since they all fall into the category of filters that linearize the statistics of the nonlinear models instead of the nonlinear models themselves. While finding the Jacobian and Hessian matrices are not needed, these filters attempt to capture the statistics of transformations via finite-difference equations. However, these filters still fall into the general category of the Kalman filter that updates the propagated states linearly as a function of the difference between estimated measurements and actual measurements.

Square root filtering is an extension to traditional filtering that propagates and updates the covariance matrix at the square root level. However, the intricate formulation often deters filter designers in implementing it. Square root filtering offers slightly better numerical accuracy and robustness especially when filtering states are close to the computer's numerical precision. Square root filtering also guarantees semi-positiveness of the covariance matrix and sometimes offers slight advantages in computational load.

The second-order Divided Difference (DD2) filter from Ref. [35] is a more generalized version of the "Unscented filter" [42, 51] that offers the same mean estimation accuracy. However, the covariance estimation is more accurate in DD2 filter from more accurate treatment of the Gaussian statistics. Also, the DD2 filter operates at the square root level. In this chapter, the first and second-order finite difference used in derivation of the DD2 filter will be replaced with complex-step derivative approximations and thus generalizes it to the complex domain. For these similarities, this chapter would follow the development of Refs. [42] and [35] closely.

## 6.2 Power Series

### 6.2.1 Scalar Case

The step size for difference and average operator are augmented with a unity magnitude complex number:

$$\delta f(x) = f\left(x + \frac{1}{2}e^{\theta i}h\right) - f\left(x - \frac{1}{2}e^{\theta i}h\right) \tag{6.1a}$$

$$\mu f(x) = \frac{1}{2}\left[f\left(x + \frac{1}{2}e^{\theta i}h\right) + f\left(x - \frac{1}{2}e^{\theta i}h\right)\right] \tag{6.1b}$$

where $\theta$ is the associated “angle” of departure from the positive real axis. Let’s take a moment and revisit the Taylor series nominally at $\bar{x}$:

$$f(x) = f(\bar{x}) + f'(\bar{x})(x-\bar{x}) + \frac{1}{2!}f''(\bar{x})(x-\bar{x})^2 + \frac{1}{3!}f^{(3)}(\bar{x})(x-\bar{x})^3 \cdots \tag{6.2}$$

and a Taylor series with step size of $+e^{\theta i}h$ and $-e^{\theta i}h$:

$$\begin{aligned} f(x) = f(\bar{x} + e^{\theta i}h) = f(\bar{x}) + f'(\bar{x})(e^{\theta i}h) + \frac{1}{2!}f''(\bar{x})(e^{\theta i}h)^2 \\ + \frac{1}{3!}f^{(3)}(\bar{x})(e^{\theta i}h)^3 + \frac{1}{4!}f^{(4)}(\bar{x})(e^{\theta i}h)^4 + \cdots \end{aligned} \tag{6.3a}$$

$$\begin{aligned} f(x) = f(\bar{x} - e^{\theta i}h) = f(\bar{x}) - f'(\bar{x})(e^{\theta i}h) + \frac{1}{2!}f''(\bar{x})(e^{\theta i}h)^2 \\ - \frac{1}{3!}f^{(3)}(\bar{x})(e^{\theta i}h)^3 + \frac{1}{4!}f^{(4)}(\bar{x})(e^{\theta i}h)^4 + \cdots \end{aligned} \tag{6.3b}$$

A Taylor series expansion evaluates the derivative information of an analytical function at a precise point and assumes these derivatives to remain valid around the vicinity of this point. For highly nonlinear functions, the derivative calculations deviate quickly from the nominal point. Therefore, derivative information with uniform performance across a region of interest should be used. This is achieved with derivatives derived by using interpolation. Another advantage is that interpolations generally require only function evaluations and not analytical derivations. In this dissertation, the Stirling interpolation [53] is chosen to obtain the derivation information. With the Stirling interpolation, an approximation of a nonlinear function can be expressed as

$$\begin{aligned} f(x) &= f(\bar{x} + e^{\theta i} ph) \\ &= f(\bar{x}) + p\mu\delta f(\bar{x}) + \frac{1}{2!}p^2\delta^2 f(\bar{x}) + \binom{p+1}{3}\mu\delta^3 f(\bar{x}) + \frac{1}{4!}p^2(p^2-1)\delta^4 \\ &\quad + \binom{p+2}{5}\mu\delta^5 f(\bar{x}) + \cdots \end{aligned} \tag{6.4}$$

as described in Appendix B.2. Generally, $-1 < p < 1$ as for interpolation within the region of interest, between $-h$ and $+h$. Concentrating only on the first two derivative expansions gives

$$f(x) \approx f(\bar{x}) + f'_{CSDA}(\bar{x})(x - \bar{x}) + \frac{1}{2!}f''_{CSDA}(\bar{x})(x - \bar{x})^2 \tag{6.5}$$

where $f'_{CSDA}(\bar{x})$ and $f''_{CSDA}(\bar{x})$ are the first and second complex-step derivative approximations from Eqs. (3.23) without the truncation error, which are repeated here for convenience:

$$f'_{CSDA}(x) = \frac{f(x + e^{i\theta}h) - f(x - e^{i\theta}h)}{2e^{i\theta}h} \tag{6.6a}$$

$$f''_{CSDA}(x) = \frac{f(x + e^{i\theta}h) - 2f(x) + f(x - e^{i\theta}h)}{(e^{i\theta}h)^2} \tag{6.6b}$$

Equation (6.5) is basically a Taylor series with derivatives replaced with CSDAs. Assuming

$f$ is analytic, substituting Eqs. (6.3) into Eqs. (6.6) for Eq. (6.5) yields

$$\begin{aligned} f(\bar{x}) + f'_{CSDA}(\bar{x})(x-\bar{x}) + \frac{1}{2!}f''_{CSDA}(\bar{x})(x-\bar{x})^2 \\ = f(\bar{x}) + f'(\bar{x})(x-\bar{x}) + \frac{1}{2!}f''(\bar{x})(x-\bar{x})^2 \\ + \left[\frac{1}{3!}f^{(3)}(\bar{x})(e^{\theta i}h)^2 + \frac{1}{5!}f^{(5)}(\bar{x})(e^{\theta i}h)^4 + \cdots\right](x-\bar{x}) \\ + \left[\frac{1}{4!}f^{(4)}(\bar{x})(e^{\theta i}h)^2 + \frac{1}{6!}f^{(6)}(\bar{x})(e^{\theta i}h)^4 + \cdots\right](x-\bar{x})^2 \end{aligned} \tag{6.7}$$

The first three terms on the right-hand-side of the equation are the first three terms of the Taylor series. The choice of $h$ has an influence only on the "remainder" terms and the optimal choice of $h$ is explored in Ref. [35] to be $h^2 = 3$ for a Gaussian distribution. Another variable to be manipulated to our advantage is the $\theta$ value which is related to the power of an imaginary number (see Appendix B).

### 6.2.2 Vector Case

The scalar analysis is now extended to the multi-variable case, with $\mathbf{x} \in \mathbb{R}^n$. The two Stirling operators in vector forms are simply

$$\delta_p \mathbf{f}(\mathbf{x}) = \mathbf{f}\left(\mathbf{x} + \frac{1}{2}e^{\theta i}h\boldsymbol{\varepsilon}_p\right) - \mathbf{f}\left(\mathbf{x} - \frac{1}{2}e^{\theta i}h\boldsymbol{\varepsilon}_p\right), \tag{6.8a}$$

$$\mu_p \mathbf{f}(\mathbf{x}) = \frac{1}{2}\left[\mathbf{f}\left(\mathbf{x} + \frac{1}{2}e^{\theta i}h\boldsymbol{\varepsilon}_p\right) + \mathbf{f}\left(\mathbf{x} - \frac{1}{2}e^{\theta i}h\boldsymbol{\varepsilon}_p\right)\right], \tag{6.8b}$$

where the subscript $p$ emphasizes it is the $p^{\text{th}}$ "partial" operator with $\boldsymbol{\varepsilon}_p$ being the $p^{\text{th}}$ column of a $p \times p$ identity matrix (the $p^{\text{th}}$ basis vector).

Let $\mathbf{y} = \mathbf{f}(\mathbf{x})$ denote a nonlinear vector transformation. Its Taylor series can be expressed as

$$\begin{aligned}\mathbf{y} = \mathbf{f}(\bar{\mathbf{x}} + \Delta\mathbf{x}) &= \sum_{p=0}^{\infty} \frac{1}{p!} D_{\Delta x}^{p} \mathbf{f} \\ &= \mathbf{f}(\bar{\mathbf{x}}) + D_{\Delta x}\mathbf{f} + \frac{1}{2!} D_{\delta x}^{2} \mathbf{f} + \frac{1}{3!} D_{\delta x}^{3} \mathbf{f} + \cdots \end{aligned} \tag{6.9}$$

where

$$D_{\Delta x}^{p}\mathbf{f} = \left[ \Delta x_1 \frac{\partial}{\partial x_1} + \Delta x_2 \frac{\partial}{\partial x_2} + \cdots + \Delta x_n \frac{\partial}{\partial x_n} \right]^{p} \mathbf{f}(\mathbf{x}) \bigg|_{\mathbf{x}=\bar{\mathbf{x}}} \tag{6.10}$$

The first two operators are simply

$$D_{\Delta x}\mathbf{f} = \left[ \sum_{p=1}^{n} \Delta x_p \frac{\partial}{\partial x_p} \right] \mathbf{f}(\mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}} \tag{6.11a}$$

$$D_{\Delta x}^{2}\mathbf{f} = \left[ \sum_{p=1}^{n} \sum_{q=1}^{n} \Delta x_p \Delta x_q \frac{\partial^2}{\partial x_p \partial x_q} \right] \mathbf{f}(\mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}} \tag{6.11b}$$

or expressed with Stirling interpolation or finite-difference as

$$\tilde{D}_{\Delta x}\mathbf{f} = \frac{1}{e^{i\theta}h} \left[ \Delta x_p \mu_p \delta_p \right] \mathbf{f}(\bar{\mathbf{x}}) \tag{6.12a}$$

$$\tilde{D}_{\Delta x}^{2}\mathbf{f} = \frac{1}{(e^{i\theta}h)^2} \left[ \sum_{p=1}^{n} (\Delta x_p)^2 \delta_p^2 + \sum_{p=1}^{n} \sum_{q=1, q\neq p}^{n} \Delta x_p \Delta x_q (\mu_p \delta_p)(\mu_q \delta_q) \right] \mathbf{f}(\bar{\mathbf{x}}) \tag{6.12b}$$

Again, restricting the series to second order only gives

$$\mathbf{y} \approx \mathbf{f}(\bar{\mathbf{x}}) + \tilde{D}_{\Delta x}\mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{2!} \tilde{D}_{\Delta x}^{2}\mathbf{f}(\bar{\mathbf{x}}) \tag{6.13}$$

Equation (6.13) is just one of the many multi-variable extensions of the Taylor series.

