Discrete Adjoint Sensitivity Analysis of Hybrid Dynamical Systems

Hong Zhang, Shrirang Abhyankar, *Member, IEEE,* Emil Constantinescu, Mihai Anitescu

***Abstract*—Sensitivity analysis is an important tool for describing power system dynamic behavior in response to parameter variations. It is a central component in preventive and corrective control applications. The existing approaches for sensitivity calculations, namely, finite-difference and forward sensitivity analysis, require a computational effort that increases linearly with the number of sensitivity parameters. In this work, we investigate, implement, and test a discrete adjoint sensitivity approach whose computational effort is effectively independent of the number of sensitivity parameters. The proposed approach is highly efficient for calculating sensitivities of larger systems and is consistent, within machine precision, with the function whose sensitivity we are seeking. This is an essential feature for use in optimization applications. Moreover, our approach includes a consistent treatment of systems with switching, such as DC exciters, by deriving and implementing the adjoint jump conditions that arise from state and time-dependent discontinuities. The accuracy and the computational efficiency of the proposed approach are demonstrated in comparison with the forward sensitivity analysis approach.**

***Index Terms*—Adjoint sensitivity, discrete sensitivity, power system dynamics, transient stability, trajectory sensitivity analysis**

I. Introduction

DYNAMIC security is a concern for system planning and operation experts because significantly higher penetration of renewable energy resources, most of which are electronically coupled to the grid, is expected in the future. This situation presents new technical challenges, particularly the increased dynamic content and reduction of system inertia through the displacement of conventional generation resources during light load periods. Thus, ensuring dynamic security along with optimal and secure steady-state operation is an important emerging problem. To this end, utilities typically design preventive or corrective actions based on a set of directives. For instance, a corrective action directive may prescribe changing the dispatch of a specific set of generators to alleviate overload problems caused by a specific contingency. These directives, based on expert operational judgement and accumulated knowledge, may not be optimal. Moreover, they may even not be secure for the new dynamics of higher renewable energy usage.

This material was based upon and approved by the U.S. Dept. of Energy, Office of Science, Advanced Scientific Computing Research program, under Contract DE-AC02-06CH11357.

Shrirang Abhyankar is with the Energy Systems Division, Argonne National Laboratory.

Hong Zhang, Emil Constantinescu, and Mihai Anitescu are with the Mathematics and Computer Science Division, Argonne National Laboratory.

Optimal and secure preventive and corrective control actions as well as suitable placement of controllers have been extensively studied by power system researchers. The central component in these studies is the calculation of first-order sensitivities of the power system dynamics trajectories with respect to the control parameters. Hiskens and Pai [1] established the theory of trajectory sensitivity analysis (TSA) for hybrid systems modeled by a differential-algebraic-discrete structure, and they developed jump conditions for the sensitivities at discrete events. Subsequently, TSA has been used in numerous applications: stability assessment and real-time emergency control [2]; determination of effectiveness of preventive generation rescheduling and shunt/series compensation in improving transient stability [3]; preventive control of voltage instability through shunt compensation and generation rescheduling [4]; assessment of best location of series-connected controllers to enhance power system transient stability [5]; suitable placement of series compensators for enhancing both transient and small-signal stability [6]; design of a fixed shunt capacitive compensator to maintain the first swing stability of micro grids [7]; dynamic security constrained rescheduling under contingencies [8]; reduction of composite load model parameters identified from field measurement [9]; identification of parameters in nonlinear dynamical models that can be reliably estimated from disturbance measurements [10]; estimation of critical parameters such as clearing time and mechanical input power [11]; transient stability assessment of power systems containing series and shunt FACTS devices [12], [13]; reduction of computational burden of model predictive control method for load shedding [14]; VAr planning in large power system heavily stressed by voltage collapse [15]; parameters design of power system stabilizers [16], [17]; and the study of parameter uncertainty in system behavior [18], [19]. An overview of the applications of TSA in the analysis of large disturbance dynamic behavior of power systems is given in [20].

The idea of sensitivity analysis is not new. For general sensitivity calculations, two approaches, continuous and discrete, have been widely used in many different fields, as shown in Fig. 1. In continuous methods, sensitivity equations are derived directly from the model equations and can be theoretically solved with integration methods and time steps different from those used for the model equations. Discrete methods, on the other hand, are based on the discretized equations, so the propagation scheme and time steps are completed determined by the simulation code. Furthermore, both these approaches have

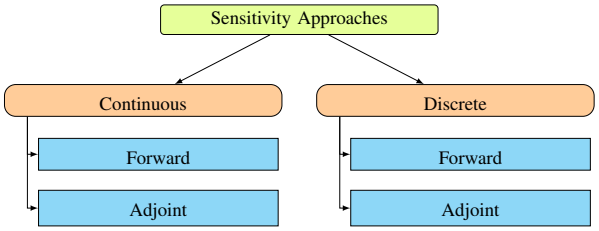


Fig. 1. Taxonomy of approaches for sensitivity calculation.

TABLE I
COMPARISON OF DISCRETE FORWARD AND ADJOINT METHOD

	Forward	Adjoint
Best to use when	$n_p \ll n_c$	$n_p \gg n_c$
Computational complexity	O(n_p)	O(n_c)
Checkpointing	No	Yes
Implementation Difficulty	Medium	High
Accuracy	High	High

n_p:number of parameters n_c:number of cost functions

two variations: forward and adjoint mode. The forward mode calculates the sensitivities by integrating a set of sensitivity equations forward in time, while for the discrete mode the sensitivity equations need a backward-in-time integration. An interesting observation is that the continuous forward approach can be equivalent to the discrete approach if using the same choices of time integration methods and time steps; however, this is not the case for continuous adjoint and discrete adjoint even if the same time integration methods and the same time steps are applied to both. Table I summarizes the comparison between the discrete forward and adjoint approaches.

The TSA approach commonly used in power systems falls into the forward approach category with the objective functions defined to be entries in the system state vector. The major drawback of TSA, just like other forward approaches, is that the computational cost of the forward approach grows linearly with the number of sensitivity parameters. By contrast, the cost of the adjoint approach is effectively constant with respect to the number of sensitivity parameters and grows linearly with the number of objective functions [21]. Therefore, the adjoint approach can be much more efficient than the forward approach when calculating the gradients of a few objective functions with respect to many parameters, a common occurrence in power system dynamics and control. In order to mitigate the performance issue of TSA for dealing with many parameters, embarrassingly parallel computing has been utilized in [22] and [23] so that each parameter sensitivity is computed in parallel. This strategy yields limited scalability, however, since the maximum number of concurrent computational threads will be limited by the number of parameters, and therefore the performance is impacted when the number of parameters and the computational resources are mismatched. This is not the case in the adjoint approach because the sensitivities for all parameters are computed simultaneously.

