Adjoint Based Hessians for Optimization Problems in System Identification

Souransu Nandi and Tarunraj Singh

***Abstract*— An adjoint sensitivity based approach to determine the gradient and Hessian of cost functions for system identification is presented. The motivation is the development of a computationally efficient approach relative to the direct differentiation technique and which overcomes the challenges of the step size selection in finite difference approaches. The discrete time measurements result in discontinuities in the Lagrange multipliers. The proposed approach is illustrated on the Lorenz 63 model where part of the initial conditions and model parameters are estimated.**

I. INTRODUCTION

In data assimilation and system identification of dynamic systems (among other fields), it is often required to solve optimization problems where cost functions are functions of the state variables at specific time instants (most often when measurements are available). When derivative based optimization techniques are used to solve these problems, gradients of the cost with respect to the optimization variables are needed. In addition, for faster convergence, Hessians of the cost with respect to the optimization variables are also desired [1].

However, for dynamic systems, calculating the Hessian of a cost function which is state dependent is computationally very expensive since it requires solving matrix and tensor differential equations. As a result, methods to efficiently evaluate these derivatives become paramount. Raffard et al. [2] propose an adjoint based approach to determine the gradients of a cost function for parameter identification of protein regulatory networks. The Hessian is approximated by the finite difference of the gradients. Numerous papers [3], [4], [5] deal with the use of adjoint based algorithms for optimal control or sensitivity analysis where the cost metric is an integral function. This paper (largely inspired by Tortorelli et al [6] where an adjoint based method is used to determine gradients and Hessians for static optimization problems) focuses on cost functions which are summations of a scalar function, reflecting the availability of measurements at discrete time instants. This results in discontinuities in the evolution of the Lagrange multipliers. The efficiency of the adjoint based approach is compared to existing approaches of Finite Difference and Direct Differentiation to illustrate the computational benefits.

The document has been structured as follows: Section I introduces the problem statement. Section II talks about the existing method of direct differentiation. Section III presents the adjoint method emphasizing the novel way to evaluate Hessians. Finally, the document ends with an system identification example problem for a chaotic dynamic system in section V and concluding remarks in section VI.

Dynamic models considered in this work are of the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}, t) \text{ with } \boldsymbol{x_0} = \boldsymbol{x}(t_0) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector of the system, $\boldsymbol{x}(t_0)$ is the initial condition of the states and $\boldsymbol{p} \in \mathbb{R}^p$ is the parameter vector on which the system model depends.
The generic scalar cost function is considered of the form

$$J(\boldsymbol{x}, \boldsymbol{p}) = \sum_{i=0}^{N} g(\boldsymbol{x}(t_i), \boldsymbol{p}) \tag{2}$$

where $g(\boldsymbol{x}(t_i), \boldsymbol{p})$ represents the value of a scalar function $g(\boldsymbol{x}(t), \boldsymbol{p})$ at the i^{th} time step (t_i). In the context of system identification problems (where the initial conditions and the parameters of the system are being estimated), $g(\boldsymbol{x}(t_i), \boldsymbol{p})$ could be regarded as the squared residual between the output model and the observations of the true plant. However, one is not restricted to a squared residual: any desired cost with $\mathcal{C}^2$ continuity may be used.
For simplicity, in all subsequent equations, $J(\boldsymbol{x}, \boldsymbol{p})$ is written as J. Similarly, $g(\boldsymbol{x}(t_i), \boldsymbol{p})$ and $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}, t)$ have also been simplified to just g_i and $\boldsymbol{f}$ respectively.
From an optimization point of view, it is required to find a $\boldsymbol{p}$ and an $\boldsymbol{x_0}$ which minimizes J. The objective of this work is to facilitate use of all derivative based optimization techniques to solve the problem, by developing efficient ways to determine the gradients and Hessians of the cost function with respect to the optimization variables (i.e. $\boldsymbol{p}$ and $\boldsymbol{x_0}$). If the optimization variables are grouped as $\boldsymbol{q} = [\boldsymbol{p}, \boldsymbol{x_0}]^T$ where $\boldsymbol{p} = [p_1, \ldots, p_p]^T$ and $\boldsymbol{x_0} = [x_{01}, \ldots, x_{0n}]$, then the goal is to evaluate $\frac{dJ}{d\boldsymbol{q}}$ ($\in \mathbb{R}^{(p+n)\times 1}$) and $\frac{d^2J}{d\boldsymbol{q}^2}$ ($\in \mathbb{R}^{(p+n)\times(p+n)}$).
It should be noted that, since $\boldsymbol{x}$ is a function of $\boldsymbol{x_0}$ and $\boldsymbol{p}$, J can be expressed as $J(\boldsymbol{q})$.
In this work, comparison of the adjoint method to other existing approaches have been done on the basis of the number of scalar integrations required in each algorithm. As more integrations require a higher computational effort, this number (referred to as $N_s^{(method)}$ in the document) has been used to provide an indirect comparison of computational efficiency of each method. $N_s^{(method)}$ is derived in terms of 2 variables: namely the number of states (n) and the number of parameters (p) in the model equation (equation (1)).

II. DIRECT DIFFERENTIATION (DD)

As the name suggests, Direct Differentiation is basically directly taking the derivative of the cost function J with respect to the optimization variables ($\boldsymbol{q}$). The method is expounded on in the following subsections.