## 6.3 Approximation of Mean and Covariance

### 6.3.1 Truth Quantities

The estimated mean and error covariance will be compared to the "true" mean and error covariance for performance comparison. The true mean and error covariance are simply

$$\bar{\mathbf{x}} = \mathrm{E}\{\mathbf{x}\} \tag{6.14a}$$

$$P_{xx} = \mathrm{E}\{[\mathbf{x} - \bar{\mathbf{x}}][\mathbf{x} - \bar{\mathbf{x}}]^T\} \tag{6.14b}$$

Additionally, the true mean after the nonlinear transformation, and its error covariance and the cross covariance need to be determined:

$$\bar{\mathbf{y}}_T = \mathrm{E}\{\mathbf{f}(\mathbf{x})\} \tag{6.15a}$$

$$P_{yy,T} = \mathrm{E}\{[\mathbf{f}(\mathbf{x}) - \bar{\mathbf{y}}_T][\mathbf{f}(\mathbf{x}) - \bar{\mathbf{y}}_T]^T\} \tag{6.15b}$$

$$P_{xy,T} = \mathrm{E}\{[\mathbf{x} - \bar{\mathbf{x}}][\mathbf{f}(\mathbf{x}) - \bar{\mathbf{y}}_T]^T\} \tag{6.15c}$$

### 6.3.2 Statistical Decoupling

Equation (6.13) is one of the many multi-variable extensions using an interpolation approximation. Other extensions can be derived by using a linear transform of the original vector:

$$\mathbf{z} = S^{-1}\mathbf{x} \tag{6.16}$$

and the new nonlinear transformation,

$$\tilde{\mathbf{f}}(\mathbf{z}) \equiv \mathbf{f}(S\mathbf{z}) = \mathbf{f}(\mathbf{x}) \tag{6.17}$$

The Taylor series expansion for Eq. (6.17) is simply

$$\mathbf{y} \approx \tilde{\mathbf{f}}(\bar{\mathbf{z}}) + \tilde{D}_{\Delta z}\tilde{\mathbf{f}}(\bar{\mathbf{z}}) + \frac{1}{2!}\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}(\bar{\mathbf{z}}) \tag{6.18}$$

As before, $\tilde{\mathbf{f}}$ will be used periodically in placement of $\tilde{\mathbf{f}}(\bar{\mathbf{z}})$ where it deems suitable for better readability. Since Eq. (6.16) is just a linear constant transformation, the Taylor series expansions in Eqs. (6.13) and (6.18) are the same. However, this is not true with interpolation in place of the derivatives. For example, consider the first-order partial part of the Taylor series (refer to Appendix A for higher order Stirling operators),

$$\begin{aligned}
\tilde{D}_{\Delta x}\mathbf{f}(\bar{\mathbf{x}}) &= \frac{1}{e^{i\theta}h}\left[\sum_{p=1}^{n}\Delta x_p\mu_p\delta_p\right]\mathbf{f}(\bar{\mathbf{x}}) \\
&= \frac{1}{2e^{i\theta}h}\sum_{p=1}^{n}\Delta x_p\left[\mathbf{f}(\bar{\mathbf{x}} + e^{i\theta}h\boldsymbol{\varepsilon}_p) + \mathbf{f}(\bar{\mathbf{x}} - e^{i\theta}h\boldsymbol{\varepsilon}_p)\right]
\end{aligned} \tag{6.19a}$$

$$\begin{aligned}
\tilde{D}_{\Delta z}\tilde{\mathbf{f}}(\bar{\mathbf{z}}) &= \frac{1}{e^{i\theta}h}\left[\sum_{p=1}^{n}\Delta z_p\mu_p\delta_p\right]\tilde{\mathbf{f}}(\bar{\mathbf{z}}) \\
&= \frac{1}{2e^{i\theta}h}\sum_{p=1}^{n}\Delta z_p\left[\tilde{\mathbf{f}}(\bar{\mathbf{z}} + e^{i\theta}h\boldsymbol{\varepsilon}_p) + \tilde{\mathbf{f}}(\bar{\mathbf{z}} - e^{i\theta}h\boldsymbol{\varepsilon}_p)\right] \\
&= \frac{1}{2e^{i\theta}h}\sum_{p=1}^{n}\Delta z_p\Big[\mathbf{f}\left(S[\bar{\mathbf{z}} + e^{i\theta}h\boldsymbol{\varepsilon}_p]\right) + \mathbf{f}\left(S[\bar{\mathbf{z}} - e^{i\theta}h\boldsymbol{\varepsilon}_p]\right)\Big] \\
&= \frac{1}{2e^{i\theta}h}\sum_{p=1}^{n}S^{-1}\Delta x_p\Big[\mathbf{f}\left(\bar{\mathbf{x}} + e^{i\theta}h\mathbf{s}_p\right) + \mathbf{f}\left(\bar{\mathbf{x}} - e^{i\theta}h\mathbf{s}_p\right)\Big]
\end{aligned} \tag{6.19b}$$

where $\mathbf{s}_p$ is the $p^{\text{th}}$ column of $S$. It is apparent from Eqs. (6.19) that Eq. (6.13) will be different than Eq. (6.18).

Any square symmetric positive-definite matrix can be decomposed into two triangular matrices, each equals the transpose of the other, $P = SS^T$. This is called the Cholesky decomposition and the decomposed matrix is referred as the Cholesky factor, $S$. Many other decomposition schemes exist, but the Cholesky decomposition is proved to be a computationally more efficient. One particularly useful transformation of $\mathbf{x}$ is by using the Cholesky factor of the state error-covariance matrix ($P_{xx} = S_x S_x^T$) to *stochastically decouple* $\mathbf{x}$ and $\mathbf{z}$,

$$\mathbf{z} = S_x^{-1}\mathbf{x} \tag{6.20}$$

so that elements of $\mathbf{z}$ are now mutually independent from each other and has unity variance with itself,

$$\mathrm{E}\left\{[\mathbf{z}-\bar{\mathbf{z}}][\mathbf{z}-\bar{\mathbf{z}}]^T\right\} = I \quad , \qquad \bar{\mathbf{z}} = \mathrm{E}\left\{\mathbf{z}\right\} \tag{6.21}$$

or

$$\Delta\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I) \tag{6.22}$$

with $\Delta\mathbf{z} = \mathbf{z} - \bar{\mathbf{z}}$; notice that the symmetrically distributed $\mathbf{z}$ translates their zero mean distribution. The advantage of this decoupling will be made clear in the subsequent analysis with $\mathbf{z}$ instead of $\mathbf{x}$ directly. Conversion back to $\mathbf{x}$ upon completion of analysis is trivial. Also, during these analysis, $\tilde{\mathbf{f}}(\mathbf{z})$ is defined $\forall\mathbf{z} \in \mathbb{R}^n$.

## 6.4 Second-Order Approximation

The second-order polynomial approximation of the true nonlinear transformation is simply

$$\begin{aligned}\mathbf{y} &\approx \tilde{f}(\bar{\mathbf{z}}) + \tilde{D}_{\Delta z}\mathbf{f} + \frac{1}{2}\tilde{D}^2_{\Delta z}\mathbf{f} \\ &= \tilde{\mathbf{f}}(\bar{\mathbf{z}}) + \frac{1}{e^{i\theta}h}\left(\sum_{p=1}^{n} \Delta z_p \mu_p \delta_p\right)\tilde{\mathbf{f}}(\bar{\mathbf{z}}) \\ &\quad + \frac{1}{2(e^{i\theta}h)^2}\left(\sum_{p=1}^{n}(\Delta z_p)^2\delta_p^2 + \sum_{p=1}^{n}\sum_{q=1,q\neq p}^{n}\Delta z_p \Delta z_q(\mu_p\delta_p)(\mu_q\delta_q)\right)\tilde{\mathbf{f}}(\bar{\mathbf{z}}) \qquad (6.23)\end{aligned}$$

and its estimated quantity, $\bar{\mathbf{y}} = \mathrm{E}\left\{\mathbf{y}\right\}$,

$$\begin{aligned}\bar{\mathbf{y}} &= E\left\{\tilde{\mathbf{f}}(\bar{\mathbf{z}}) + \frac{1}{2(e^{i\theta}h)^2}\left(\sum_{p=1}^{n}(\Delta z_p)^2\delta_p^2\right)\tilde{\mathbf{f}}(\bar{\mathbf{z}})\right\} \\ &= \tilde{\mathbf{f}}(b\bar{f}z) + \frac{\sigma_2}{2(e^{i\theta}h)^2}\sum_{p=1}^{n}\delta_p^2\tilde{\mathbf{f}}(\bar{\mathbf{z}}) \\ &= \tilde{\mathbf{f}}(\bar{\mathbf{z}}) + \frac{1}{2(e^{i\theta}h)^2}\sum_{p=1}^{n}\left[\tilde{\mathbf{f}}(\bar{\mathbf{z}} + e^{i\theta}h\mathbf{e}_p) + \tilde{\mathbf{f}}(\bar{\mathbf{z}} - e^{i\theta}h\mathbf{e}_p)\right] - \frac{n}{(e^{i\theta}h)^2}\tilde{\mathbf{f}}(\bar{\mathbf{z}}) \\ &= \frac{(e^{i\theta}h)^2 - n}{(e^{i\theta}h)^2}\tilde{\mathbf{f}}(\bar{\mathbf{z}}) + \frac{1}{2(e^{i\theta}h)^2}\sum_{p=1}^{n}\left[\tilde{\mathbf{f}}(\bar{\mathbf{z}} + e^{i\theta}h\mathbf{e}_p) + \tilde{\mathbf{f}}(\bar{\mathbf{z}} - e^{i\theta}h\mathbf{e}_p)\right] \\ &= \frac{(e^{i\theta}h)^2 - n}{(e^{i\theta}h)^2}\mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{2(e^{i\theta}h)^2}\sum_{p=1}^{n}\left[\mathbf{f}(\bar{\mathbf{x}} + e^{i\theta}h\mathbf{s}_{x,p}) + \mathbf{f}(\bar{\mathbf{x}} - e^{i\theta}h\mathbf{s}_{x,p})\right] \qquad (6.24)\end{aligned}$$

From Eq. (6.23), notice that $\tilde{\mathbf{f}}(\bar{\mathbf{z}})$ is a deterministic term, thus instead of $\mathbf{y}$, $\mathbf{y} - \tilde{\mathbf{f}}(\bar{\mathbf{z}})$ could be used in derivation of the covariance for $\mathbf{y}$ as this would simplify the intermediate analysis,

$$\begin{aligned}P_{yy,T} &= \mathrm{E}\left\{[\mathbf{y} - \mathrm{E}\left\{\mathbf{y}\right\}][\mathbf{y} - \mathrm{E}\left\{\mathbf{y}\right\}]^T\right\} \\ &= \mathrm{E}\left\{[\mathbf{y}][\mathbf{y}]^T\right\} - \mathrm{E}\left\{\mathbf{y}\right\}\mathrm{E}\left\{\mathbf{y}\right\}^T \\ &= \mathrm{E}\left\{[\mathbf{y} - \tilde{\mathbf{f}}(\bar{\mathbf{z}})][\mathbf{y} - \tilde{\mathbf{f}}(\bar{\mathbf{z}})]^T\right\} - E\left\{\mathbf{y} - \tilde{\mathbf{f}}(\bar{\mathbf{z}})\right\} E\left\{\mathbf{y} - \tilde{\mathbf{f}}(\bar{\mathbf{z}})\right\}^T \qquad (6.25)\end{aligned}$$

This leaves the estimate covariance as,

$$\begin{aligned}
P_{yy} &= \mathrm{E}\left\{\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}} + \frac{1}{2}\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right]\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}} + \frac{1}{2}\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right]^T\right\} \\
&\quad - \mathrm{E}\left\{\tilde{D}_{\Delta z}\tilde{\mathbf{f}} + \frac{1}{2}\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right\}\mathrm{E}\left\{\tilde{D}_{\Delta z}\tilde{\mathbf{f}} + \frac{1}{2}\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right\}^T \\
&= \overset{\textcircled{1}}{\mathrm{E}\left\{\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}}\right]\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}}\right]^T\right\}} + \frac{1}{4}\overset{\textcircled{2}}{\mathrm{E}\left\{\left[\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right]\left[\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right]^T\right\}} - \frac{1}{4}\overset{\textcircled{3}}{\mathrm{E}\left\{\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right\}\mathrm{E}\left\{\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right\}^T} \qquad (6.26)
\end{aligned}$$