So far, few efforts have been made to explore the application of the adjoint approach in power systems, probably because of the following reasons:

1) Derivation of adjoints can be complicated depending on the complexity of the dynamic models.
2) Accounting for switching actions in hybrid systems in sensitivity analysis poses both theoretical challenges and implementation challenges.
3) Efficient implementation of adjoint calculations requires development of checkpointing schemes and a proper method to solve linear systems involving Jacobian transpose when an implicit time integration method is used. Checkpointing is a scheme for dramatically reducing the memory requirements of the adjoint method, while moderately increasing the computational cost through recomputations of the trajectory [21].

A noteworthy work along this line is [24], which has applied a continuous adjoint equation method to evaluate the gradient of a stability metric for optimal power flow and demonstrated significant improvement in efficiency. Not addressed, however, were the state-dependent nature of the switching conditions and, as a result, the jump conditions of sensitivity variables that are characteristic of hybrid systems such as DC exciters.

In this work we particularly investigate discrete adjoint sensitivity approaches because the sensitivities computed by discrete adjoint methods equal the derivatives of the function applied to the discretized dynamical system, up to the order of the machine precision. This is not the case for sensitivity computations that use numerical integration of continuous adjoints [25] and may lead to difficulties in convergence if the gradients are used for solving optimization problems [26]. The contributions of this work are as follows.

1) We design, describe, and analyze a workflow to compute discrete adjoints for single-step multistage methods (such as theta methods and Runge-Kutta methods).
2) We implement this workflow in the widely used open-source library Portable, Extensible Toolkit for Scientific Computing (PETSc) [27]. This makes our approach available to the research community for a large class of numerical integration schemes.
3) We extend the discrete adjoint sensitivity analysis of [28] to hybrid systems with state-dependent jumps.
4) We show the potential of our approach and the software for real applications by validating the accuracy of our approach and the expected behavior of adjoint differentiation [21] on several test cases that include 9- and 118-bus dynamics with DC exciters, where the state-dependent switching dynamics of the latter demonstrates the correctness of our jump conditions.

The paper is organized as follows. The power system dynamic model and its numerical solution are discussed in Section II. The formulation of the discrete adjoint method, along with the sensitivity equations and handling of state-dependent jumps, is proposed in Section III. Section IV presents the accuracy and computational efficiency of the different approaches on several test systems. In Section V we summarize our conclusions and briefly discuss future work.

II. Power system dynamic equations

To carry out the objectives described in Section I, we present a hybrid system abstraction of the target dynamical systems. Such a framework is useful for solving both forward problems and inverse problems [29]. We assume that the continuous dynamics is governed by systems of parameter-dependent differential-algebraic equations (DAEs) and that the discrete events are reflected by a jumping mechanism between those systems. This results in a piecewise smooth dynamical systems For an initial value problem, the system visits the smooth pieces in succession, with the states at the entrance in a smooth piece depending on the states at the exit of the previous one. Mathematically this can be described by

$$\dot{x}^{(i)} = f^{(i)}(x, y; p) \tag{1}$$
$$\gamma^{(i)}(x^{(i)}, y^{(i)}; p) = 0 \tag{2}$$
$$0 = g^{(i)}(x, y; p), \tag{3}$$

where $x \in \mathbb{R}^{n_x}$ are the dynamic state variables such as machine angles and velocities, $y \in \mathbb{R}^{n_y}$ are the algebraic variables such as load bus voltage magnitudes and angles, and $p \in \mathbb{R}^{n_p}$ are the system parameters such as line reactances, generator mechanical input power, and fault clearing time. An event is triggered when the stage-dependent condition (2) is satisfied. The equations change at that point, resulting in discontinuities in the state/algebraic variables. The superscript (i) identifies the different sets of equations modeling the events. Initial conditions are given by

$$x(t_0) = \mathcal{I}_{x_0}(p), \quad y(t_0) = \mathcal{I}_{y_0}(p), \tag{4}$$

where we assume that we start in the interior of the smooth piece (0) and thus x_0 and y_0 must satisfy the algebraic constraints for that piece:

$$g(x_0, y_0; p) = 0. \tag{5}$$

We employ the usual assumption that $g_y^{(i)}$ is nonsingular along the trajectories, so each set of equations is a semi-explicit index-1 DAE system [30]. We start with the numerical solution and discrete sensitivity analysis for a single DAE system:

$$\dot{x} = f(x, y; p) \tag{6a}$$
$$0 = g(x, y; p), \tag{6b}$$

and we then extend the approaches to the hybrid cases. The DAE system (6) can be cast into a general form

$$\mathcal{M}\dot{X} = F(X; p), \tag{7}$$

where

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, \; F = \begin{bmatrix} f \\ g \end{bmatrix}, \; \mathcal{M} = \begin{bmatrix} \mathbf{I}_{n_x \times n_x} & \\ & \mathbf{0}_{n_y \times n_y} \end{bmatrix}.$$

To solve (7), we can directly apply, for example, theta methods:

$$\begin{aligned} \mathcal{M}X_{n+1} = \mathcal{M}X_n &+ h_n(1-\theta)F(X_n; p) \\ &+ h_n\theta F(X_{n+1}; p), \; n = 0, \ldots, N-1. \end{aligned} \tag{8}$$

As special cases, the methods with $\theta = 1$ and $\theta = 0.5$ give backward Euler and Crank-Nicolson (also known as trapezoidal) methods, respectively.

III. Discrete Adjoint Sensitivity Calculation

For deriving the discrete adjoint workflow, we assume that system (7) is integrated with a one-step method

$$X_{n+1} = \mathcal{N}_n(X_n), \quad n = 0, \ldots, N-1, \quad X_0 = \mathcal{I}, \tag{9}$$

where $\mathcal{I}$ are the initial values and the solution at the end of the simulation is given by X_N. We aim to efficiently compute sensitivities of an objective function (sometimes called response function) with respect to initial values or system parameters. A general form of the objective function, involving a terminal and trajectory term, can be written as

$$\Psi = \psi(X(t_F); p) + \int_{t_0}^{t_F} r(t, X; p)dt. \tag{10}$$

Note that so-called trajectory sensitivity [20], known as the derivative of the final solution (corresponding to $\Psi = X(t_F)$ in (10)) with respect to initial values, is just a special case of what we are considering. For notational brevity, in the rest of the discussion we drop the argument p in ψ, r, F.

Continuous sensitivity approaches that are then discretized do not take into consideration that the objective function is approximated by numerical values, not the exact values. For example, the exact function $\psi(X(t_F))$ is approximated by $\psi(X_N)$, and the numerical approximation of the continuous gradient with respect to the sensitivity parameters is the gradient of $\psi(X_N)$ only up to numerical integration tolerance. Discrete approaches, on the other hand, compute *algebraical derivatives* of $\psi(X_N)$ and thus have an error on the order of machine precision. For low- and moderate-precision integration (which, in real-time contexts, for example, would be necessary) the latter error is much smaller. This is essential in optimization applications, where we would like to make sure that we have very accurate descent directions for $\psi(X_N)$.