A. Gradient

The cost function of interest is $J(\boldsymbol{q}) = \sum_{i=0}^{N} g_i$. After defining the following notations:

$$\underbrace{\frac{d\boldsymbol{a}}{d\boldsymbol{b}}}_{(p\times n)} = \begin{bmatrix} \frac{da_1}{db_1} & \cdots & \frac{da_n}{db_p} \\ \vdots & \ddots & \vdots \\ \frac{da_1}{db_p} & \cdots & \frac{da_n}{db_p} \end{bmatrix}; \underbrace{\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{b}}}_{(p\times n)} = \begin{bmatrix} \frac{\partial a_1}{\partial b_1} & \cdots & \frac{\partial a_n}{\partial b_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial a_1}{\partial b_p} & \cdots & \frac{\partial a_n}{\partial b_p} \end{bmatrix};$$
$$\underbrace{\frac{\partial (.)}{\partial \boldsymbol{a}}}_{n\times 1} = \begin{bmatrix} \frac{\partial (.)}{\partial a_1} & \cdots & \frac{\partial (.)}{\partial a_n} \end{bmatrix}^T \quad (3)$$

where $\boldsymbol{a} = [a_1, \ldots, a_n]^T$, $\boldsymbol{b} = [b_1, \ldots, b_p]^T$ and $(.)$ is a scalar quantity, the derivative of the cost function can be shown to be

$$\frac{dJ}{d\boldsymbol{q}} = \begin{bmatrix} \frac{dJ}{d\boldsymbol{p}} \\ \frac{dJ}{d\boldsymbol{x_0}} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{N} (\frac{\partial g}{\partial p} + \frac{d\boldsymbol{x}}{d\boldsymbol{p}} \frac{\partial g}{\partial \boldsymbol{x}})_i \\ \sum_{i=0}^{N} (\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}} \frac{\partial g}{\partial \boldsymbol{x}})_i \end{bmatrix}. \quad (4)$$

$(.)_i$ represents $(.)$ evaluated at time t_i. Except the sensitivity of the states to the parameters and initial conditions (i.e. $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$ and $\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}}$), all other terms are known since g is known. $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$ and $\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}}$ can be found from the derivatives of the dynamic model equation (Equation 1) with respect to $\boldsymbol{p}$ and $\boldsymbol{x_0}$ respectively. These equations are

$$\frac{d\dot{\boldsymbol{x}}}{d\boldsymbol{q}} = \begin{bmatrix} \frac{d\dot{\boldsymbol{x}}}{d\boldsymbol{p}} \\ \frac{d\dot{\boldsymbol{x}}}{d\boldsymbol{x_0}} \end{bmatrix} = \begin{bmatrix} \frac{d\boldsymbol{x}}{d\boldsymbol{p}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}} \\ \frac{d\boldsymbol{x}}{d\boldsymbol{x_0}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \end{bmatrix}. \quad (5)$$

Integrating equation (5) allows one to determine $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$ and $\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}}$ over time. Once they are known in time, they can be substituted in equation (4) to evaluate the desired gradient. It should be noted that equation (5) is a matrix differential equation which needs $(p + n) \times n$ simultaneous scalar integrations.

B. Tensor-Matrix Operators

Before the derivation of the Hessian via DD is presented, 3 operators are defined which operate on 2^{nd} order tensors (or 3-D matrices) and 2-D matrices. These operators can then be used to express the Hessian in a concise manner.

1) Operator 1: If T is a 3D matrix of dimensions $(a \times b \times c)$ and M is 2-D matrix of dimension $(c \times d)$, then

$$(T^{\rightarrow M})_{i,j,l} = \sum_{k=1}^{c} T_{i,j,k} M_{k,l} \quad (6)$$

where $T^{\rightarrow M} \in \mathbb{R}^{(a\times b\times d)}$. Operator 1 is developed to write the following derivative in short hand.

$$\frac{d}{d\underbrace{\boldsymbol{b}}_{b\times 1}} \left(\underbrace{A(\boldsymbol{b})}_{a\times c} \underbrace{M}_{c\times d} \right) = (\underbrace{\frac{dA(\boldsymbol{b})}{d\boldsymbol{b}}}_{T})^{\rightarrow M} \quad (7)$$

2) Operator 2: If T is a 3D matrix of dimensions $(b \times c \times d)$ and M is 2-D matrix of dimension $(a \times b)$, then

$$(^{M\rightarrow}T)_{i,l,k} = \sum_{j=1}^{b} T_{j,l,k} M_{i,j} \quad (8)$$

where $^{M\rightarrow}T \in \mathbb{R}^{(a\times c\times d)}$. Operator 2 is developed to write the following derivative in short hand.

$$\frac{d}{d\underbrace{\boldsymbol{c}}_{c\times 1}} \left(\underbrace{M}_{a\times b} \underbrace{A(\boldsymbol{c})}_{b\times d} \right) = {}^{M\rightarrow} \underbrace{\frac{dA(\boldsymbol{c})}{d\boldsymbol{c}}}_{T} \quad (9)$$

3) Operator 3: If T is a 3D matrix of dimensions $(a \times b \times c)$ and M is 2-D matrix of dimension $(b \times d)$, then

$$(T \rightarrow M)_{i,l,k} = \sum_{j=1}^{b} T_{i,j,k} M_{j,l} \quad (10)$$

where $T \rightarrow M \in \mathbb{R}^{(a\times d\times c)}$. Operator 3 is developed to write the following derivative in short hand.

$$\frac{d}{d\underbrace{\boldsymbol{d}}_{d\times 1}} \left(\underbrace{A(\boldsymbol{x}(\boldsymbol{d}))}_{a\times c} \right) = \underbrace{\frac{dA(\boldsymbol{x})}{d\boldsymbol{x}}}_{T} \rightarrow \underbrace{\frac{d\boldsymbol{x}}{d\boldsymbol{d}}}_{b\times d} \quad (11)$$

Operators 1 through 3 have been used in the subsequent sections to represent tensor-matrix products.