Note that all odd moments in Eq. (6.26) evaluate to zero due to the symmetric distribution of elements in $\mathbf{z}$ and they are uncorrelated with each other. Given the length of the individual analysis, each of the term ①, ② and ③ will be evaluated separately. The $p^{\text{th}}$ moment of and element of $\mathbf{z}$ is denoted as $\sigma_p$. Also, as from Eq. (6.21), the second moment of each $\mathbf{z}$ element is unity:

① $\mathrm{E}\left\{\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}}\right]\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}}\right]^T\right\}$

$$\begin{aligned}
\mathrm{E}\left\{\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}}\right]\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}}\right]^T\right\} &= \frac{1}{(e^{i\theta}h)^2}\mathrm{E}\left\{\left[\sum_{p=1}^{n}\Delta z_p\mu_p\delta_p\tilde{\mathbf{f}}(\bar{\mathbf{z}})\right]\left[\sum_{p=1}^{n}\Delta z_p\mu_p\delta_p\tilde{\mathbf{f}}(\bar{\mathbf{z}})\right]^T\right\} \\
&= \frac{\sigma_2}{(e^{i\theta}h)^2}\sum_{p=1}^{n}\left[\mu_p\delta_p\tilde{\mathbf{f}}(\bar{\mathbf{z}})\right]\left[\mu_p\delta_p\tilde{\mathbf{f}}(\bar{\mathbf{z}})\right]^T \\
&= \frac{1}{4(e^{i\theta}h)^2}\sum_{p=1}^{n}\left[\tilde{\mathbf{f}}(\bar{\mathbf{z}} + e^{i\theta}h\boldsymbol{\varepsilon}_p)\right]\left[\tilde{\mathbf{f}}(\bar{\mathbf{z}} + e^{i\theta}h\boldsymbol{\varepsilon}_p)\right]^T \\
&= \frac{1}{4(e^{i\theta}h)^2}\sum_{p=1}^{n}\left[\mathbf{f}(\bar{\mathbf{x}} + e^{i\theta}h\mathbf{s}_{x,p})\right]\left[\mathbf{f}(\bar{\mathbf{x}} + e^{i\theta}h\mathbf{s}_{x,p})\right]^T \qquad (6.27)
\end{aligned}$$

where $\mathbf{s}_{x,p}$ is the $p^{\text{th}}$ column of the square Cholesky factor from Eq. (6.20).

② $\underline{\mathrm{E}\left\{\left[\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right]\left[\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right]^T\right\}}$ consists of three kinds of terms, $\forall p$, $\forall q$, $p \neq q$,

$$\mathrm{E}\left\{\left[(\Delta z_p)^2\delta_p^2\tilde{\mathbf{f}}\right]\left[(\Delta z_p)^2\delta_p^2\tilde{\mathbf{f}}\right]^T\right\} = \left[\delta_p^2\tilde{\mathbf{f}}\right]\left[\delta_p^2\tilde{\mathbf{f}}\right]^T\sigma_4 \tag{6.28a}$$

$$\mathrm{E}\left\{\left[(\Delta z_p)^2\delta_p^2\tilde{\mathbf{f}}\right]\left[(\Delta z_q)^2\delta_q^2\tilde{\mathbf{f}}\right]^T\right\} = \left[\delta_p^2\tilde{\mathbf{f}}\right]\left[\delta_q^2\tilde{\mathbf{f}}\right]^T\sigma_2^2 \tag{6.28b}$$

$$\mathrm{E}\left\{\left[\Delta z_p\Delta z_q\mu_p\delta_q\mu_p\delta_q\tilde{\mathbf{f}}\right]\left[\Delta z_p\Delta z_q\mu_p\delta_q\mu_p\delta_q\tilde{\mathbf{f}}\right]^T\right\} = \left[\mu_p\delta_p\mu_q\delta_q\tilde{\mathbf{f}}\right]\left[\mu_p\delta_p\mu_q\delta_q\tilde{\mathbf{f}}\right]^T\sigma_2^2 \tag{6.28c}$$

③ $\underline{\mathrm{E}\left\{\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right\}\mathrm{E}\left\{\tilde{D}^2_{\Delta z}\tilde{\mathbf{f}}\right\}^T}$ consists of two kinds of terms, $\forall p$, $\forall q$, $p \neq q$,

$$\mathrm{E}\left\{(\Delta z_p)^2\delta_p\tilde{\mathbf{f}}\right\}\mathrm{E}\left\{(\Delta z_p)^2\delta_p\tilde{\mathbf{f}}\right\}^T = \left[\delta_p^2\tilde{\mathbf{f}}\right]\left[\delta_p^2\tilde{\mathbf{f}}\right]^T\sigma_2^2 \tag{6.29a}$$

$$\mathrm{E}\left\{(\Delta z_p)^2\delta_p\tilde{\mathbf{f}}\right\}\mathrm{E}\left\{(\Delta z_q)^2\delta_q\tilde{\mathbf{f}}\right\}^T = \left[\delta_p^2\tilde{\mathbf{f}}\right]\left[\delta_q^2\tilde{\mathbf{f}}\right]^T\sigma_2^2 \tag{6.29b}$$

Terms in Eqs. (6.28b) and (6.29b) cancel each other. Terms in Eq. (6.29b) will be discarded from analysis for the reason explained in Ref. [54]. Basically they do not constitute to better filter accuracy, while the computationally expensive to calculate. The Stirling operators portion from Eqs. (6.28a) and (6.29a) are expanded as

$$\begin{aligned}
\left[\delta_p^2\tilde{\mathbf{f}}(\bar{\mathbf{z}})\right]\left[\delta_p^2\tilde{\mathbf{f}}(\bar{\mathbf{z}})\right]^T &= \left[\tilde{\mathbf{f}}(\bar{\mathbf{z}} + e^{i\theta}h\boldsymbol{\varepsilon}_p) - \tilde{\mathbf{f}}(\bar{\mathbf{z}} - e^{i\theta}h\boldsymbol{\varepsilon}_p)\right] \times \\
&\qquad\qquad \left[\tilde{\mathbf{f}}(\bar{\mathbf{z}} + e^{i\theta}h\boldsymbol{\varepsilon}_p) - \tilde{\mathbf{f}}(\bar{\mathbf{z}} - e^{i\theta}h\boldsymbol{\varepsilon}_p)\right]^T \\
&= \left[\mathbf{f}(\bar{\mathbf{x}} + e^{i\theta}h\mathbf{s}_{x,p}) - \mathbf{f}(\bar{\mathbf{x}} - e^{i\theta}h\mathbf{s}_{x,p})\right] \times \\
&\qquad\qquad \left[\mathbf{f}(\bar{\mathbf{x}} + e^{i\theta}h\mathbf{s}_{x,p}) - \mathbf{f}(\bar{\mathbf{x}} - e^{i\theta}h\mathbf{s}_{x,p})\right]^T
\end{aligned} \tag{6.30}$$

Again, $\sigma_2 = 1$ from the way $\mathbf{z}$ is generated. The terms $\sigma_4 = h^2$, and $h^2 = 3$ if the distribution is Gaussian, for the analysis refer to §3.3 of Ref. [54]. Finally $P_{yy}$ becomes

$$\begin{aligned}
P_{yy} = &\frac{1}{4(e^{i\theta}h)^2}\sum_{p=1}^{n}\left[\mathbf{f}(\bar{\mathbf{x}}+e^{i\theta}h\mathbf{s}_{x,p})-\mathbf{f}(\bar{\mathbf{x}}-e^{i\theta}h\mathbf{s}_{x,p})\right]\left[\mathbf{f}(\bar{\mathbf{x}}+e^{i\theta}h\mathbf{s}_{x,p})-\mathbf{f}(\bar{\mathbf{x}}-e^{i\theta}h\mathbf{s}_{x,p})\right]^T \\
&+\frac{(e^{i\theta}h)^2-1}{2(e^{i\theta}h)^4}\sum_{p=1}^{n}\left[\mathbf{f}(\bar{\mathbf{x}}+e^{i\theta}h\mathbf{s}_{x,p})+\mathbf{f}(\bar{\mathbf{x}}-e^{i\theta}h\mathbf{s}_{x,p})-2\mathbf{f}(\bar{\mathbf{x}})\right] \\
&\qquad\left[\mathbf{f}(\bar{\mathbf{x}}+e^{i\theta}h\mathbf{s}_{x,p})+\mathbf{f}(\bar{\mathbf{x}}-e^{i\theta}h\mathbf{s}_{x,p})-2\mathbf{f}(\bar{\mathbf{x}})\right]^T
\end{aligned} \tag{6.31}$$

Similarly, the cross-covariance can be derived as

$$\begin{aligned}
P_{xy} &= \mathrm{E}\left\{[\mathbf{x}-\bar{\mathbf{x}}][\mathbf{y}-\bar{\mathbf{y}}]^T\right\} \\
&= \mathrm{E}\left\{[S_x\Delta\mathbf{z}]\left[\tilde{f}(\bar{\mathbf{z}})+\tilde{D}_{\Delta z}\mathbf{f}+\frac{1}{2}\tilde{D}^2_{\Delta z}\mathbf{f}-\tilde{f}(\bar{\mathbf{z}})-\frac{1}{2}\mathrm{E}\left\{\tilde{D}^2_{\Delta z}\mathbf{f}\right\}\right]^T\right\} \\
&= \mathrm{E}\left\{[S_x\Delta\mathbf{z}]\left[\tilde{D}_{\Delta z}\tilde{\mathbf{f}}\right]^T\right\} \\
&= \frac{1}{2e^{i\theta}h}\sum_{p=1}^{n}[\mathbf{s}_{x,p}]\left[\mathbf{f}(\bar{\mathbf{x}}+e^{i\theta}h\mathbf{s}_{x,p})-\mathbf{f}(\bar{\mathbf{x}}-e^{i\theta}h\mathbf{s}_{x,p})\right]^T
\end{aligned} \tag{6.32}$$

again, odd moments evaluated to zero.

## 6.5 Second-Order Complex Divided Difference Filter

This section summarizes the algorithms for a second-order complex-step filter based on the DD2 filter. For the resemblance with DD2 filter, the interested reader is again referred to Ref. [54] for complete derivations.

The filter starts off with initializations of states, process noise covariance, measurement covariance and states error covariance. The filter then enters into measurement update and propagation loop until the last available measurement. The measurement update is sometimes referred as the *a posteriori* update while propagation is also called the *a priori* update or time update. For other common filter algorithms refer to Chapter 4.

Common notations are superscript of "+" denotes updated values, "−" denotes propagated values and subscript of "$k$" or in parenthesis denotes time index. A computationally efficient Cholesky square factor is used to maintain (and update if necessary) the covariances at square root level. A Householder triangulation is used as an efficient way to maintain the square Cholesky factor for the rectangular matrix.

## Dynamical and Measurement Models

1. The system dynamics and measurement model are modeled as

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \quad , \quad \mathbf{v}_k \sim \mathcal{N}(\bar{\mathbf{v}}, Q_k) \tag{6.33}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{w}_k) \quad , \quad \mathbf{w}_k \sim \mathcal{N}(\bar{\mathbf{w}}, R_k) \tag{6.34}$$

   where $\mathbf{v}_k$ and $\mathbf{w}_k$ are i.i.d. (independent & identically distributed) random noise with given means and covariances.