We first consider a simple case in which we compute sensitivities of $\Psi^{\text{approx}} = \psi(X_N)$ to initial values. Here, Ψ^{approx} is the numerical approximation to the target objective function. We use the Lagrange multipliers $\lambda_0, \ldots, \lambda_N$ to account for the constraint from each time step as well as the initial condition, resulting in

$$\mathcal{L} = \Psi^{\text{approx}} - \lambda_0^T \left(X_0 - \mathcal{I}\right) - \sum_{n=0}^{N-1} \lambda_{n+1}^T \left(X_{n+1} - \mathcal{N}(X_n)\right). \tag{11}$$

Differentiating equation (11) w.r.t $\mathcal{I}$ leads to

$$\begin{aligned} \frac{d\mathcal{L}}{d\mathcal{I}} = \lambda_0^T - \left(\frac{d\psi}{dX}(X_N) - \lambda_N^T\right) \frac{\partial X_N}{\partial \mathcal{I}} \\ - \sum_{n=0}^{N-1} \left(\lambda_n^T - \lambda_{n+1}^T \frac{d\mathcal{N}}{dX}(X_n)\right) \frac{\partial X_n}{\partial \mathcal{I}}. \end{aligned} \tag{12}$$

By defining λ to be the solution of the discrete adjoint model,

$$\begin{aligned} \lambda_N = \left(\frac{d\psi}{dX}(X_N)\right)^T, \quad \lambda_n = \left(\frac{d\mathcal{N}}{dX}(X_n)\right)^T \lambda_{n+1}, \\ n = N-1, \ldots, 0, \end{aligned} \tag{13}$$

we obtain the gradient $\nabla_{\mathcal{I}} \Psi^{\text{approx}} = \lambda_0$.

For the general case where the objective function contains integral terms as in the general form (10) and sensitivities to parameters are also desired, the discrete adjoint model can be derived in a similar way from the extended system

$$\underline{\mathcal{M}}\dot{\underline{X}} = \underline{F}(t, \underline{X}), \tag{14}$$

where

$$\underline{\mathcal{M}} = \begin{bmatrix} \mathcal{M} & & \\ & \mathbf{I}_{n_p \times n_p} & \\ & & 1 \end{bmatrix}, \underline{X} = \begin{bmatrix} X \\ p \\ q \end{bmatrix}, \underline{F} = \begin{bmatrix} F \\ \mathbf{0}_{n_p \times 1} \\ r \end{bmatrix}.$$

The second equation enforces constant parameters during the simulation, and the last equation comes from a transformation of the integral

$$q = \int_{t_0}^{t_F} r(t, X) dX.$$

The initial condition for the extended system is $\underline{X}_0 = [\mathcal{I}\ \mathbf{0}_{1\times n_p}\ 0]^T$.

With the basic framework established, the discrete adjoint for any one-step method can be easily derived. For example, the adjoint theta method (8) is

$$\mathcal{M}^T \lambda_s = \lambda_{n+1} + h_n \theta F_X^T(X_{n+1})\lambda_s + h_n \theta r_X^T(t_{n+1}, X_{n+1}), \tag{15}$$

$$\begin{aligned} \lambda_n = {} & \mathcal{M}^T \lambda_s + h_n(1-\theta)\, F_X^T(X_n)\lambda_s \\ & + h_n(1-\theta)\, r_X^T(t_n, X_n), \end{aligned} \tag{16}$$

$$\begin{aligned} \mu_n = {} & \mu_{n+1} + h_n \left(\theta\, F_p^T(X_{n+1}) + (1-\theta)\, F_p^T(X_n)\right) \lambda_s \\ & + h_n \left(\theta\, r_p^T(t_{n+1}, X_{n+1}) + (1-\theta)\, r_p^T(t_n, X_n)\right), \end{aligned} \tag{17}$$

$$n = N-1, \ldots, 0,$$

with the terminal conditions

$$\lambda_N = \left(\frac{d\psi}{dX}(X_N)\right)^T, \quad \mu_N = \left(\frac{d\psi}{dp}(X_N)\right)^T. \tag{18}$$

The gradients of the objective functions are given as

$$\nabla_{\mathcal{I}} \Psi^{\text{approx}} = \lambda_0, \quad \nabla_p \Psi^{\text{approx}} = \mu_0.$$

If the terminal condition for λ_N in (18) is applied to the discrete adjoint of a DAE system, however, there would be conflicts with the constraints brought up by the algebraic equations. Consider the simple case $\theta = 1$ (backward Euler method), and let λ^x and λ^y be the discrete differential and algebraic adjoint variables, respectively. One can see that (16) will lead to $\lambda^y = 0$ regardless of the terminal condition for λ_N^y. According to the implicit function theorem, the algebraic variable y can be locally solved from (6) as

$$y = \varphi(x; p).$$

Substituting y into the objective functions in the terminal conditions (18) leads to that λ_N^x depend only on x and λ_N^y should be set to zero.

For the sensitivity calculation, we observe the following.

- The discrete adjoint equations (15) propagate the sensitivity variables backward in time following exactly the same trajectory with the forward run. Thus, there is no time step control in the backward run. While this approach may result in increased memory requirements compared with forward approaches, that requirement can be dramatically reduced with small increases in recomputation by using advanced checkpointing techniques [21].
- The number of variables λ and/or μ is the same as the number of objective functions.
- For each objective function, only one linear system needs to be solved for the theta methods in (15) at each backward step, regardless of the number of parameters. In the general case, the number of linear solves depends on the time-stepping method. For example, implicit Runge-Kutta methods may require as many linear solves as the number of stages.
- The "prediction" matrix $\mathcal{M}^T/(h_n\,\theta) - F_X^T$ from (15) is the transpose of the one used in solving the nonlinear equation (8). The Jacobian F_X can be reused in the adjoint run.
- The adjoint computation may require some extra Jacobian functions depending on the needs of the application, as summarized in Table II.

TABLE II
JACOBIANS REQUIRED WHEN CALCULATING ADJOINT SENSITIVITIES W.R.T. INITIAL CONDITIONS AND PARAMETERS RESPECTIVELY

	ψ Only	With Integral
initial conditions	F_X	F_X, r_X
parameters	F_X, F_p	F_X, F_p, r_X, r_p

For the examples in our experiments, we store the entire forward trajectory in memory in order to avoid recomputation, since the memory capacity is sufficient. Nevertheless, we have also implemented a variety of advanced checkpointing schemes [21] for large-scale problems.

A. Sensitivity calculation with discontinuities

For illustration, we consider the following case of the hybrid system (1)–(3) that has a single discontinuity at time τ separating the system into two stages:

$$\begin{aligned} X^{(1)}(t_0) &= \mathcal{I}, \\ \mathcal{M}\dot{X}^{(1)} &= F^{(1)}(X^{(1)}), \quad t \in [t_0, \tau], \\ \gamma(X^{(1)}(\tau)) &= 0, \\ \mathcal{M}\dot{X}^{(2)} &= F^{(2)}(X^{(2)}), \quad t \in (\tau, t_f]. \end{aligned}$$

Here $\mathcal{I}$ is the initial condition, and γ is transition function between stages. The approach for this case can be straightforwardly extended to multiple stages. We again assume that the discretization of the hybrid systems is performed with one-step methods:

$$\begin{aligned} X_{k+1}^{(1)} &= \mathcal{N}^{(1)}(X_k^{(1)}), \quad k = 0 \ldots N_1 - 1, \\ X_{k+1}^{(2)} &= \mathcal{N}^{(2)}(X_k^{(2)}), \quad k = N_1 \ldots N - 1, (N = N_1 + N_2). \end{aligned}$$

The objective function Ψ is approximated by using the numerical solution:

$$\Psi \approx \Psi^{\texttt{approx}} = \psi(X_{N_2}).$$

The following assumptions are made about this model for the convenience of analysis.