C. Hessian

Similar to the method of gradient, the Hessian of the cost function via DD can be evaluated by directly differentiating the gradient ($\frac{dJ}{d\boldsymbol{q}}$) with respect to the $\boldsymbol{q}$ vector. Therefore, on differentiating equation (4), we get

$$\frac{d^2J}{d\boldsymbol{q}^2} = \begin{bmatrix} \frac{d^2J}{d\boldsymbol{p}^2} & \frac{d^2J}{d\boldsymbol{p}d\boldsymbol{x_0}} \\ \frac{d^2J}{d\boldsymbol{x_0}d\boldsymbol{p}} & \frac{d^2J}{d\boldsymbol{x_0}^2} \end{bmatrix} \quad (12)$$

where

$$\frac{d^2J}{d\boldsymbol{p}^2} = \sum_{i=0}^{N} (\frac{\partial^2 g}{\partial \boldsymbol{p}^2} + \frac{\partial^2 g}{\partial \boldsymbol{p}\partial \boldsymbol{x}} \frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T + \frac{d\boldsymbol{x}}{d\boldsymbol{p}} \frac{\partial^2 g}{\partial \boldsymbol{x}\partial \boldsymbol{p}} + \frac{d\boldsymbol{x}}{d\boldsymbol{p}} \frac{\partial^2 g}{\partial \boldsymbol{x}^2} \frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T + \frac{d^2\boldsymbol{x}}{d\boldsymbol{p}^2}^{\rightarrow \frac{\partial g}{\partial \boldsymbol{x}}})_i \ , \quad (13)$$

$$\frac{d^2J}{d\boldsymbol{x_0}^2} = \sum_{i=0}^{N} (\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}} \frac{\partial^2 g}{\partial \boldsymbol{x}^2} \frac{d\boldsymbol{x}}{d\boldsymbol{x_0}}^T + \frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}^2}^{\rightarrow \frac{\partial g}{\partial \boldsymbol{x}}})_i \text{ and} \quad (14)$$

$$\frac{d^2J}{d\boldsymbol{x_0}d\boldsymbol{p}} = \sum_{i=0}^{N} (\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}} \frac{\partial^2 g}{\partial \boldsymbol{x}\partial \boldsymbol{p}} + \frac{d\boldsymbol{x}}{d\boldsymbol{x_0}} \frac{\partial^2 g}{\partial \boldsymbol{x}^2} \frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T + \frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}d\boldsymbol{p}}^{\rightarrow \frac{\partial g}{\partial \boldsymbol{x}}})_i. \quad (15)$$

The known quantities ($\frac{\partial^2 g}{\partial \boldsymbol{x}^2}$, $\frac{\partial^2 g}{\partial \boldsymbol{x}\partial \boldsymbol{p}}$, $\frac{\partial^2 g}{\partial \boldsymbol{p}\partial \boldsymbol{x}}$ and $\frac{\partial^2 g}{\partial \boldsymbol{p}^2}$) can be defined in a manner similar to equation (**??**).
The unknown quantities in equations (13) through (15) are the tensors $\frac{d^2\boldsymbol{x}}{d\boldsymbol{p}^2}$, $\frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}d\boldsymbol{p}}$ and $\frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}^2}$. These quantities need to be calculated dynamically from a tensor differential equations

derived by differentiating equation (5) with respect to $\boldsymbol{p}$ and $\boldsymbol{x_0}$. On doing so, we get 3 independent equations

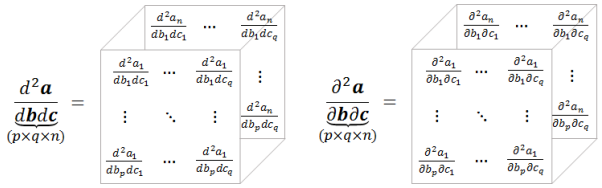
$$\frac{d^2\dot{\boldsymbol{x}}}{d\boldsymbol{p}^2} = \frac{\partial^2 f}{\partial \boldsymbol{p}^2} + \left(\frac{\partial^2 f}{\partial \boldsymbol{p}\partial \boldsymbol{x}} \rightarrow \frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T\right) + \left(^{\frac{d\boldsymbol{x}}{d\boldsymbol{p}}}\rightarrow \frac{\partial^2 f}{\partial \boldsymbol{x}\partial \boldsymbol{p}}\right) + \left[\left(^{\frac{d\boldsymbol{x}}{d\boldsymbol{p}}}\rightarrow \frac{\partial^2 f}{\partial \boldsymbol{x}^2}\right) \rightarrow \frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T\right] + \frac{d^2\boldsymbol{x}}{d\boldsymbol{p}^2}^{\rightarrow \frac{\partial f}{\partial \boldsymbol{x}}}, \quad (16)$$

$$\frac{d^2\dot{\boldsymbol{x}}}{d\boldsymbol{x_0}d\boldsymbol{p}} = \left(^{\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}}}\rightarrow \frac{\partial^2 f}{\partial \boldsymbol{x}\partial \boldsymbol{p}}\right) + \left[\left(^{\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}}}\rightarrow \frac{\partial^2 f}{\partial \boldsymbol{x}^2}\right) \rightarrow \frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T\right] + \frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}^2}^{\rightarrow \frac{\partial f}{\partial \boldsymbol{x}}} \text{ and } \quad (17)$$

$$\frac{d^2\dot{\boldsymbol{x}}}{d\boldsymbol{x_0}^2} = \left[\left(^{\frac{d\boldsymbol{x}}{d\boldsymbol{x_0}}}\rightarrow \frac{\partial^2 f}{\partial \boldsymbol{x}^2}\right) \rightarrow \frac{d\boldsymbol{x}}{d\boldsymbol{x_0}}^T\right] + \frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}d\boldsymbol{p}}^{\rightarrow \frac{\partial f}{\partial \boldsymbol{x}}}. \quad (18)$$

where the tensors can be defined via the notation in Figure 1.