## Initialization

1. Initialize the states, error covariance

$$\mathbf{x}_0^- \quad , \quad P_0^- \tag{6.35}$$

2. Find the cholesky of the process noise covariance, measurement noise covariance and

error covariance

$$Q = S_v S_v^T \quad , \quad R = S_w S_w^T \tag{6.36}$$

$$P^- = S_x^- S_x^{-T} \quad , \quad P^+ = S_x^+ S_x^{+T} \tag{6.37}$$

**Measurement Update**

1. With $S_{yx^-}^{(1)}(k) = \left[ S_{yx^-}^{(1)}(p,q) \right]$ and similarly for other terms, compute the following terms

$$S_{yx^-}^{(1)}(k) = \frac{1}{2e^{i\theta}h} \left[ \mathbf{g}_p(\mathbf{x}_k^- + e^{i\theta}h\mathbf{s}_{x,q}^-, \bar{\mathbf{w}}_k) - \mathbf{g}_p(\mathbf{x}_k^- - e^{i\theta}h\mathbf{s}_{x,q}^-, \bar{\mathbf{w}}_k) \right] \tag{6.38a}$$

$$S_{yw}^{(1)}(k) = \frac{1}{2e^{i\theta}h} \left[ \mathbf{g}_p(\mathbf{x}_k^-, \bar{\mathbf{w}}_k + e^{i\theta}h\mathbf{s}_{w,q}) - \mathbf{g}_p(\mathbf{x}_k^-, \bar{\mathbf{w}}_k - e^{i\theta}h\mathbf{s}_{w,q}) \right] \tag{6.38b}$$

$$S_{yx^-}^{(2)}(k) = \frac{\sqrt{(e^{i\theta}h)^2 - 1}}{2(e^{i\theta}h)^2} \left[ \mathbf{g}_p(\mathbf{x}_k^- + e^{i\theta}h\mathbf{s}_{x,q}^-, \bar{\mathbf{w}}_k) + \mathbf{g}_p(\mathbf{x}_k^- - e^{i\theta}h\mathbf{s}_{x,q}^-, \bar{\mathbf{w}}_k) - 2\mathbf{g}_p(\mathbf{x}_k^-, \bar{\mathbf{w}}_k) \right] \tag{6.38c}$$

$$S_{yw}^{(2)}(k) = \frac{\sqrt{(e^{i\theta}h)^2 - 1}}{2(e^{i\theta}h)^2} \left[ \mathbf{g}_p(\mathbf{x}_k^-, \bar{\mathbf{w}}_k + e^{i\theta}h\mathbf{s}_{w,q}) + \mathbf{g}_p(\mathbf{x}_k^-, \bar{\mathbf{w}}_k - e^{i\theta}h\mathbf{s}_{w,q}) - 2\mathbf{g}_p(\mathbf{x}_k^-, \bar{\mathbf{w}}_k) \right] \tag{6.38d}$$

2. Compute the estimated measurement

$$\begin{aligned} \bar{\mathbf{y}}_k = &\frac{(e^{i\theta}h)^2 - n_x - n_w}{(e^{i\theta}h)^2} \mathbf{g}(\mathbf{x}_k^-, \bar{\mathbf{w}}_k) \\ &+ \frac{1}{2(e^{i\theta}h)^2} \sum_{p=1}^{n_x} \mathbf{g}(\mathbf{x}_k^- + e^{i\theta}h\mathbf{s}_{x,p}^-, \bar{\mathbf{w}}_k) + \mathbf{g}(\mathbf{x}_k^- - e^{i\theta}h\mathbf{s}_{x,p}^-, \bar{\mathbf{w}}_k) \\ &+ \frac{1}{2(e^{i\theta}h)^2} \sum_{p=1}^{n_w} \mathbf{g}(\mathbf{x}_k^-, \bar{\mathbf{w}}_k + e^{i\theta}h\mathbf{s}_{w,p}) + \mathbf{g}(\mathbf{x}_k^-, \bar{\mathbf{w}}_k - e^{i\theta}h\mathbf{s}_{w,p}) \end{aligned} \tag{6.39}$$

3. Perform a Householder Triangulation

$$S_y(k) = \mathcal{H}\left\{\left[\begin{array}{cccc} S^{(1)}_{yx^-}(k) & S^{(1)}_{yw}(k) & S^{(2)}_{yx^-}(k) & S^{(2)}_{yw}(k) \end{array}\right]\right\} \tag{6.40}$$

where $\mathcal{H}\{\cdot\}$ denotes the Householder Triangulation operation.

4. Calculate the Kalman gain, $K_k$,

$$P_{xy}(k) = S_x^-(k) S^T_{yx^-}(k) \tag{6.41}$$

$$K_k = P_{xy}(k)\left[S_y(k) S_y^T(k)\right]^{-1} \tag{6.42}$$

5. Update states estimate, $\mathbf{x}, k$, and $S_x(k)$

$$\mathbf{x}_k^+(k) = \mathbf{x}_k^- + K_k(\tilde{\mathbf{y}}_k - \mathbf{y}_k^-) \tag{6.43}$$

$$S_x^+(k) = \mathcal{H}\left\{\left[\begin{array}{cccc} S_x^- - K_k S^{(1)}_{yx}(k) & K_k S^{(1)}_{yw}(k) & K_k S^{(2)}_{yx} & K_k S^{(2)}_{yw}(k) \end{array}\right]\right\} \tag{6.44}$$

6. If the error covariance matrix is desired, can be computed via $P^+ = S_x^+ S_x^{+T}$ or

$$\begin{aligned} P^+(k) = {} & \left[S_x^- - K S^{(1)}_{yx}\right]\left[S_x^- - K S^{(1)}_{yx}\right]^T + \left[K S^{(1)}_{yw}\right]\left[K S^{(1)}_{yw}\right]^T \\ & + \left[K S^{(2)}_{yx}\right]\left[K S^{(2)}_{yx}\right]^T + \left[K S^{(2)}_{yw}\right]\left[K S^{(2)}_{yw}\right]^T \end{aligned} \tag{6.45}$$

## Propagation

1. Calculate the following terms

$$S^{(1)}_{xx^+}(k) = \frac{1}{2e^{i\theta}h}\left[\mathbf{f}_p(\mathbf{x}_k^+ + e^{i\theta}h\mathbf{s}^+_{x,q}, \mathbf{u}_k, \bar{\mathbf{v}}_k) - \mathbf{f}_p(\mathbf{x}_k^+ - e^{i\theta}h\mathbf{s}^+_{x,q}, \mathbf{u}_k, \bar{\mathbf{v}}_k)\right] \tag{6.46}$$

$$S_{xv}^{(1)}(k) = \frac{1}{2e^{i\theta}h}\left[\mathbf{f}_p(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k + e^{i\theta}h\mathbf{s}_{v,q}) - \mathbf{f}_p(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k - e^{i\theta}h\mathbf{s}_{v,q})\right] \tag{6.47}$$

$$S_{xx^+}^{(2)}(k) = \frac{\sqrt{(e^{i\theta}h)^2 - n_x - n_w}}{2(e^{i\theta}h)^2}\Bigg[\mathbf{f}_p(\mathbf{x}_k^+ + e^{i\theta}h\mathbf{s}_{x,q}^+, \mathbf{u}_k, \bar{\mathbf{v}}_k) + \mathbf{f}_p(\mathbf{x}_k^+ - e^{i\theta}h\mathbf{s}_{x,q}^+, \mathbf{u}_k, \bar{\mathbf{v}}_k) - 2\mathbf{f}_p(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k)\Bigg] \tag{6.48}$$

$$S_{xv}^{(2)}(k) = \frac{\sqrt{(e^{i\theta}h)^2 - n_x - n_w}}{2(e^{i\theta}h)^2}\Bigg[\mathbf{f}_p(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k + e^{i\theta}h\mathbf{s}_{v,q}) + \mathbf{f}_p(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k - e^{i\theta}h\mathbf{s}_{v,q}) - 2\mathbf{f}_p(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k)\Bigg] \tag{6.49}$$

2. Propagate the states

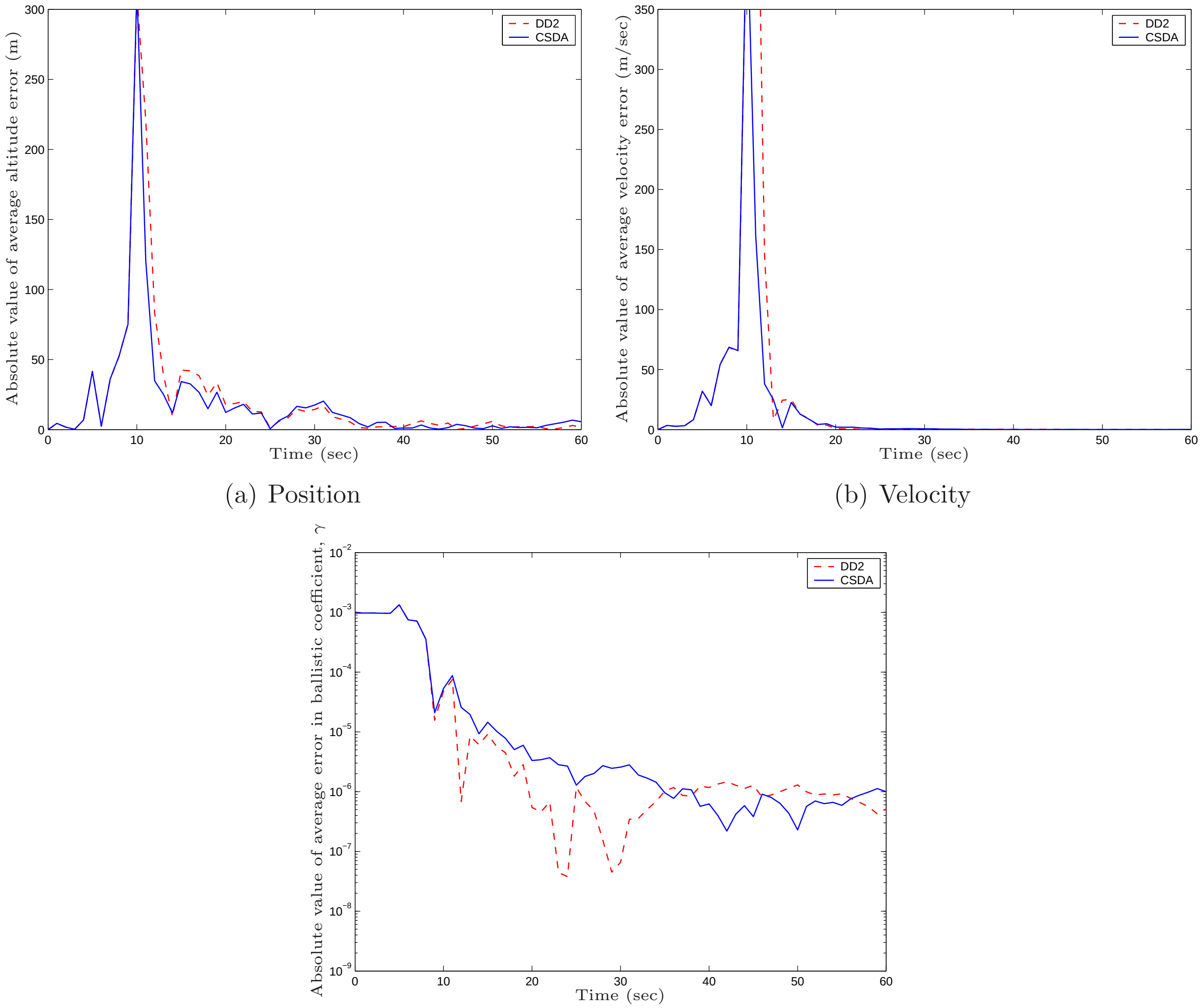
$$\begin{aligned}\mathbf{x}_{k+1}^- = {} & \frac{(e^{i\theta}h)^2 - n_x - n_v}{(e^{i\theta}h)^2}\mathbf{f}(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k) \\ & + \frac{1}{2(e^{i\theta}h)^2}\sum_{p=1}^{n_x}\mathbf{f}(\mathbf{x}_k^+ + e^{i\theta}h\mathbf{s}_{x,p}^+, \mathbf{u}_k, \bar{\mathbf{v}}_k) + \mathbf{f}(\mathbf{x}_k^+ - e^{i\theta}h\mathbf{s}_{x,p}^+, \mathbf{u}_k, \bar{\mathbf{v}}_k) \\ & + \frac{1}{2(e^{i\theta}h)^2}\sum_{p=1}^{n_x}\mathbf{f}(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k + e^{i\theta}h\mathbf{s}_{v,p}) + \mathbf{f}(\mathbf{x}_k^+, \mathbf{u}_k, \bar{\mathbf{v}}_k - e^{i\theta}h\mathbf{s}_{v,p})\end{aligned} \tag{6.50}$$