1) The differential states in $X^{(2)}$ and $X^{(1)}$ are continuous at the junction time

$$x^{(2)}(\tau) = x^{(1)}(\tau).$$

2) $F^{(1)}$, $F^{(2)}$, and γ are $\mathcal{C}^1$.
3) The transversality condition is satisfied [31]:

$$\frac{d\gamma}{dX}(\tau)F^{(1)}(X^{(1)}(\tau)) \neq 0.$$

B. Jump conditions for discrete adjoint method

Similar to the steps taken in Section III, we build the Lagrangian function:

$$\begin{aligned}\widehat{\mathcal{L}} = \Psi^{\text{approx}} &- \left(\lambda_0^{(1)}\right)^T \left(X_0^{(1)} - \mathcal{I}\right) \\ &- \sum_{k=0}^{N_1-1} \left(\lambda_{k+1}^{(1)}\right)^T \left(X_{k+1}^{(1)} - \mathcal{N}^{(1)}(X_k^{(1)})\right) \\ &- \sum_{k=N_1}^{N-1} \left(\lambda_{k+1}^{(2)}\right)^T \left(X_{k+1}^{(2)} - \mathcal{N}^{(2)}(X_k^{(2)})\right). \quad (19)\end{aligned}$$

Differentiating the Lagrangian function (19) at $\mathcal{I}$ and cancelling out identical terms yield

$$\begin{aligned}\frac{d\widehat{\mathcal{L}}}{d\mathcal{I}} &= \frac{d\psi}{dX}(X_{N_2})\frac{\partial X_{N_2}}{\partial \mathcal{I}} - \cancel{\frac{d\lambda_0^T}{d\mathcal{I}}(X_0 - \mathcal{I})} - \left(\lambda_0^{(1)}\right)^T \frac{\partial X_0^{(1)}}{\partial \mathcal{I}} \\ &+ \left(\lambda_0^{(1)}\right)^T - \cancel{\sum_{k=0}^{N_1-1} \left(\frac{d\lambda_{k+1}^{(1)}}{d\mathcal{I}}\right)^T \left(X_{k+1}^{(1)} - \mathcal{N}^{(1)}(X_k^{(1)})\right)} \\ &- \sum_{k=0}^{N_1-1} \left(\lambda_{k+1}^{(1)}\right)^T \left(\frac{\partial X_{k+1}^{(1)}}{\partial \mathcal{I}} - \frac{d\mathcal{N}^{(1)}}{dX}(X_k^{(1)})\frac{\partial X_k^{(1)}}{\partial \mathcal{I}}\right) \\ &- \cancel{\sum_{k=N_1}^{N-1} \left(\frac{d\lambda_{k+1}^{(2)}}{d\mathcal{I}}\right)^T \left(X_{k+1}^{(2)} - \mathcal{N}^{(2)}(X_k^{(2)})\right)} \\ &- \sum_{k=N^{(1)}}^{N-1} \left(\lambda_{k+1}^{(2)}\right)^T \left(\frac{\partial X_{k+1}^{(2)}}{\partial \mathcal{I}} - \frac{d\mathcal{N}^{(2)}}{dX}(X_k^{(2)})\frac{\partial X_k^{(2)}}{\partial \mathcal{I}}\right) \\ &= \frac{d\psi}{dX}(X_{N_2})\frac{\partial X_{N_2}}{\partial \mathcal{I}} - \left(\lambda_0^{(1)}\right)^T \frac{\partial X_0^{(1)}}{\partial \mathcal{I}} + \left(\lambda_0^{(1)}\right)^T \\ &- \sum_{k=1}^{N_1} \left(\lambda_k^{(1)}\right)^T \frac{\partial X_k^{(1)}}{\partial \mathcal{I}} + \sum_{k=0}^{N_1-1} \left(\lambda_{k+1}^{(1)}\right)^T \frac{\partial \mathcal{N}^{(1)}}{\partial X}(X_k^{(1)})\frac{\partial X_k^{(1)}}{\partial \mathcal{I}} \\ &- \sum_{k=N_1+1}^{N} \left(\lambda_k^{(2)}\right)^T \frac{\partial X_k^{(2)}}{\partial \mathcal{I}} \\ &+ \sum_{k=N_1}^{N-1} \left(\lambda_{k+1}^{(1)}\right)^T \frac{d\mathcal{N}^{(2)}}{dX}(X_k^{(2)})\frac{\partial X_k^{(2)}}{\partial \mathcal{I}}.\end{aligned}$$

Substituting

$$\begin{aligned}\sum_{k=1}^{N_1} \left(\lambda_k^{(1)}\right)^T \frac{\partial X_k^{(1)}}{\partial \mathcal{I}} &= \left(\lambda_{N_1}^{(1)}\right)^T \frac{\partial X_{N_1}^{(1)}}{\partial \mathcal{I}} \\ &- \left(\lambda_0^{(1)}\right)^T \frac{\partial X_0^{(1)}}{\partial \mathcal{I}} + \sum_{k=0}^{N_1-1} \left(\lambda_k^{(1)}\right)^T \frac{\partial X_k^{(1)}}{\partial \mathcal{I}}, \\ \sum_{k=N_1+1}^{N} \left(\lambda_k^{(2)}\right)^T \frac{\partial X_k^{(2)}}{\partial \mathcal{I}} &= \left(\lambda_N^{(2)}\right)^T \frac{\partial X_{N_2}}{\partial \mathcal{I}} \\ &- \left(\lambda_{N_1}^{(2)}\right)^T \frac{\partial X_{N_1}^{(2)}}{\partial \mathcal{I}} + \sum_{k=N_1}^{N-1} \left(\lambda_k^{(2)}\right)^T \frac{\partial X_k^{(1)}}{\partial \mathcal{I}}\end{aligned}$$

and then reorganizing leads to

$$\begin{aligned}\frac{d\widehat{\mathcal{L}}}{d\mathcal{I}} &= \left(\lambda_0^{(1)}\right)^T + \left(\frac{d\psi}{dX}(X_{N_2}) - \left(\lambda_N^{(2)}\right)^T\right)\frac{\partial X_{N_2}}{\partial \mathcal{I}} \\ &- \left(\lambda_{N_1}^{(1)}\right)^T \frac{\partial X_{N_1}^{(1)}}{\partial \mathcal{I}} + \left(\lambda_{N_1}^{(2)}\right)^T \frac{\partial X_{N_1}^{(2)}}{\partial \mathcal{I}} \\ &- \sum_{k=0}^{N_1-1} \left(\left(\lambda_k^{(1)}\right)^T - \left(\lambda_{k+1}^{(1)}\right)^T \frac{d\mathcal{N}^{(1)}}{dX}(X_k^{(1)})\right)\frac{\partial X_k^{(1)}}{\partial \mathcal{I}} \\ &- \sum_{k=N_1}^{N-1} \left(\left(\lambda_k^{(2)}\right)^T - \left(\lambda_{k+1}^{(2)}\right)^T \frac{d\mathcal{N}^{(2)}}{dX}(X_k^{(2)})\right)\frac{\partial X_k^{(2)}}{\partial \mathcal{I}}. \quad (20)\end{aligned}$$