Since equations (16) through (18) are tensor differential



(a) $\frac{d^2\boldsymbol{a}}{d\boldsymbol{b}d\boldsymbol{c}}$ (b) $\frac{\partial^2\boldsymbol{a}}{\partial\boldsymbol{b}\partial\boldsymbol{c}}$

Fig. 1. Visualization of the second derivative tensors where $\boldsymbol{a} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^p$ and $\boldsymbol{c} \in \mathbb{R}^q$

equations of $\frac{d^2\boldsymbol{x}}{d\boldsymbol{p}^2}$, $\frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}d\boldsymbol{p}}$ and $\frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}^2}$, it needs (p^2n), (pn^2) and (n^3) scalar integrations to compute respectively. Once, $\frac{d^2\boldsymbol{x}}{d\boldsymbol{p}^2}$, $\frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}d\boldsymbol{p}}$ and $\frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}^2}$ are known over time, they can be substituted in equation (12) to evaluate the Hessian.

III. ADJOINT METHOD

The DD approach provides a fairly straight forward method to obtaining the gradients and the Hessians. However, in DD, while calculating the gradient ($\frac{dJ}{d\boldsymbol{q}}$) the first step was to determine the sensitivity of the states ($\frac{d\boldsymbol{x}}{d\boldsymbol{q}}$). Similarly, while calculating the Hessian ($\frac{d^2J}{d\boldsymbol{q}^2}$), the second derivative of the states to the variables $\frac{d^2\boldsymbol{x}}{d\boldsymbol{q}^2}$ was needed, both of which are relatively expensive.

To avoid those expensive calculations, an alternative method of computing gradients and Hessians can be devised called the Adjoint Method. It provides an efficient way to determine gradients without having to calculate the sensitivity of the states ($\frac{d\boldsymbol{x}}{d\boldsymbol{q}}$) and a way to determine Hessians without computing the expensive $\frac{d^2\boldsymbol{x}}{d\boldsymbol{q}^2}$; thereby improving the computational efficiency. The following sections elaborate the adjoint method.

A. Gradient

The derivation starts with an augmented cost function of interest (L_1) (of the form of a Lagrangian) as:

$$L_1(\boldsymbol{q}) = \sum_{i=0}^{N} g_i + \sum_{i=0}^{N-1}\left(\int_{t_i^+}^{t_{(i+1)}^-} (\dot{\boldsymbol{x}} - \boldsymbol{f}(\boldsymbol{x},\boldsymbol{p},t))^T \boldsymbol{\lambda} dt\right) \quad (19)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^n$ is a new set of introduced states (also called the co-states and are analogous to Lagrange multipliers). $\boldsymbol{\lambda}$ is assumed to be a discontinuous variable being non-differentiable at time points t_i (i.e. the value of $\boldsymbol{\lambda}$ jumps at the edges of the intervals). This is why the total integral over time has been broken into N integrals over N time intervals. When the state equations are satisfied, the second summation term in the L_1 (equation (19)) becomes 0, making $L_1 = J$. In that case, $\frac{dL_1}{d\boldsymbol{q}} = \frac{dJ}{d\boldsymbol{q}}$ also holds true. Since, the states ($\boldsymbol{x}$) are always determined by solving the state equation, the gradient of the cost function ($\frac{dJ}{d\boldsymbol{q}}$) is always equal to the gradient of the augmented cost function ($\frac{dL_1}{d\boldsymbol{q}}$). Using this property, the gradient of the augmented cost is ultimately evaluated.

An expression for the gradient can be determined by differentiating equation (19) with respect to $\boldsymbol{q}$ to get $\frac{dL_1}{d\boldsymbol{q}} = [\frac{dL_1}{d\boldsymbol{p}} \ \frac{dL_1}{d\boldsymbol{x_0}}]^T$. The development of only $\frac{dL_1}{d\boldsymbol{p}}$ is presented here since deriving $\frac{dL_1}{d\boldsymbol{x_0}}$ is almost identical.

An expression for $\frac{dL_1}{d\boldsymbol{p}}$ is obtained from L_1 as

$$\frac{dL_1}{d\boldsymbol{p}} = \sum_{i=0}^{N}\left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\frac{\partial g}{\partial \boldsymbol{x}} + \frac{\partial g}{\partial \boldsymbol{p}}\right)_i + \sum_{i=0}^{N-1}\left(\int_{t_i^+}^{t_{(i+1)}^-}\left(\frac{d\dot{\boldsymbol{x}}}{d\boldsymbol{p}} - \frac{d\boldsymbol{x}}{d\boldsymbol{p}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\right)\boldsymbol{\lambda} dt\right). \quad (20)$$

Now, using the properties of integration by parts on the term $\frac{d\dot{\boldsymbol{x}}}{d\boldsymbol{p}}$, we get

$$\int_{t_i^+}^{t_{(i+1)}^-} \frac{d\dot{\boldsymbol{x}}}{d\boldsymbol{p}}\boldsymbol{\lambda} dt = \left[\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\boldsymbol{\lambda}\right]_{t_i^+}^{t_{(i+1)}^-} - \int_{t_i^+}^{t_{(i+1)}^-}\left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\dot{\boldsymbol{\lambda}}\right) dt. \quad (21)$$

With the substitution of equation (21), equation (20) simplifies to

$$\frac{dL_1}{d\boldsymbol{p}} = \sum_{i=0}^{N}\left(\frac{\partial g}{\partial \boldsymbol{p}}\right)_i + \sum_{i=1}^{N-1}\left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_i\left[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_i + \boldsymbol{\lambda}_{i^-} - \boldsymbol{\lambda}_{i^+}\right] + \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_0\left[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_0 - \boldsymbol{\lambda}_{0^+}\right] + \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_N\left[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_N + \boldsymbol{\lambda}_{N^-}\right] + \sum_{i=0}^{N-1}\left(\int_{t_i^+}^{t_{(i+1)}^-}\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\left(-\dot{\boldsymbol{\lambda}} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{\lambda}\right) dt\right) + \sum_{i=0}^{N-1}\left(\int_{t_i^+}^{t_{(i+1)}^-}\left(-\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\boldsymbol{\lambda}\right) dt\right). \quad (22)$$