3. And the propagation of $S_x$

$$S_x^-(k+1) = \mathcal{H}\left\{\left[\begin{array}{cccc} S_{xx^+}^{(1)}(k) & S_{xv}^{(1)}(k) & S_{xx^+}^{(2)}(k) & S_{xv}^{(2)}(k)\end{array}\right]\right\} \tag{6.51}$$

## 6.6 Performance Evaluation

This section presents some performance related figures to compare the performance of the standard DD2 and CSDA filters. The sample problem chosen for the comparison was Athan's problem [48]. This is the same problem used for second-order Kalman filter evaluation in §5.4 and thus the dynamical and measurements are not repeated here. All simulations in



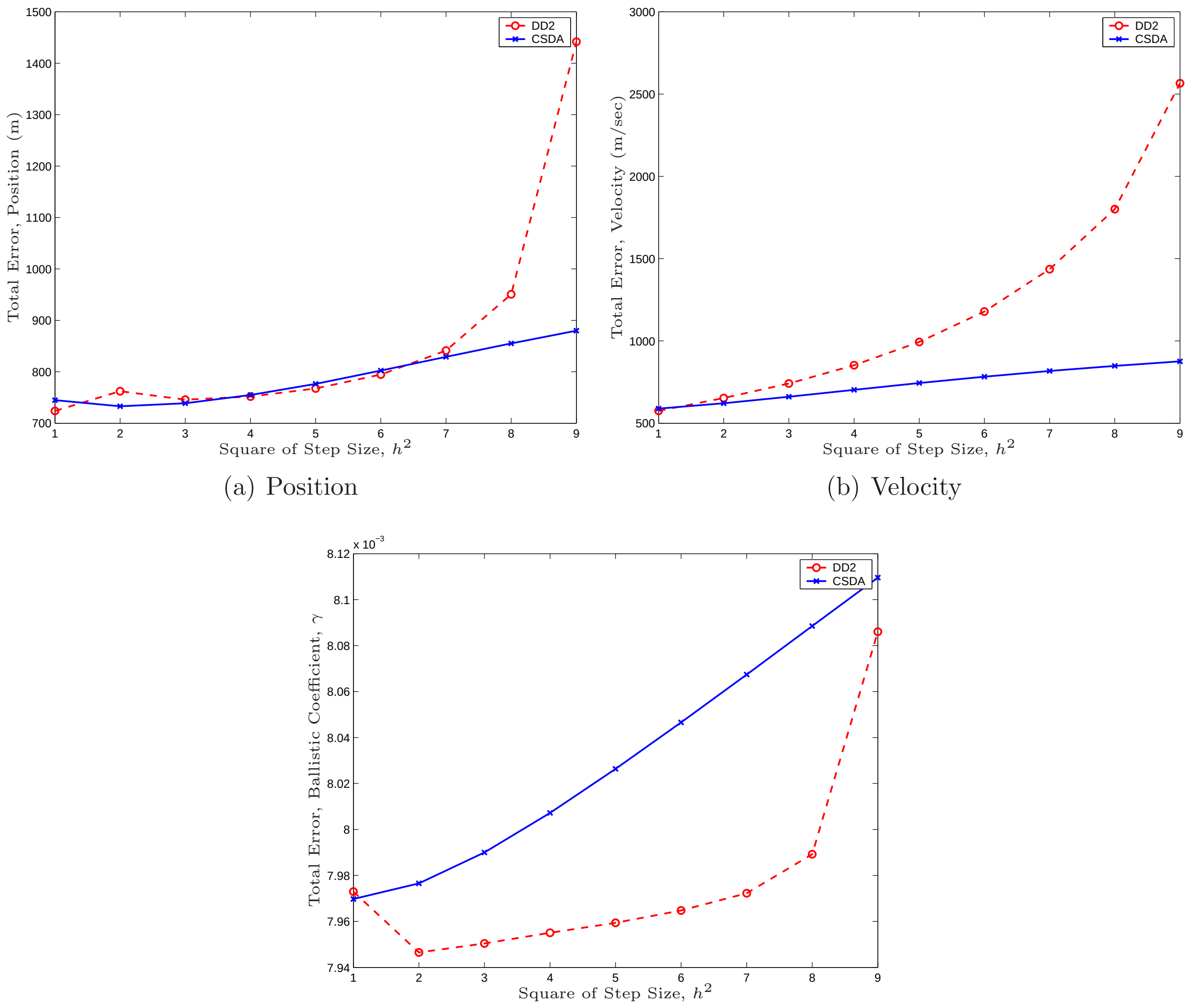
(a) Position

(b) Velocity

(c) Ballistic Coefficient

Figure 6.1: A sample states estimate output from DD2 and CSDA filters. Simulations performed with $h^2 = 2$, CSDA angle= $45°$ and measurement covariance $R = 1 \times 10^4$

this section are run with time step (also the sampling interval) of 1 sec for 50 sec. Fifty Monte Carlo runs are performed and their average is used for performance comparison. A 4th Runge-Kutta is used to propagate the states between each sampling interval. A few variables that are manipulated to assess the performance comparison include $h^2$, the CSDA angle and the measurement covariance. Nominally, $h^2 = \sqrt{3}$ for a Gaussian distribution and the measurement covariance is $1 \times 10^4$ m$^2$, except when its influence on performance is being



(a) Position

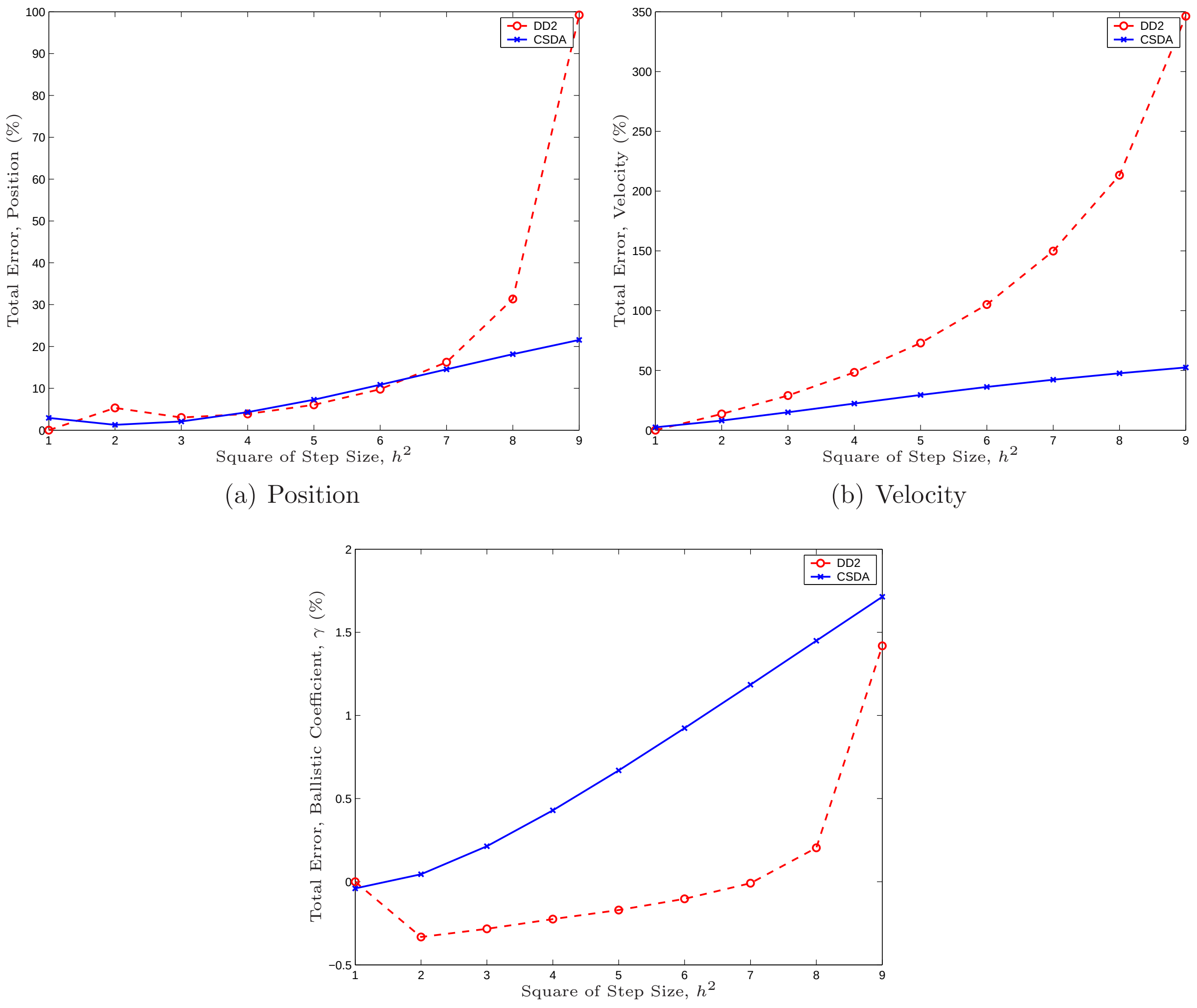
(b) Velocity

(c) Ballistic Coefficient

Figure 6.2: Absolute error, varying $h^2$. Simulations performed with CSDA angle= 45° and measurement covariance $R = 1 \times 10^4$.

evaluated.

Figure 6.1 shows a sample of state estimation from both the DD2 and CSDA filters. The CSDA angle is set to 90°. The simulation is run with measurement model covariance of $2 \times 10^4$ m$^2$ instead of the "nominal" value of $1 \times 10^4$ m$^2$. Figures 6.2-6.6 show the performance comparison between DD2 and CSDA filter. Figure 6.2 compares the total absolute between the two filters but subsequent figures compare them in terms of percentage.