We define λ to be the solution of the discrete adjoint model:

$$\begin{aligned}&\lambda_N^{(2)} = \left(\frac{d\psi}{dX}(X_{N_2})\right)^T, \\ &\lambda_k^{(2)} = \left(\frac{d\mathcal{N}^{(2)}}{dX}(X_k^{(2)})\right)^T \lambda_{k+1}^{(2)},\ k = N-1, \ldots, N_1, \\ &\left(\lambda_{N_1}^{(1)}\right)^T \frac{\partial X_{N_1}^{(1)}}{\partial \mathcal{I}} = \left(\lambda_{N_1}^{(2)}\right)^T \frac{\partial X_{N_1}^{(2)}}{\partial \mathcal{I}}, \\ &\lambda_k^{(1)} = \left(\frac{d\mathcal{N}^{(1)}}{dX}(X_k^{(1)})\right)^T \lambda_{k+1}^{(1)},\ k = N_1 - 1, \ldots, 0.\end{aligned}$$

Then we have

$$\nabla_{\mathcal{I}}\Psi^{\text{approx}} = \left(d\widehat{\mathcal{L}}/d\mathcal{I}\right)^T = \lambda_0^{(1)}.$$

To avoid computing the forward sensitivities $\partial X_{N_1}^{(1)}/\partial \mathcal{I}$ and $\partial X_{N_1}^{(2)}/\partial \mathcal{I}$, we use the results from [31, Equation 50 and Theorem 1]:

$$\frac{d\tau}{d\mathcal{I}} = -\frac{\frac{d\gamma}{dX}(X_{N_1}^{(1)})\frac{\partial X_{N_1}^{(1)}}{\partial \mathcal{I}}}{\frac{d\gamma}{dX}(X_{N_1}^{(1)})\frac{\partial X_{N_1}^{(1)}}{\partial t}}$$

and

$$\frac{\partial X_{N_1}^{(2)}}{\partial \mathcal{I}} = \frac{\partial X_{N_1}^{(1)}}{\partial \mathcal{I}} - \left(\frac{\partial X_{N_1}^{(2)}}{\partial t} - \frac{\partial X_{N_1}^{(1)}}{\partial t}\right)\frac{d\tau}{d\mathcal{I}}.$$

Then we obtain the sensitivity transfer equation

$$\lambda_{N_1}^{(1)} = \left(\mathbf{I} + \left(\frac{\partial X_{N_1}^{(2)}}{\partial t} - \frac{\partial X_{N_1}^{(1)}}{\partial t} \right) \frac{\frac{d\gamma}{dX}(X_{N_1}^{(1)})}{\frac{d\gamma}{dX}(X_{N_1}^{(1)}) \frac{\partial X_{N_1}^{(1)}}{\partial t}} \right)^T \lambda_{N_1}^{(2)}. \quad (21)$$

If we apply the analysis to the extended system (14), we will obtain an additional transfer equation for the sensitivity variable μ as

$$\mu_{N_1}^{(1)} = \mu_{N_1}^{(2)} + \left(\left(\frac{\partial X_{N_1}^{(2)}}{\partial t} - \frac{\partial X_{N_1}^{(1)}}{\partial t} \right) \frac{\frac{d\gamma}{dp}(X_{N_1}^{(1)})}{\frac{d\gamma}{dX}(X_{N_1}^{(1)}) \frac{\partial X_{N_1}^{(1)}}{\partial t}} \right)^T \lambda_{N_1}^{(2)}. \quad (22)$$

A similar derivation for the forward sensitivity analysis is developed in Appendix A.

C. Implementation

The sensitivity analysis capabilities have been implemented in the open-source high-performance numerical library PETSc [27], [32] freely available at https://bitbucket.org/petsc/. Figure 2 describes the overall structure of the PETSc adjoint solver. The solver first solves the ODE/DAE equation checkpointing the solutions in the forward run and then conducts a backward run propagating the sensitivity variables. `TSStep` implements a time stepping scheme to propagate the solution one step ahead and the choice can be changed at run time. `TSMonitor` executes callbacks from users to monitor the solution (read-only) at each time step. These are existing components in PETSc's time-stepping library `TS`. For this work, three additions (made by the first three authors and available with the PETSc distribution) were developed necessary for handling discontinuities and discrete sensitivity calculations.

1) The `TSEvent` object supports detecting events (zero-crossing of discontinuities), while performing the numerical integration, through an interpolation-based root-finding approach. It also allows users to add a post-event handler that can conduct important operations such as modification of the right-hand side function, re-initialization of the DAE system, and application of the jump conditions.
2) The `TSTrajectory` object provides a variety of sophisticated online and offline checkpointing schemes for computing platforms with single-level storage media (e.g. RAM) or multilevel storage media (e.g., RAM and external disk/tape). Checkpoints can be stored with the `TSTrajectorySet` method and restored with the `TSTrajectoryGet` method. `TSTrajectoryGet` may recompute from restored solution to obtain the information needed by an adjoint step. Recomputation typically happens when checkpoints are stored only at selective time steps because of limited storage capacity with the strategy determined by a checkpointing scheduler such as `revolve` [33].
3) The `TSAdjoint` object calculates the sensitivities using a discrete adjoint approach. By design, its components are either reused or modified from the original `TS` solver. For example, `TSAdjointStep` corresponds to the adjoint version of `TSStep` and they have similar underlying infrastructures.

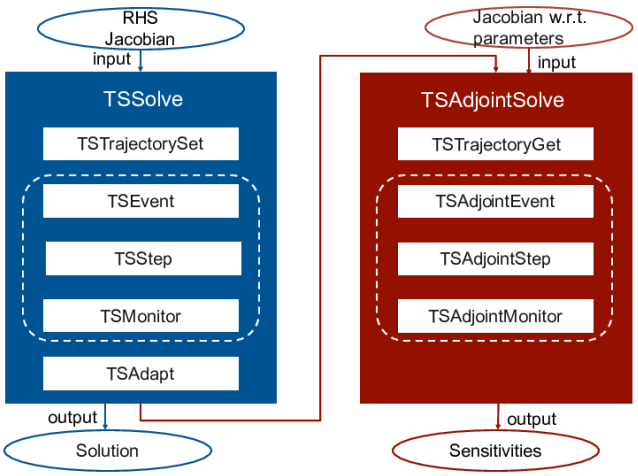


Fig. 2. Structure of PETSc implementation for adjoint sensitivity analysis.