It should be noted that $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$ and $\frac{\partial g}{\partial \boldsymbol{x}}$ are continuous functions, i.e. $(\frac{d\boldsymbol{x}}{d\boldsymbol{p}})_i = (\frac{d\boldsymbol{x}}{d\boldsymbol{p}})_{i^-} = (\frac{d\boldsymbol{x}}{d\boldsymbol{p}})_{i^+}$ and $(\frac{\partial g}{\partial \boldsymbol{x}})_i = (\frac{\partial g}{\partial \boldsymbol{x}})_{i^-} = (\frac{\partial g}{\partial \boldsymbol{x}})_{i^+}$. Equation (22) is still dependent on $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$. However, the whole purpose of the adjoint method was to avoid calculating

$\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$. To make this possible, all the terms that are associated with $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$ somehow needs to be eliminated. The first step is to solve the co-state differential equation

$$-\dot{\boldsymbol{\lambda}} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{\lambda} = 0. \tag{23}$$

Once equation (23) is solved and $\boldsymbol{\lambda}$ is known over time, equation (22) simplifies to

$$\frac{dL_1}{d\boldsymbol{p}} = \sum_{i=0}^{N}\left(\frac{\partial g}{\partial \boldsymbol{p}}\right)_i + \sum_{i=1}^{N-1}\left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_i\Bigg[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_i + \boldsymbol{\lambda}_{i^-} - \boldsymbol{\lambda}_{i^+}\Bigg] + \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_0\left[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_0 - \boldsymbol{\lambda}_{0^+}\right] + \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_N\Bigg[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_N + \boldsymbol{\lambda}_{N^-}\Bigg] + \sum_{i=0}^{N-1}\left(\int_{t_i^+}^{t_{(i+1)}^-}\left(-\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\boldsymbol{\lambda}\right)dt\right). \tag{24}$$

Equation (23) needs to be solved separately over each time interval (since $\boldsymbol{\lambda}$ is discontinuous at the edges of the intervals). A set of boundary conditions for each of those intervals are also necessary. A smart selection of these boundary conditions can be used to eliminate some of the other terms containing $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$ in equation (24). Assuming $\boldsymbol{\lambda}_{N^-} = -(\frac{\partial g}{\partial \boldsymbol{x}})_N$, equation (23) can be solved by integrating it backward in time from t_{N^-} to t_{N-1^+}. This eliminates the need for the term $(\frac{d\boldsymbol{x}}{d\boldsymbol{p}})_N$ from equation (24) since it now multiplies 0. The next step is to evaluate $\boldsymbol{\lambda}$ in the time intervals between t_{i+1^-} and t_{i^+}. This is again done by solving equation (23) by integrating it back in the respective time intervals. However, this time, the boundary conditions $\boldsymbol{\lambda}_{i+1^-}$ for each interval is calculated by solving the algebraic equation

$$\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_i + \boldsymbol{\lambda}_{i^-} - \boldsymbol{\lambda}_{i^+} = 0 \tag{25}$$

where $\boldsymbol{\lambda}_{i^+}$ is known: as it is the terminal $\boldsymbol{\lambda}$ from the solution of equation (23) in the previous time interval. Such selection of boundary conditions removes the need for evaulating $(\frac{d\boldsymbol{x}}{d\boldsymbol{p}})_i$. In summary, $\boldsymbol{\lambda}$ needs to be solved separately over each time interval (using equation (23)). At the end of each integration, the boundary value of $\boldsymbol{\lambda}$ for the next integration (or time interval) is determined (using equation (25)).

Solving equation (25) causes the second term in equation (24) to be 0 and simplifies it to

$$\frac{dJ}{d\boldsymbol{p}} = \frac{dL_1}{d\boldsymbol{p}} = \sum_{i=0}^{N}\left(\frac{\partial g}{\partial \boldsymbol{p}}\right)_i + \sum_{i=0}^{N-1}\left(\int_{t_i^+}^{t_{(i+1)}^-}\left(-\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\boldsymbol{\lambda}\right)dt\right). \tag{26}$$

since the initial value of the states are independent of the parameters, i.e., $(\frac{d\boldsymbol{x}}{d\boldsymbol{p}})_0 = 0$.

Equation (23) is also called the adjoint equation (and hence the name: Adjoint method). Equation (26) represents the final equation that needs to be evaluated to calculate the gradient. Similarly, a gradient equation for the initial conditions can also be obtained. It can be shown that

$$\frac{dJ}{d\boldsymbol{x_0}} = \frac{dL_1}{d\boldsymbol{x_0}} = \left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_0 - \boldsymbol{\lambda}_{0^+} \tag{27}$$

where the co-states $\boldsymbol{\lambda}$ are solved in an identical fashion.

B. Hessian

A huge drawback of the DD method for calculating the Hessian was the need to calculate $\frac{d^2\boldsymbol{x}}{d\boldsymbol{p}^2}$, $\frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0}d\boldsymbol{p}}$ and $\frac{d^2\boldsymbol{x}}{d\boldsymbol{x_0^2}}$. To solve for them over time, one needs to solve 3 tensor differential equations involving $(p^2n + n^2p + n^3)$ scalar integrations; which is computationally expensive. The adjoint method once again allows us to bypass those integrations and evaluate the Hessian in an alternate manner, thus improving the computational efficiency.

Once again, only the development of $\frac{d^2J}{d\boldsymbol{p}^2}$ is presented since the other parts of the Hessian ($\frac{d^2J}{d\boldsymbol{x_0^2}}$ and $\frac{d^2J}{d\boldsymbol{x_0}d\boldsymbol{p}}$) can be derived in the same way. Similar to the derivation of the gradient, first an augmented cost function (L_2) is written as

$$\boldsymbol{L_2} = \frac{dL_1}{d\boldsymbol{p}} + \sum_{i=0}^{N-1}\left(\int_{t_i^+}^{t_{(i+1)}^-}\eta(\dot{\boldsymbol{x}} - \boldsymbol{f})dt + \int_{t_i^+}^{t_{(i+1)}^-}\gamma\left(-\dot{\boldsymbol{\lambda}} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{\lambda}\right)dt\right), \tag{28}$$

where $\eta \in \mathbb{R}^{(p\times n)}$ and $\gamma \in \mathbb{R}^{(p\times n)}$ are Lagrangian multipliers.