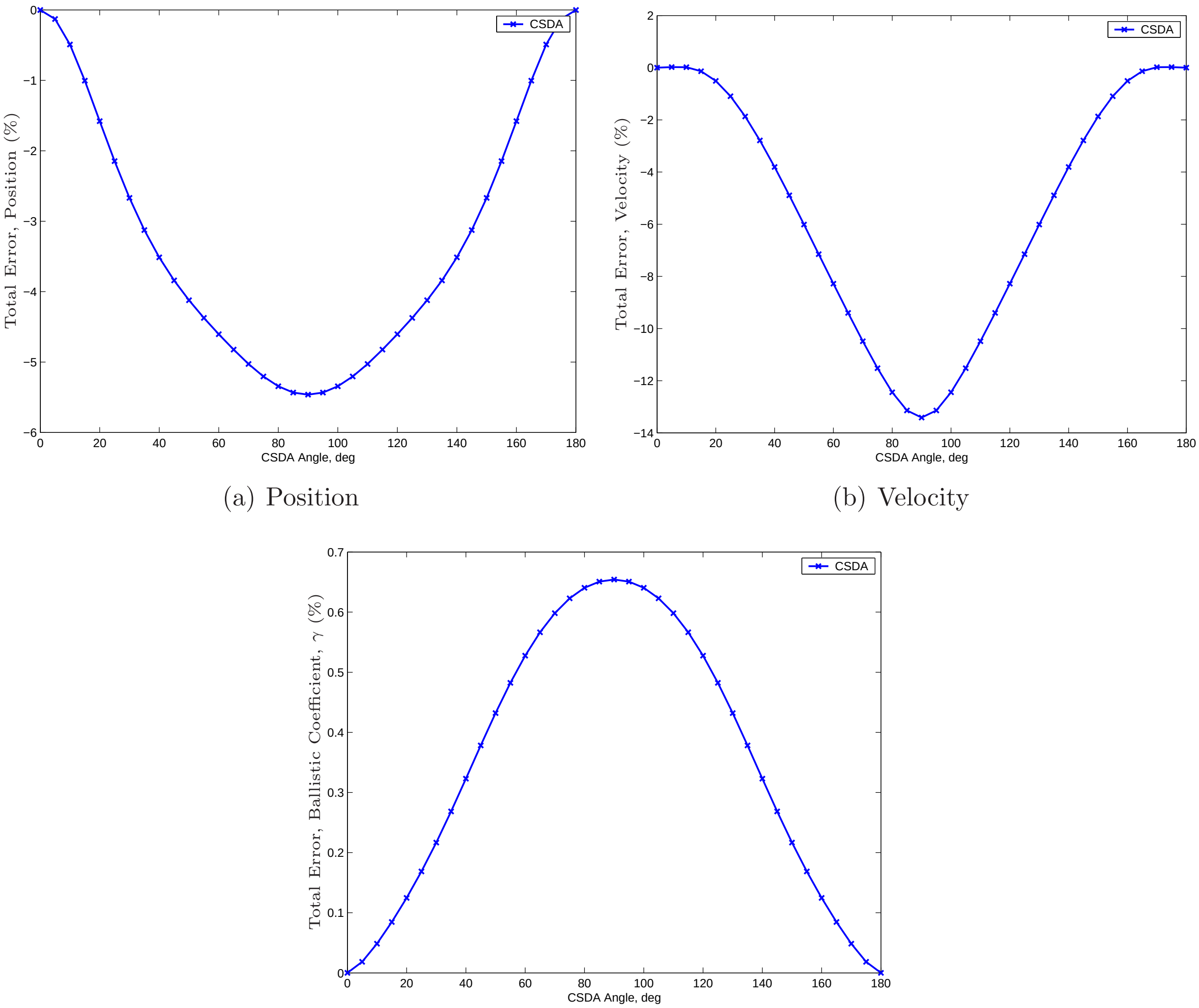
(a) Position

(b) Velocity

(c) Ballistic Coefficient

Figure 6.3: Percentage of error, varying $h^2$. Simulations performed with CSDA angle= 45° and measurement covariance $R = 1 \times 10^4$.

For Figs. 6.2 and 6.3, the CSDA angle of 45° is used since it has same magnitude for both the real and imaginary components. In these figures, $h^2$ is increased from 1 until one of the filter becomes unstable. This is also a way to evaluate the Gaussianity of the system. Then in Fig. 6.4, the CSDA angle is varied from 0° (which is essentially standard DD2 filter) to 180° to look for the most suitable angle. Next in Fig. 6.5, different $h^2$ values are tested again. Finally in Fig. 6.6, the measurement covariance is slowly increased from $1 \times 10^4$ m$^2$



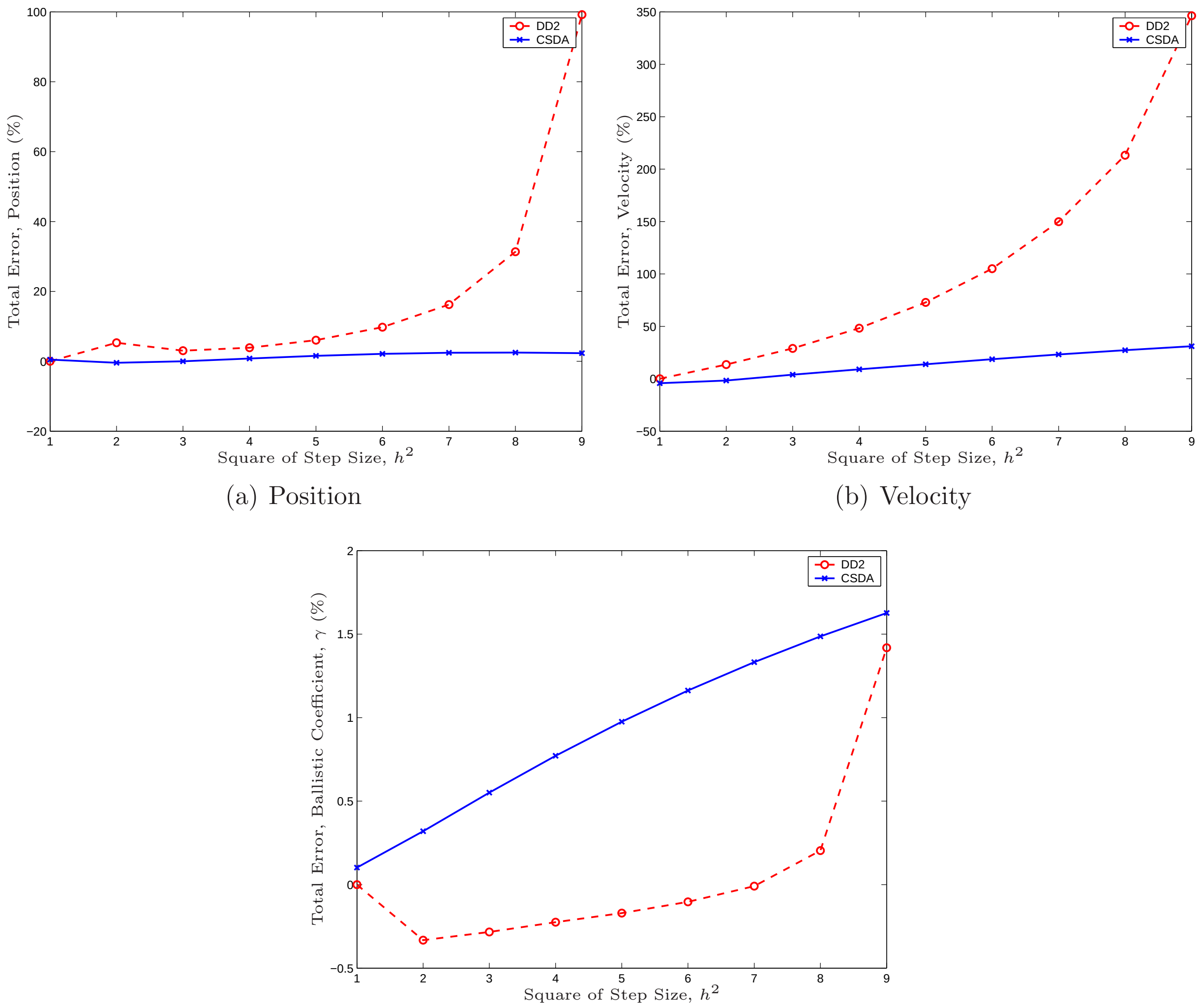
(a) Position

(b) Velocity

(c) Ballistic Coefficient

Figure 6.4: Percentage of error, varying CSDA angle. Simulations performed with $h^2 = 2$ and measurement covariance $R = 1 \times 10^4$.

to the point of an unstable filter condition. The absolute error is first used as a performance measure to gauge range of performance. This is the total error over the 50 sec filter run averaging from fifty Monte Carlo simulations. Later, only the percentage difference is used to more accurately present the relative performance change. The DD2 results are first used as a performance base for each of the comparisons.



(a) Position

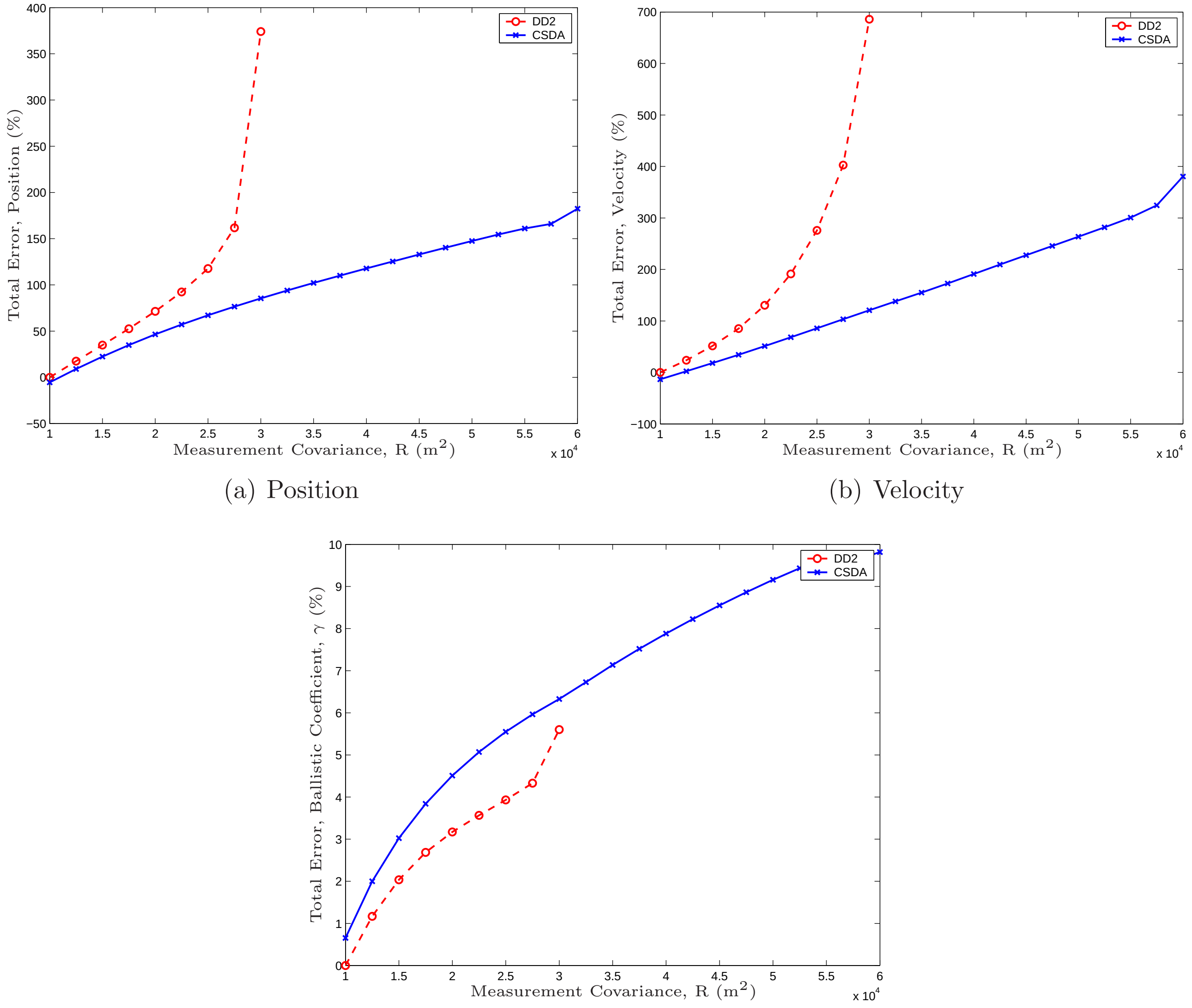
(b) Velocity

(c) Ballistic Coefficient

Figure 6.5: Percentage of error, varying $h^2$. Simulations performed with CSDA angle= 90° and measurement covariance $R = 1 \times 10^4$.