All the components including existing ones and newly developed ones are compatible with each other and used together within the highly composable solver for calculating the sensitivities of hybrid systems.

Furthermore, PETSc is specifically designed to be scalable at extreme scale. This feature can also be enjoyed by our sensitivity analysis framework. The parallelism in the original simulation naturally carries over to the adjoint run when the sensitivities to initial conditions are calculated. Parallelization of the adjoint run for computing the sensitivity to random parameters cannot be automatically performed by the solver. Nevertheless, it is achievable by implementing application-dependent decomposition strategy though the PETSc `DM` object.

IV. Numerical Examples

This section illustrates the accuracy and computational efficiency of the adjoint discrete sensitivity analysis approaches on hybrid systems. First, we compare the accuracy of the discrete forward and adjoint approach on the hybrid system example given in [1]. Next, we illustrate the benefit of using the adjoint sensitivity results in optimization. Then, we present the computational efficiency comparison of the discrete sensitivity approaches on 9-bus and 118-bus test examples. All simulations are performed with PETSc. The solutions are checkpointed at each time step so that no recomputation is involved, constituting an ideal scenario for adjoint computation.

A. Hybrid system example

The hybrid system given in [1] is governed by

$$\dot{x} = A_i x, \quad (23)$$

where x has two components x_1 and x_2 and A_i is an matrix that changes from

$$A_1 = \begin{bmatrix} 1 & -100 \\ 10 & 1 \end{bmatrix} \quad \text{to} \quad A_2 = \begin{bmatrix} 1 & 10 \\ -100 & 1 \end{bmatrix}$$

when the switching condition $x_2 = 2.75x_1$ is satisfied and from A_2 to A_1 when $x_2 = 0.36x_1$. The initial condition is $\mathcal{I} = [0\ 1]^T$, and A_1 is used. We are interested in the trajectory sensitivities of x_1 and x_2 to the parameter $p = 2.75$ in the first switching condition.

Figure 3 shows the trajectory sensitivities to the perturbation of p computed with the discrete forward approach and the discrete adjoint approach. The system is discretized by using the Crank-Nicolson scheme with an initial time step of 0.001 seconds. PETSc monitors signs of the switching conditions (e.g., $x_2 - 2.75x_1$) at each time step and rolls back the step if the signs change indicating that an event has been stepped over. A new time step estimated by using linear interpolation will then be attempted repeatedly until the event point is reached within a certain numerical tolerance that the user can control (by default it is set to $1e-6$). After the event, the step size will be adjusted so that the two steps before and after the event sum up to 0.001.

To see how the theory on the adjoints is applied to this problem, consider the calculation of the trajectory sensitivity of the solution component x_1 with respect to initial condition and parameter p, represented by $\lambda = [\partial x_1/\partial \mathcal{I}]$ and $\mu = \partial x_1/\partial p$, respectively. The terminal conditions are $\lambda = [1\ 0]$ and $\mu = 0$. According to the jump conditions (21) (22), the adjoint variables should be transferred at the switching point by

$$\lambda^{new} = \left(\mathbf{I} + (A_2 - A_1)x\frac{[-2.75\ 1]}{[-2.75\ 1]A_1x}\right)^T \lambda^{old}$$
$$\mu^{new} = \mu^{old} + \left((A_2 - A_1)x\frac{[-2.75\ 1]}{[-2.75\ 1]A_1x}\right)^T \lambda^{old}$$

when A_2 switches to A_1 in the adjoint run and

$$\lambda^{new} = \left(\mathbf{I} + (A_1 - A_2)x\frac{[-0.36\ 1]}{[-0.36\ 1]A_1x}\right)^T \lambda^{old}$$
$$\mu^{new} = \mu^{old} + \left((A_1 - A_2)x\frac{[-0.36\ 1]}{[-0.36\ 1]A_1x}\right)^T \lambda^{old}$$

when A_1 changes to A_2.

The sensitivities are plotted for different simulation times ranging from 0 to 0.2 seconds. As expected, both sensitivities $\partial x_1/\partial p$ and $\partial x_2/\partial p$ jump at switching points and decay to zero as the trajectory approaches the equilibrium point. The results of the two different methods show good agreement with each other (the numerical values match for 15 digits), as well as with the result presented in Fig. 6 of [1].

B. Maximization of generator mechanical power input

This simple power system example highlights the impact of the accuracy of adjoint sensitivities on the optimization process. Typically, such an approximated dynamic model of the generator may not be applied in practice. We provide this simplistic example merely to illustrate the accuracy of the discrete adjoint method. We consider a maximization objective of the mechanical power input P_m subject to the generator swing equations and a constraint on the maximum rotor deviation $\delta(t) \leq \delta_{max}, \forall t$. This can be reformulated as a minimization with a penalty term on the rotor angle deviation as follows:

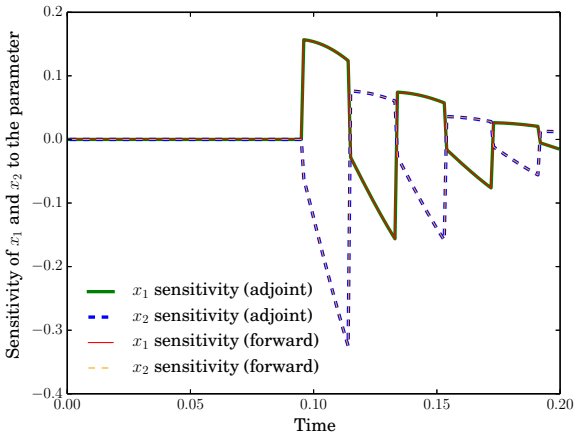


Fig. 3. Trajectory sensitivities for a hybrid example from [1].

$$\min -P_m + \sigma \int_{t_0}^{t_F} \max\left(0, \delta - \delta_{max}\right)^\eta \ \mathrm{d}t \tag{24}$$

s.t.

$$\frac{d\delta}{dt} = \omega_B\left(\omega - \omega_s\right) \tag{25}$$
$$\frac{d\omega}{dt} = \frac{\omega_s}{2H}\left(P_m - P_{max}\sin(\delta) - D(\omega - \omega_s)\right). \tag{26}$$

Here, η is an exponent to ensure sufficient smoothness, and σ is a multiplier to ensure decent progress of the optimization. The optimization problem is solved with the bounded limited-memory variable-metric (BLMVM) algorithm in the TAO solver included in the PETSc package. The initial guess of P_m is 1.06. The convergence tolerances are all set to 10^{-14}, and δ_{max} is set to 1 radian (i.e., 57.27 degrees).

Table III shows the gradients computed with the two discrete adjoint approaches compared with finite differences, with a step size 7.45×10^{-9} (comparable to the optimal choice of square root of machine precision [21]), at the first three iterations of the optimization. As shown in Table III, the results of the two discrete approaches agree with each other and are close to the finite-difference approximations. Figure 4 shows the convergence behavior using the gradients from the three different methods. The forward and adjoint sensitivities can make the optimization process converge to the optimal value 1.0079 after 13 iterations. On the other hand, the finite-difference approximations cause the optimization to stall with a residual of 10^{-6}. This is an expected downside of the reduced precision of finite differences, now demonstrated on a power grid example.