With a similar argument as the one used for L_1, when the state and the co-state equations are satisfied, the integral terms in equation (28) become 0 making $\boldsymbol{L_2} = \frac{dL_1}{d\boldsymbol{p}} = \frac{dJ}{d\boldsymbol{p}}$. In that case, $\frac{dL_2}{d\boldsymbol{p}} = \frac{d^2J}{d\boldsymbol{p}^2}$ also holds true. Since, the states ($\boldsymbol{x}$) are always determined by solving the state equation, and the co-states $\boldsymbol{\lambda}$ are also always determined using the co-state dynamic equation: the Hessian of the cost function ($\frac{d^2J}{d\boldsymbol{p}^2}$) is always equal to the first derivative of the augmented cost function ($\frac{dL_2}{d\boldsymbol{p}}$). Using this property, the derivative of the augmented cost L_2 is ultimately evaluated.

Now, substituting $\frac{dL_1}{d\boldsymbol{p}}$ from equation (24), we get

$$\boldsymbol{L_2} = \sum_{i=0}^{N}\left(\frac{\partial g}{\partial \boldsymbol{p}}\right)_i + \sum_{i=1}^{N-1}\left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_i\Bigg[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_i + \boldsymbol{\lambda}_{i^-} - \boldsymbol{\lambda}_{i^+}\Bigg] + \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_0\left[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_0 - \boldsymbol{\lambda}_{0^+}\right] + \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_N\Bigg[\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)_N + \boldsymbol{\lambda}_{N^-}\Bigg] + \sum_{i=0}^{N-1}\left(\int_{t_i^+}^{t_{(i+1)}^-}\left(-\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\boldsymbol{\lambda}\right)dt + \int_{t_i^+}^{t_{(i+1)}^-}\eta(\dot{\boldsymbol{x}} - \boldsymbol{f})dt + \int_{t_i^+}^{t_{(i+1)}^-}\gamma\left(-\dot{\boldsymbol{\lambda}} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{\lambda}\right)dt\right). \tag{29}$$

To evaluate $\frac{dL_2}{d\boldsymbol{p}}$, equation (29) is first differentiated with respect to $\boldsymbol{p}$. Then, terms which are inherently 0 (such as $(\frac{d^2\boldsymbol{x}}{d\boldsymbol{p}^2})_0$ and $(\frac{d\boldsymbol{x}}{d\boldsymbol{p}})_0$) are eliminated. Furthermore, since the terms multiplying the boundary conditions for $\boldsymbol{\lambda}$ are satisfied when evaluating $\boldsymbol{\lambda}$, we can also eliminate them. Finally, using the properties of integration by parts on the resultant,

we get

$$\frac{dL_2}{d\boldsymbol{p}} = \sum_{i=0}^{N-1}\Bigg[\int_{t_i^+}^{t_{(i+1)}^-}\Bigg(-\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{p}^2}^{\rightarrow\boldsymbol{\lambda}} - \gamma\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}\partial \boldsymbol{p}}^{\rightarrow\boldsymbol{\lambda}} - \eta\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}^T\Bigg)dt + \int_{t_i^+}^{t_{(i+1)}^-}\Bigg(-\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{p}\partial \boldsymbol{x}}^{\rightarrow\boldsymbol{\lambda}} - \gamma\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}^2}^{\rightarrow\boldsymbol{\lambda}} - \eta\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}^T - \dot{\eta}\Bigg)\frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T dt + \int_{t_i^+}^{t_{(i+1)}^-}\Bigg(-\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}} - \gamma\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \dot{\gamma}\Bigg)\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}^T dt\Bigg] + \sum_{i=0}^{N}\left(\frac{\partial^2 g}{\partial \boldsymbol{p}^2}\right)_i + \sum_{i=1}^{N}\Bigg[\Bigg(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\frac{\partial^2 g}{\partial \boldsymbol{x}\partial \boldsymbol{p}} + \frac{d\boldsymbol{x}}{d\boldsymbol{p}}\frac{\partial^2 g}{\partial \boldsymbol{x}^2}\frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T + \frac{\partial^2 g}{\partial \boldsymbol{p}\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T\Bigg)_i + \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_i \frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{i^-}^T - \gamma_i\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{i^-}^T\Bigg] + \sum_{i=1}^{N-1}\Bigg[\gamma_i\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{i^+}^T - \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_i\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{i^+}^T\Bigg] + \gamma_0\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{0^+}^T - \sum_{i=1}^{N-1}(\eta_{i^-} - \eta_{i^+})\left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T\right)_i + \eta\frac{d\boldsymbol{x}}{d\boldsymbol{p}}_{N^-}^T - \eta\frac{d\boldsymbol{x}}{d\boldsymbol{p}}_{0^+}^T. \quad (30)$$

Equation (30) is the expression to calculate the Hessian. However, this expression is still dependent on some terms which are unknown to us (for example: $\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}$). To solve this problem, all terms associated with them need to be eliminated. An approach similar to the adjoint equation during gradient calculation is exercised.