## 6.7 Discussion

Figure 6.1 gives the first glance into behavior of the DD2 and CSDA filters. It clearly shows that the CSDA filter is better in both position and velocity estimates. At about 10 secs into simulation the system experiences the lowest observability and thus is most challenging for the filters. Physically, this occurs when the object is on the same horizontal level as the radar. The CSDA filter consistently outperforms the DD2 filter during this low observability period.



(a) Position

(b) Velocity

(c) Ballistic Coefficient

Figure 6.6: Percentage of error, varying measurement covariance. Simulations performed with $h^2 = 2$ and CSDA angle= 90°.

This indicates that the CSDA is able to better capture the system nonlinearity. However, Fig. 6.1(c) does not show a clear advantage for estimation of the ballistic coefficient. The total absolute error in the ballistic coefficient estimation in almost every case favors the DD2 filter. Though, it is merely by a small percentage margin as shown in subsequent runs. The DD2 filter shows more abrupt movement in estimating the ballistic coefficient, which is more smooth for the CSDA filter.

Figure 6.2 evaluates the effect of $h^2$ in the form of total estimation error of each state. Figure 6.3 shows the same information but as a percentage change based on the first DD2 results. From now onwards, the percentage change will be used as relative performance index. A value of $h^2 = \sqrt{3}$ is optimal for Gaussian systems, but obviously our nonlinear system is not be very Gaussian. In reference to Fig. 6.3, low $h^2$ values work better for both filters. At $h^2 = 1$, both filters exhibit the best possible performance characteristics for position and velocity estimation. Conversely, estimation error of ballistic coefficient is minimized at $h^2 = 2$. Estimation error in the ballistic coefficient for CSDA filter, however, bottoms at $h^2 = 1$ and increases with increasing $h^2$. For these reasons, $h^2 = 2$ embodies a good compromise for all three states and this will be used for succeeding runs. The DD2 filter becomes unstable at $h^2$ beyond 9, but the CSDA filter is still able to give reasonable estimates. Moreover, the performance deteriorates a lot slower than the DD2 filter, once again showing the robustness of the CSDA filter.

Now let's examine the optimal angle for the CSDA filter. Figure 6.4 shows the effect of different angles from 0° (this reduces down to just plain DD2 filter) to 180°. It clearly reveals that 90° is the most optimal angle, which means using just a purely imaginary number, $i$. At this angle, the position estimate is improved by over 5% while the velocity estimate is improved impressively by over 12%. The less than 0.7% deterioration of the ballistic coefficient estimate is negligible compared to the vast performance gained in both position and velocity estimate. Thus $e^{90^\circ i}h$ or just $ih$ should be used as the step size.

Now let's examine the effect of $h^2$ with the optimal CSDA angle. Figure 6.5 shows that the sensitivity of the newly improved CSDA filter to various $h^2$ is significantly reduced. The CSDA has so far shown to be more robust than DD2 filter in almost every measure. The last test is the influence of measurement noise. The measurement noise is slowly increased until the filters become unstable. From Fig. 6.6, the DD2 filter becomes unstable with measurement covariance, $R$, greater than $3 \times 10^4$, however, it takes $R$ to be greater than $6 \times 10^4$ to make the CSDA filter unstable. Figures 6.6(a) and 6.6(b) show the "terminal" position and velocity estimates of the CSDA filter are still within the limit of operation for the DD2 filter, however, the ballistic coefficient estimate for the CSDA filter at $R = 6 \times 10^4$ is evidently higher than the limit of DD2 filter. Thus it is speculated that the failure of CSDA filter beyond $R = 6 \times 10^4$ may be due to unreasonably accuracy in the ballistic coefficient estimate.

## 6.8 Conclusion

In this chapter, the first and second-order CSDA were used in substitution of the first and second-order central divided (finite) difference formulae in the derivation of the DD2 filter. The analysis shows that it is as easy as substituting $h$ in central divided-difference formulae with one involving a unity magnitude complex number, $e^{i\theta}h$. This method generalized the DD2 filter with a complex step size. Assessments were carried out to determine suitable values for $h^2$ and CSDA angle. Also, the robustness of the CSDA was shown with higher measurement noise. All measures favor the CSDA filter. These proved the robustness of CSDA filter in the face of high nonlinearity. The square root level filtering of the DD2 and CSDA filters may bear numerical stability advantages in handling extremely small variables or states with large magnitude differences among them. In conclusion, the complex-step DD2 filter (or CSDA filter) should replace the implementation of traditional DD2 and Unscented filters.

# Chapter 7

# Conclusion & Future Work

## 7.1 Conclusion

This dissertation further enhanced the significance of the complex number to the engineering community, particularly for state estimation. The complex step size was applied to the Taylor series expansion, which in turn was used to derive first and second derivative approximations, using the proper *angle*, i.e. the right mix of real and imaginary components with unity magnitude from the origin. The crucial part of the series truncation error is placed in the real domain and if the imaginary part of the analysis is taken into consideration, the real part naturally does not play a part to the output. As a result, the truncation error is reduced to the next series with imaginary component. The extension to the vector case allowed vast implementation of the Jacobian and Hessian matrices derived from the complex-step derivative approximation (CSDA) to virtually anywhere a conventional numerical finite difference is applied.

The CSDA generated Jacobian and Hessian information was applied to a nonlinear system for state estimation. Due to a lower truncation error, the second-order Kalman filter (SOKF) using the CSDA proved to be more robust compared to the standard SOKF with Jacobian and Hessian information obtained from standard finite difference approaches. The CSDA was then applied directly in the derivation of a new filter that linearizes the statistical properties instead of the nonlinear equations. Derivation of the second-order Divided Difference (DD2) filter was closely followed with a replacement of complex step size. The result was a remarkably robust filter in the face of severe nonlinearity, particularly with large deviations from the true mean present. This CSDA-based DD2 filter (or simply the CSDA filter) also executes faster than the SOKF while improving the robustness. The increased robustness is mainly due to the factthat the CSDA filter employs an extrapolation approximation instead of precision point Taylor series expansion in the region of interest. Furthermore, as in the DD2 filter, the CSDA filter does not require tuning of numerous parameters involved in other filters, such as the Unscented filter.

Simpler analytical expressions with higher accuracy saved a lot of tedious analytical endeavors. For these advantages, the complex step size approach is recommended to applications where the step size is required and used an intermediate step to find other information.

## 7.2 Future Work

The benefits of evaluating functions with a complex step do not end here. This dissertation barely scratched the potential of the complex-step function evaluation and leaves much room for improvement to many other fields. Future research work on this topic includes, but is not limited to:

1. Using double-double precision to further decrease roundoff error.

2. Investigation of the accuracy of various components in the Hessian matrix.

3. Function evaluation with complex number are often a bit slower and not supported in many programming languages. There may be some hardware level optimization possible to handle complex numbers. Standard mathematical library could be developed for various programming languages to facilitate development in this field.

4. Show an analysis to obtain the optimal step size, especially for the case of obtaining the second derivative or the Hessian matrix.

5. Higher-order filters could be derived by replacing the first- and second-order CSDA in deriving the CSDA filter with a Richardson extrapolation.

6. It is possible to further generalize the first- and second-order to higher-order derivatives or non-central CSDA depending on the applications.

7. Partial derivative has been hailed as better representation of hard nonlinearity, including geometric modeling of clouds and coastlines. It may be possible to generalize the CSDA to derivative of non-integer order.

8. Application of the CSDA to multidisciplinary design optimizations, including various optimization techniques, such as the second-order Newton's method, where Jacobian or Hessian information is needed.

9. CSDA filter based smoother or even particle filter.

10. Possible application of the complex step approach to the control field.

11. Hyper-complex step-size extension of CSDA, for example, the quaternions, or even octonions.

# Bibliography

[1] Greenberg, M. D., *Advanced Engineering Mathematics*, Prentice Hall, Upper Saddle River, New Jersey, 2nd ed., 1998.

[2] Lyness, J. N., "Numerical Algorithms Based on the Theory of Complex Variable," Proceedings - A.C.M. National Meeting, 1967, pp. 125–133.

[3] Groemer, H., *Geometric Applications of Fourier Series and Spherical Harmonics*, Cambridge University Press, New York, NY, 1996.

[4] Hobson, E. W., *The theory of Spherical and Ellipsoidal Harmonics*.

[5] Stein, E. and Weiss, G., *Fourier Analysis on Euclidean Spaces*, Princeton University Press, Princeton, NJ, 1971.

[6] McLean, S., Macmillan, S., Maus, S., Lesur, V., Thomsan, A., and Dater, D., "The US/UK World Magnetic Model for 2005-2010," Tech. rep., NOAA Technical Report NESDIS/NGDC-1, Dec. 2004.

[7] Langel, R. A., "The Main Field," *Geomagnetism*, edited by J. A. Jacobs, Academic Press, Orlando, FL, 1987, pp. 249–512.

[8] Campbell, W. H., *Introduction to Geomagnetic Fields*, Cambridge University Press, 1987.

[9] Merrill, R. T. and McElhinny, M. W., *The Earth's Magnetic Field - Its History, Origin and Planetary Prespective*, Vol. 32 of *International Geophysics Series*, Academic Press, London, 1983.

[10] Roithmayr, C. M., "Contributions of Spherical Harmonics to Magnetic and Gravitational Fields," Tech. rep., (NASA/TM-2004-213007) Langley Research Center, Hampton, Virginia, Mar. 2004.

[11] Pines, S., "Uniform Representation of the Gravitational Potential and its Derivatives," *AIAA Journal*, Vol. 11, Nov. 1973, pp. 1508–1511.

[12] Jung, H. and Psiaki, M. L., "Tests of Magnetometer/Sun-Sensor Orbit Determination Using Flight Data," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 3, May-June 2002, pp. 582–590.

[13] Psiaki, M. L., Martel, F., and Pal, P. K., "Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, May-June 1990, pp. 506–514.

[14] Psiaki, M. and Martel, F., "Autonomous Magnetic Navigation for Earth Orbiting Spacecraft," *Proceedings of thge Third Annual AIAA/USU Conference on Small Satellites*, Longan, UT, Sept. 1989.

[15] Psiaki, M. L., "Autonomous Low-Earth-Orbit Determination from Magnetometer and Sun Sensor Data," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, March-April 1999, pp. 296–304.

[16] Psiaki, M. L., "Autonomous Orbit and Magnetic Field Determination Using Magnetometer and Star Sensing Data," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, May-June 1995, pp. 584–592.

[17] "Current State of the Art on Multidisciplinary Design Optimization (MDO)," AIAA White Paper, 1991, American Institute of Aeronautics and Astronautics, ISBN 1-56347-021-7.