TABLE III
COMPARISON OF PARAMETER AND GRADIENT OBTAINED WITH DIFFERENT METHODS DURING OPTIMIZATION PROCESS

Iteration No.		Adjoint	Forward	Finite Difference
0	P_m	1.06	1.06	1.06
	gradient	140.0487958	140.0487958	140.0487323
1	P_m	1.032130009	1.032130009	1.032129996
	gradient	45.40765371	45.40765371	45.40760848
2	P_m	1.018758331	1.018758331	1.018758323
	gradient	14.84698503	14.84698503	14.84697329

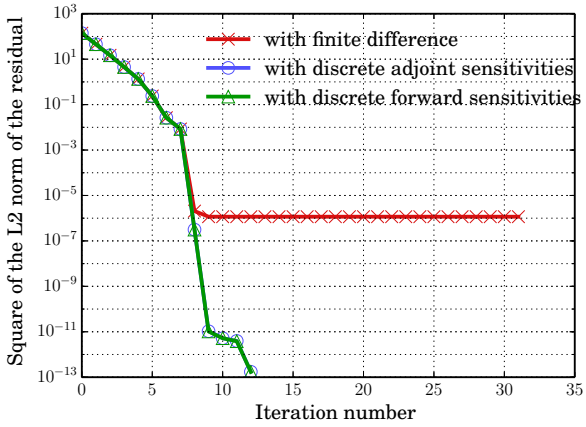


Fig. 4. Convergence of the optimization process using gradients obtained with three methods.

C. Sensitivity of frequency violations

Sensitivity-based approaches are necessary for solving dynamic security-constrained OPF (DSCOPF) problems that include a frequency constraint. In [8], [34], the sensitivities of the generator frequency violations have been used to obtain a transient security-constrained dispatch. The computational costs of the approaches proposed therein, finite-differencing and forward sensitivity, can be high, especially when the number of parameters to be optimized becomes large. We have compared forward and adjoint sensitivity calculation approaches for DSCOPF on two test systems: the IEEE 9-bus and 118-bus. The 9-bus test case used in this work is the 3-generator, 9-bus system available in [35] with the dynamic data from Chapter 7 of [36]. All generators are modeled as a fourth-order *dq* two-axis model with an IEEE Type-1 DC exciter, shown in Figure 5, for maintaining the generator terminal voltage, and a TGOV1 type steam turbine governor model to regulate the generator frequency. The power system dynamics equations are integrated by using the implicit trapezoidal method with a time step of 0.01 seconds and a simulation horizon of 1 second. In our model we include discontinuities that are both time-based (they occur at prescribed times) and state-based (at which the transition is induced by state-dependent switching function). The time-based discontinuities are initiated by faults, and they consist of a six-cycle self-clearing three-phase fault applied on bus 1 for the 118-bus system and bus 9 for the 9-bus system. The state based discontinuity is initiated when the voltage regulator output

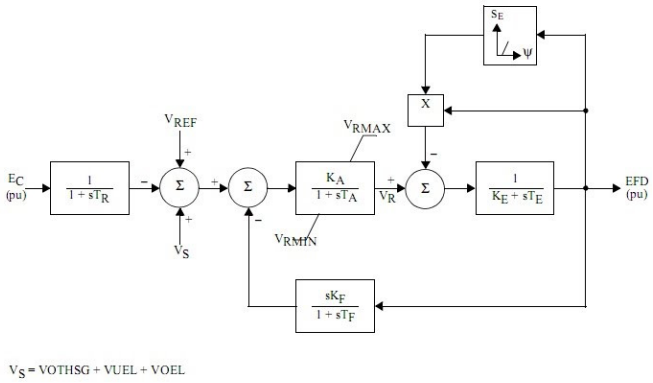


Fig. 5. IEEE type-1 DC exciter model.

reaches its minimum or maximum limit.

Following [34], the sensitivities are evaluated for the following dynamic security metric that measures the severity of frequency violation for each generator:

$$H_i(x,y) = \sigma \int_0^T \left[\max(0, \omega_i - \omega^+, \omega^- - \omega_i)\right]^\eta dt, \quad i = 1, \ldots, m. \quad (27)$$

Here, ω_i is the speed of the generator i, m is the total number of generators, σ and η follow the conventions in (24), and ω^+ and ω^- are the maximum and minimum limits, respectively, on the generator frequencies. We aim to find the sensitivity of the constraint function H_i with respect to the parameters (i.e., the generator active and reactive dispatch and the bus voltage magnitudes and angles at time t_0). The number of states for the differential-algebraic system and the parameters associated with the two systems are listed in Table IV. Note that other metrics may exist for dynamic security. They can also be easily handled in our framework as soon as they can be represented in the general form of the objective function 10.

TABLE IV
SETTINGS FOR THE 9-BUS AND 118-BUS SYSTEMS

	No. of Variables	No. of Parameters	No. of Functions
9 bus	42	24	3
118 bus	884	344	54

Figure 6 shows the generator frequencies, the frequency violations, and the sensitivities of the frequency violations w.r.t the initial dispatch of the three generators for the 9-bus system. Following a fault on bus 9, the frequencies of the three generators deviate from the nominal trajectory, with generator 3 having the largest frequency deviation because of its close proximity to the fault location. The shaded regions in the frequency plots represent the contributions to the frequency violation measure H_i. Generator 1, with the largest inertia, has the smallest frequency deviation and does not exceed ω^+=60.5 Hz. As a result H_1=0. The sensitivities of the frequency violation measure w.r.t. generator initial dispatch, $\partial H/\partial P_g$

are also shown in Fig. 6 and Table V. As seen in Table V, generators 2 and 3 have the largest sensitivities for a fault at bus 9, while generator 1 has the smallest one. This sensitivity information can serve as an important metric for performing generator redispatch decisions.

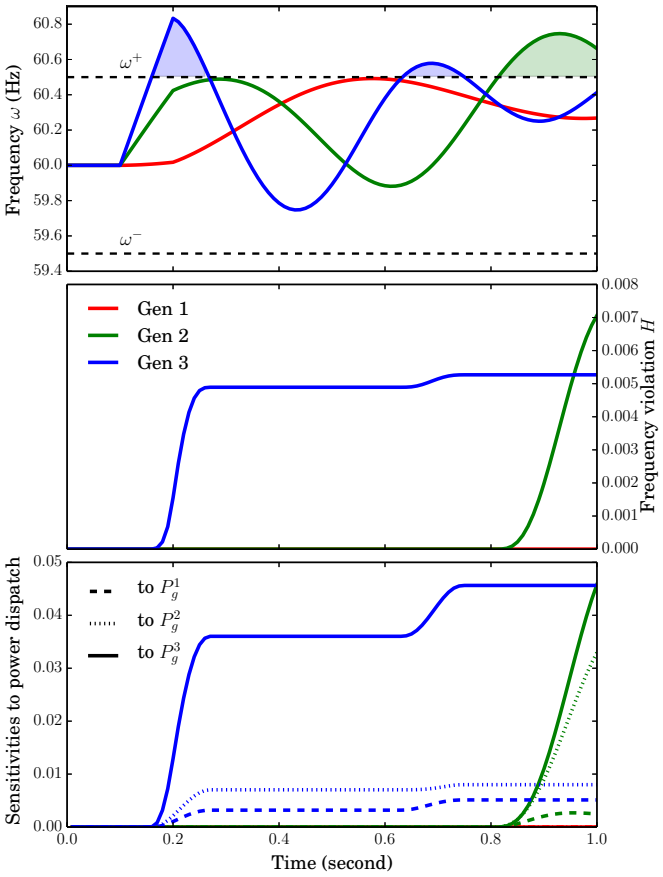


Fig. 6. Plots of generator frequencies (top), frequency violations (middle), and sensitivities of frequency violations to initial power dispatch (bottom) for the 9-bus system. The frequency limits are ω^+=60.5 Hz and ω^-=59.5 Hz.