The third integral term of equation (30) is eliminated by solving the differential equation in γ

$$-\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}} - \gamma\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \dot{\gamma} = 0, \text{ with } \gamma(0) = 0. \quad (31)$$

The second integral term of equation (30) is eliminated by solving the differential equation in η

$$-\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{p}\partial \boldsymbol{x}}^{\rightarrow\boldsymbol{\lambda}} - \eta\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}^T - \dot{\eta} - \gamma\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}^2}^{\rightarrow\boldsymbol{\lambda}} = 0 \quad (32)$$

with a terminal boundary condition $\eta(T) = 0$ and other interval boundary conditions derived from $\eta_{i^-} - \eta_{i^+} = 0$. The method to solve for η is similar to that of $\boldsymbol{\lambda}$ (i.e. requires separate integrations for each interval with boundary conditions calculated for each of those intervals separately). Furthermore, recognising that $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}(0) = 0$, equation (30) simplifies to

$$\frac{dL_2}{d\boldsymbol{p}} = \sum_{i=0}^{N-1}\Bigg[\int_{t_i^+}^{t_{(i+1)}^-}\Bigg(-\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{p}^2}^{\rightarrow\boldsymbol{\lambda}} - \gamma\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}\partial \boldsymbol{p}}^{\rightarrow\boldsymbol{\lambda}} - \eta\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}^T\Bigg)dt\Bigg] + \sum_{i=0}^{N}\left(\frac{\partial^2 g}{\partial \boldsymbol{p}^2}\right)_i + \sum_{i=1}^{N}\Bigg[\Bigg(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\frac{\partial^2 g}{\partial \boldsymbol{x}\partial \boldsymbol{p}} + \frac{d\boldsymbol{x}}{d\boldsymbol{p}}\frac{\partial^2 g}{\partial \boldsymbol{x}^2}\frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T + \frac{\partial^2 g}{\partial \boldsymbol{p}\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{d\boldsymbol{p}}^T\Bigg)_i + \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_i \frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{i^-}^T - \gamma_i\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{i^-}^T\Bigg] + \sum_{i=1}^{N-1}\Bigg[\gamma_i\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{i^+}^T - \left(\frac{d\boldsymbol{x}}{d\boldsymbol{p}}\right)_i\frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}_{i^+}^T\Bigg]. \quad (33)$$

At this point, it seems that the intermittent values of $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$ are needed to evaluate equation (33). However, on observing the problem carefully, one can see that the γ equation (equation (31)) is in fact essentially the same as the $\frac{d\boldsymbol{x}}{d\boldsymbol{p}}$ equation (equation (5)). Therefore, it turns out that when calculating the Hessian of the cost function, one has to evaluate the sensitivity of the states to the parameters over time: whether directly as in DD or indirectly as in the Adjoint method. Since $\frac{d\boldsymbol{x}}{d\boldsymbol{p}} = \gamma$, it can be substituted in equation (33).

Hence, the final value of the Hessian is given by

$$\frac{d^2 J}{d\boldsymbol{p}^2} = \frac{dL_2}{d\boldsymbol{p}} = \sum_{i=0}^{N-1}\Bigg[\int_{t_i^+}^{t_{(i+1)}^-}\Bigg(-\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{p}^2}^{\rightarrow\boldsymbol{\lambda}} - \gamma\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}\partial \boldsymbol{p}}^{\rightarrow\boldsymbol{\lambda}} - \eta\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}^T\Bigg)dt\Bigg] + \sum_{i=0}^{N}\left(\frac{\partial^2 g}{\partial \boldsymbol{p}^2}\right)_i + \sum_{i=1}^{N}\Bigg[\Bigg(\gamma\frac{\partial^2 g}{\partial \boldsymbol{x}\partial \boldsymbol{p}} + \gamma\frac{\partial^2 g}{\partial \boldsymbol{x}^2}\gamma^T + \frac{\partial^2 g}{\partial \boldsymbol{p}\partial \boldsymbol{x}}\gamma^T\Bigg)_i. \quad (34)$$

Similarly, it can be shown that

$$\frac{d^2 J}{d\boldsymbol{x}_{\boldsymbol{0}}^2} = \sum_{i=0}^{N}\left(\gamma_2\frac{\partial^2 g}{\partial \boldsymbol{x}^2}\gamma_2^T\right)_i - (\eta_2)_{0^+} \quad (35)$$

where $-\gamma_2\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \dot{\gamma_2} = 0$, with $\gamma_2(0) = I$ and $-\eta_2\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}^T - \dot{\eta_2} - \gamma_2\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}^2}^{\rightarrow\boldsymbol{\lambda}} = 0$, with $\eta_2(T) = 0$ & $(\eta_2)_{i^-} - (\eta_2)_{i^+} = 0$. It can also be shown that

$$\frac{d^2 J}{d\boldsymbol{x}_{\boldsymbol{0}} d\boldsymbol{p}} = \sum_{i=0}^{N-1}\Bigg[\int_{t_i^+}^{t_{(i+1)}^-}\Bigg(-\gamma_2\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}\partial \boldsymbol{p}}^{\rightarrow\boldsymbol{\lambda}} - \eta_2\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}^T\Bigg)dt\Bigg] + \sum_{i=0}^{N}\left(\gamma_2\frac{\partial^2 g}{\partial \boldsymbol{x}^2}\gamma^T + \gamma_2\frac{\partial^2 g}{\partial \boldsymbol{x}\partial \boldsymbol{p}}\right)_i. \quad (36)$$

This concludes the presentation of all the results for the gradient and the Hessian calculations using the Adjoint method. The next section presents a brief summary of the computational efficiencies of each method.

IV. COMPUTATIONAL EFFICIENCY

Considering that the cost function is a function of states from a dynamic system, a good measure of the expense is to compare the number of scalar integrations needed in each method. The comparison has been made for (1) when only the gradient is calculated and (2) when both the gradient and Hessian are calculated. Table. I compares the computation cost of the FD, DD and Adjoint approach clearly illustrating the benefit of the Adjoint approach. The

TABLE I
COMPARISON OF COMPUTATIONAL EXPENSE

Algorithm	Gradient	Gradient + Hessian
$N_s^{(FD)}$	$2np + 2n^2$	$n^3 + 2n^2p + p^2n + n^2 + pn + n$
$N_s^{(DD)}$	$n + pn + n^2$	$n^2 + n + pn + p^2n + n^3 + n^2p$
$N_s^{(Adjoint)}$	$2n + p$	$p + p^2 + 2n + 2n^2 + 3pn$

computational advantage of the Adjoint method is evident from the table which suggests that the growth of N_s is given by 3^{rd} order polynomials for FD as well as DD while it is only a 2^{nd} order polynomial for the Adjoint.