[18] "A Summary of Industry MDO Applications and Needs," AIAA White Paper, 1998, American Institute of Aeronautics and Astronautics, http://endo.sandia.gov/AIAA_MDOTC/sponsored/mao98_whitepaper.html.

[19] Padula, S. L., Alexandrov, N., and Green, L. L., "MDO Test Suite at NASA Langley Research Center," *Sixth AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (AIAA 96-4028) Bellevue, Washington, September 1996.

[20] Bischof, C., Carle, A., Corliss, G., Griewank, A., and Hovland, P., "ADIFOR - Generating Derivative Codes from Fortran Programs," *Scientific Programming*, 1992.

[21] Bischof, C., Carle, A., Khademi, P., and Mauer, A., "The ADIFOR 2.0 System for the Automatic Differentiation of Fortran 77," Tech. Rep. CRPC-TR94491, Center for Research on Parallel Computation, Rice University, Houston, TX, 1994.

[22] Boudjemaa, R., Cox, M. G., Forbes, A. B., and Harris, P. M., "Automatic Differentiation: Techniques and their Application in Metrology," NPL Report CMSC 26/03, National Measurement Directorate, 2003.

[23] Lyness, J. N. and Moler, C. B., "Numerical Differentiation of Analytic Functions," *SIAM Journal for Numerical Analysis*, Vol. 4, No. 2, June 1967, pp. 202–210.

[24] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The Connection Between the Complex-Step Derivative Approximation and Algorithmic Differentiation," *AIAA Paper* 2001-0921, Jan. 2001.

[25] Kim, J., Bates, D. G., and Postlethwaite, I., "Nonlinear Robust Performance Analysis using Complex-Step Gradient Approximations," *Automatica*, Vol. 42, No. 2, 2006, pp. 177–182.

[26] Cerviño, L. I. and Bewley, T. R., "On the extension of the complex-step derivative technique to pseudospectral algorithms," *Journal of Computational Physics*, Vol. 187, No. 2, 2003, pp. 544–549.

[27] Squire, W. and Trapp, G., "Using Complex Variables to Estimate Derivatives of Read Functions," *SIAM Review*, Vol. 40, No. 1, Mar. 1998, pp. 110–112.

[28] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The Complex-Step Derivative Approximation," *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, Sept. 2003, pp. 245–262.

[29] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The Connection Between the Complex-Step Derivative Approximation and Algorithmic Differentiation," *American Institute of Aeronautics and Astronautics*, 2001, AIAA-2001-0921.

[30] Martins, J. R. R. A., Kroo, I. M., and Alonso, J. J., "An Automated Method for Sensitivity Analysis Using Complex Variables," *American Institute of Aeronautics and Astronautics*, 2000, AIAA-2000-0689.

[31] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, No. 1, March 2005, pp. 33–62.

[32] Alonso, J. J., LeGresley, P., van der Weide, E., Martins, J. R. R. A., and Reuther, J. J., "pyMDO: A Framework for High-Fidelity Multi-Disciplinary Optimization," *AIAA Paper* 2004-4480, Aug. 2004.

[33] Pozrikidis, C., *Numerical Computation in Science and Engineering*, chap. 1, Oxford University Press., New York, NY, 1998, pp. 45–47.

[34] Turner, J. D., "Quaternion-Based Partial Derivative and State Transition Matrix Calculations for Design Optimization," *AIAA Paper* 2002-0447, 2002.

[35] Nørgaard, M., Poulsen, N. K., and Ravn, O., "New Developments in State Estimation for Nonlinear Systems," *Automatica*, Vol. 36, No. 11, Nov. 2000, pp. 1627–1638.

[36] Mathews, J. H. and Fink, K. D., *Numerical Methods Using Matlab*, Pearson Prentice Hall, fourth edition ed., 2004.

[37] West, B. J., Bologna, M., and Grigolini, P., *Physics of Fractal Operators*, Institude for Nonlinear Science, Springer New York, New York, 2003.

[38] Miller, K. S. and Ross, B., *An Introduction to the Fractional Calculus and Fractional Differential Equations*, John Wiley and Sons, New York, 1993.

[39] Kalman, Rudolph, E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45.

[40] Jazwinski, A. H., *Stochastic Processes and Filtering theory*, Vol. 64 of *Mathematics in Science and Engineering*, Academic Press, New York, 1970.

[41] Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic System*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[42] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., "A New Approach for Filtering Nonlinear Systems," *American Control Conference*, Seattle, WA, June 1995, pp. 1628–1632.

[43] Wan, E. A. and van der Merwe, R., “The Unscented Kalman Filter for Nonlinear Estimation,” *in Proc. of IEEE Symposium 2000 (AS-SPCC)*, Lake Louise, Alberta, Canada, October 2000.

[44] Wan, E. and van der Merwe, R., “The Unscented Kalman Filter,” *Kalman Filtering and Neural Networks*, edited by S. Haykin, chap. 7, John Wiley & Sons, New York, NY, 2001.

[45] Julier, S. J., “The Scaled Unscented Transformation,” *Proceedings of the American Control Conference*, Anchorage, AK, May 2002, pp. 1108–1114.

[46] Stengel, R. F., *Optimal Control and Estimation*, Dover Publications, New York, NY, 1994.

[47] Chatfield, A. B., *Fundamentals of High Accuracy Inertial Navigation*, chap. 10, American Institute of Aeronautics and Astronautics, Inc., Reston, VA, 1997.

[48] Athans, M., Wishner, R. P., and Bertolini, A., “Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements,” *IEEE Transactions on Automatic Control*, Vol. 13, No. 5, Oct. 1968, pp. 504–514.

[49] Gelb, A., editor, *Applied Optimal Estimation*, The MIT Press, Cambridge, MA, 1974.

[50] Nam, K. and Tahk, M.-J., “A Second-Order Stochastic Filter Involving Coordonate Transformation,” *IEEE Transactions on Automatic Control*, Vol. 44, No. 3, Mar. 1999, pp. 603–608.

[51] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., “A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators,” *IEEE Transactions on Automatic Control*, Vol. AC-45, No. 3, March 2000, pp. 477–482.

[52] Lefebvre, T., Bruyninckx, H., and Shutter, J. D., "Kalman Filters for Nonlinear Systems: A Comparison of Performance," *International Journal of Control*, Vol. 77, No. 7, May 2004, pp. 639–653.

[53] Fröberg, C.-E., *Introduction to Numerical Analysis*, Addison-Wesley Publishing Company, Reading, MA, second edition ed., 1969.

[54] Nørgaard, M., Poulsen, N. K., and Ravn, O., "Advances in Derivative-Free State Estimation for Nonlinear Systems," Technical Report IMM-REP-1998-15, Department of Mathematical Modelling, DTU, 1998, Revised April 2000.

# Appendix A

# Divided Difference

This appendix collects some higher-order expansions of the "Divided Difference" or Stirling operators for the convenience of deriving filter equations. Only real-valued step sizes will be presented, however, complex step-size expansion can be obtained by simple replacement of $h$ with $e^{i\theta}h$. To begin are the difference and average operators, defined as

$$\delta f(x) = f\left(x + \frac{1}{2}h\right) - f\left(x - \frac{1}{2}h\right) \tag{A.1a}$$

$$\mu f(x) = \frac{1}{2}\left[f\left(x + \frac{1}{2}h\right) + f\left(x - \frac{1}{2}h\right)\right] \tag{A.1b}$$

The higher orders of the $\delta$ operator are given below:

$$\delta f(x) = f\left(x + \frac{1}{2}h\right) - f\left(x - \frac{1}{2}h\right) \tag{A.2a}$$

$$\delta^2 f(x) = f\left(x + h\right) - 2f(x) + f\left(x - h\right) \tag{A.2b}$$

$$\delta^3 f(x) = f\left(x + \frac{3}{2}h\right) - 3f\left(x + \frac{1}{2}h\right) + 3f\left(x - \frac{1}{2}h\right) - f\left(x - \frac{3}{2}h\right) \tag{A.2c}$$

$$\delta^4 f(x) = f\left(x + 2h\right) - 4f\left(x + h\right) + 6f\left(x\right) - 4f\left(x - h\right) + f\left(x - 2h\right) \tag{A.2d}$$

$$\delta^5 f(x) = f\left(x+\frac{5}{2}h\right) - 5f\left(x+\frac{3}{2}h\right) - 10f\left(x-\frac{1}{2}h\right) + 5f\left(x-\frac{3}{2}h\right) - f\left(x-\frac{5}{2}h\right) \tag{A.2e}$$

When the function evaluation of fractional step size is not available, the average operator could apply,

$$\mu\delta f(x) = \frac{1}{2}\Big[f(x+h) - f(x-h)\Big] \tag{A.3a}$$

$$\mu\delta^3 f(x) = \frac{1}{2}\Big[f(x+2h) + 2f(x+h) - 2f(x-h) - f(x-2h)\Big] \tag{A.3b}$$

$$\mu\delta^5 f(x) = \frac{1}{2}\Big[f(x+3h) - 4f(x+2h) + 5f(x+h) - 5f(x-h) + 4f(x-2h) - f(x-3h)\Big] \tag{A.3c}$$

# Appendix B

# Mathematics

This appendix gathers some miscellaneous mathematical formulae that are useful for this dissertation.

## B.1 Trigonometry and Complex Number

The angles are in radian unless otherwise noted. The focus here is on sine and cosine since these are fundamental in trigonometry. Table B.1 summarizes the trigonometric evaluation at some common angles.

Table B.1: Trigonometry of Common Angles

| Angle (deg) | 0° | 30° | 45° | 60° | 90° |
|---|---|---|---|---|---|
| $\sin^2(\theta)$ | $\frac{0}{4}$ | $\frac{1}{4}$ | $\frac{2}{4}$ | $\frac{3}{4}$ | $\frac{4}{4}$ |
| $\cos^2(\theta)$ | $\frac{4}{4}$ | $\frac{3}{4}$ | $\frac{2}{4}$ | $\frac{1}{4}$ | $\frac{0}{4}$ |
| $\tan^2(\theta)$ | $\frac{0}{4}$ | $\frac{1}{3}$ | $\frac{2}{2}$ | $\frac{3}{1}$ | $\frac{4}{0}$ |

### B.1.1 Identities

The Pythagorean theorem states that $c^2 = a^2 + b^2$ and equivalently in sine and cosine

$$\cos^2\theta + \sin^2\theta = 1 \tag{B.1}$$

The summation and subtraction of angles for sine and cosine are

$$\sin(x \pm y) = \sin x \cos y \pm \cos x \sin y \tag{B.2a}$$

$$\cos(x \pm y) = \cos x \cos y \mp \sin x \sin y \tag{B.2b}$$

Euler's identity or Euler's relation bridges the field of geometry and complex domain,

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{B.3}$$

Some important periodicity of trigonometry are:

$$\sin(\theta + 2\pi) = \sin\theta \tag{B.4a}$$

$$\cos(\theta + 2\pi) = \cos\theta \tag{B.4b}$$

$$\sin(-\theta) = -\sin\theta \tag{B.4c}$$

$$\cos(-\theta) = \cos\theta \tag{B.4d}$$

$$ \tag{B.4e}$$

and in terms of Euler's identity

$$e^{i(\theta+\pi)} = -e^{i\theta} \tag{B.5}$$

## B.2 Some Useful Mathematical Functions

1. Kronecker function

$$\delta_{jk} = \begin{cases} 0 & \text{when } j \neq k \\ 1 & \text{when } j = k \end{cases} \tag{B.6}$$

2. Binomial Coefficient

$$\binom{n}{k} = \frac{n!}{k!(n! - k!)} \tag{B.7}$$