V. SYSTEM IDENTIFICATION PROBLEM

The Lorenz-63 model given by equations:

$$\dot{x}_1 = -p_1(x_1 - x_2) \tag{37}$$

$$\dot{x}_2 = x_1(p_2 - x_3) - x_2 \tag{38}$$

$$\dot{x}_3 = x_1 x_2 - p_3 x_3. \tag{39}$$

with unknown parameters and initial conditions, is used as an example to illustrate the proposed technique. This particular system is chosen since it is known to be chaotic. It is highly sensitive to initial conditions and parameters making it an ideal choice to illustrate the accuracy and the feasibility of the Adjoint method.

It is assumed that all the parameters p_1, p_2, p_3 and the initial conditions for x_2 & x_3 are unknown. For system identification, it is also assumed that a time series of observations $\hat{x}_1, \hat{x}_2, \hat{x}_3$, (at intervals of $\Delta t = 0.01$ up to $t = 1$) is available for x_1, x_2 and x_3. The time series is generated from the following values: $p_1 = 10, p_2 = 60, p_3 = 8/3, x_1(0) = 20, x_2(0) = 25$ and $x_3(0) = 30$. The intention is to estimate $\boldsymbol{q} = [p_1, p_2, p_3, x_2(0), x_3(0)]'$ correctly. A similar structure for the Lorenz system identification problem can also be found in [7].

To identify the optimal $\boldsymbol{q}$, an optimization problem is posed with a cost function

$$J = \sum_{i=0}^{N} \sum_{j=1}^{3} (x_j(t_i) - \hat{x}_j(t_i))^2. \tag{40}$$

subject to equations (37) through (39).

The problem is solved iteratively. The initial guesses are chosen to be $\hat{p}_1 = 20, \hat{p}_2 = 75, \hat{p}_3 = 10, \hat{x}_2(0) = 10$ and $\hat{x}_3(0) = 15$ for all the methods.

The gradients and the Hessians of the cost function J with respect to $\boldsymbol{q}$ are evaluated at every iteration. These quantities are then used to employ a gradient based algorithm in Matlab to converge to a solution. This is done for each of the methods discussed in the document under identical optimization environments. Figure 2 shows a comparative plot of convergence between the different techniques. The only difference while simulating them was the source of gradients and Hessians that were fed to the optimizer.

Four distinct simulations were made for the FD method with each simulation having a distinct step size to calculate the gradients and Hessians. The step sizes have been listed in Figure 2. Curves corresponding to step sizes 10^{-5} and 10^{-7} show comparatively slow convergences. FD with a step size of 10^{-6} performs very well while FD with a step size of 10^{-8} fails to converge. These results illustrate the variability in the accuracy of FD. Since it is impossible to know the magnitude of the optimal step size beforehand, a method independent of step size is motivated. The DD and

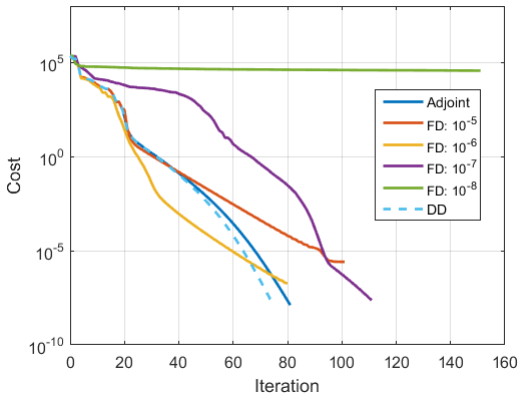


Fig. 2. Convergence of Different Algorithms

the Adjoint method have comparable performance although the DD proves to be the best algorithm in this example having converged the quickest with the lowest terminal cost. However, it must be noted that the Adjoint approach is far less expensive as compared to FD as well as DD.

VI. CONCLUSION

Stimulated by the tradeoff between truncation error and numerical accuracy in finite difference estimates of gradients and Hessians, this paper presents a detailed development of the adjoint sensitivity approach for the determination of the exact Hessian for system identification problems where the measurements are available at discrete times. The proposed approach is illustrated on the Lorenz 63 problem.

ACKNOWLEDGMENT

This material is based upon work supported through National Science Foundation (NSF) under Award No. CMMI-1537210. All results and opinions expressed in this article are those of the authors and do not reflect opinions of NSF.

REFERENCES

[1] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[2] R. L. Raffard, K. Amonlirdviman, J. D. Axelrod, and C. J. Tomlin, "Parameter identification via the adjoint method: Application to protein regulatory networks," *IFAC Proceedings Volumes*, vol. 39, no. 2, pp. 475–482, 2006.

[3] A. Sandu, D. N. Daescu, and G. R. Carmichael, "Direct and adjoint sensitivity analysis of chemical kinetic systems with kpp: Part itheory and software tools," *Atmospheric Environment*, vol. 37, no. 36, pp. 5083–5096, 2003.

[4] M. Shichitake and M. Kawahara, "Optimal control applied to water flow using second order adjoint method," *International Journal of Computational Fluid Dynamics*, vol. 22, no. 5, pp. 351–365, 2008.

[5] S. Liu and T. R. Bewley, "Adjoint-based system identification and feedforward control optimization in automotive powertrain subsystems," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3. IEEE, 2003, pp. 2566–2571.

[6] D. A. Tortorelli and P. Michaleris, "Design sensitivity analysis: overview and review," *Inverse problems in Engineering*, vol. 1, no. 1, pp. 71–105, 1994.

[7] F. Lu, D. Xu, and G. Wen, "Estimation of initial conditions and parameters of a chaotic evolution process from a short time series," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 14, no. 4, pp. 1050–1055, 2004.