TABLE V
SENSITIVITIES OF FREQUENCY VIOLATIONS (H_i) WITH RESPECT TO POWER DISPATCH PARAMETERS (P_g) AT TIME $t = 1$ SECONDS

	P_g^1	P_g^2	P_g^3
H_1	0	0.002465	0.005118
H_2	0	0.032956	0.007997
H_3	0	0.045809	0.045636

Figure 7 shows the dynamics of the voltage regulator outputs $V_R^i, i = 1, 2, 3$. V_R^2 and V_R^3 reach their maximum limit. Generator 2 continues to operate at its maximum voltage limit, while V_R^3 drops below the maximum limit after about 0.9 seconds. The sensitivities of the voltage regulator output w.r.t. the generator terminal voltage magnitudes, $\partial V_R^i / \partial V_m^i$, are shown in Fig. 7. This plot shows a jump in the adjoint sensitivities when the maximum limit is reached or abandoned, which is accurately captured by our method.

Table VI compares the difference between forward and adjoint sensitivity values in terms of maximum norm for

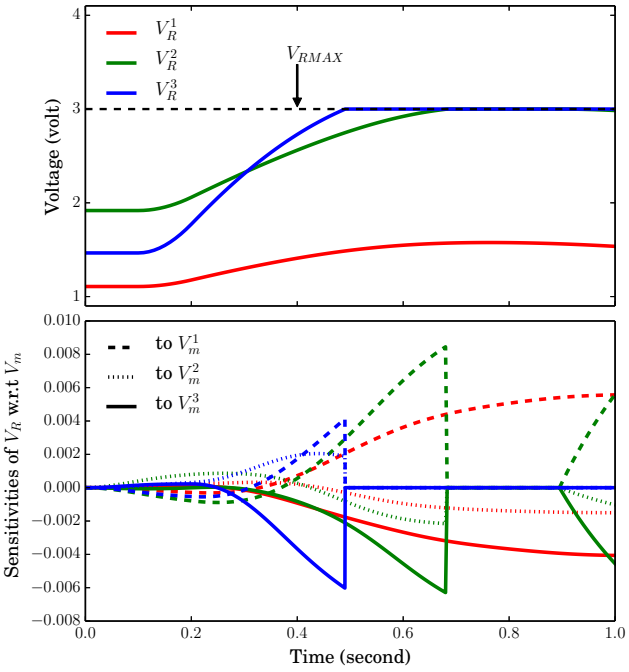


Fig. 7. Response of voltage regulator output (top) and its sensitivities w.r.t. generator terminal voltage magnitude (bottom).

various time simulation intervals. All the observed discrepancies are close to machine precision (around 1e−15). Table

TABLE VI
DIFFERENCE OF THE SENSITIVITY RESULTS FOR ADJOINT AND FORWARD APPROACHES (IN MAXIMUM NORM)

	$t = 0.5$s	$t = 0.6$s	$t = 0.7$s	$t = 0.8$s	$t = 0.9$s	$t = 1$s
9 bus	2.1e−16	3.3e−16	7.2e−16	8.3e−16	1.1e−15	1.8e−15
118 bus	6.1e−16	8.9e−16	1.0e−15	1.3e−15	1.7e−15	1.2e−15

VII presents the computational results of the two sensitivity approaches. For both systems, one can see that the adjoint approach is faster than the forward sensitivity approach. Note that the execution time listed in Table VII for the forward and adjoint approaches also includes the execution time for the dynamics simulation. The adjoint approach is faster than the forward approach by 2.4X and 7.7X for the 9-bus and 118-bus systems, respectively. Larger speedups can be expected for larger networks or systems with more parameters.

These results demonstrate that discrete adjoint approaches are significantly more effective than their forward versions in the regimes described in this paper and that they can accurately compute derivatives of numerically simulated trajectories, even when the added complexity of system switching is present.

TABLE VII
TIMING RESULTS FOR THE 9-BUS AND 118-BUS SYSTEMS

	Forward	Adjoint	Simulation
9 bus	0.12 s	0.05 s	0.03 s
118 bus	14.00 s	1.82 s	0.33 s

V. Conclusions

This paper presents an efficient approach for computing sensitivities of large-scale power systems using a discrete adjoint method and shows that it is a competitive alternative to forward sensitivity analysis approaches such as TSA when many number of system parameters are involved. To accommodate the switching dynamics present in many applications, such as the one induced by DC exciters, we derived the adjoint jump conditions that allow the accurate computation of parametric derivatives by an adjoint approach. Numerical results on several test systems and examples have been compared with the forward sensitivity approach demonstrating the machine-precision-level accuracy and efficiency of the proposed method. In particular, the discrete sensitivity approach has been demonstrated to be much faster compared with the forward sensitivity approach, and in the 118-bus case it resulted in 7.7X speedup. To our knowledge, this is the first time discrete adjoint computations have been demonstrated in the power systems area for test cases of the size discussed here, which moreover included switching dynamics. Their significance resides in the fact that no other sensitivity method both achieves the kind of speedup described above and provides a gradient to a target functional to machine-precision quality. For applications such as optimization of large systems with transient security constraints, both these features are important. All the algorithms described in this paper are publicly available through the widely used open-source numerical library PETSc. At the time this paper was written, the sensitivity analysis capability in PETSc had been applied to parameter estimation of dynamical power systems with uncertainty [37]. Future extensions will include the usage of advanced checkpointing for reducing the memory footprint and sensitivity calculation of larger systems, such as interconnect-size ones.

Appendix A
Discrete forward approach

We take the derivative of the one-step time integration algorithm and obtain the discrete forward model. For example, differentiation of the theta methods (8) at parameter p will lead to

$$\begin{aligned}\mathcal{M}\mathcal{S}_{\ell,n+1} = \mathcal{M}\mathcal{S}_{\ell,n} + h_n\big((1-\theta)\left(F_X(X_n)\mathcal{S}_{\ell,n} + F_{p_\ell}(X_n)\right) \\ + \theta\left(F_X(X_{n+1})\mathcal{S}_{\ell,n+1} + F_{p_\ell}(X_{n+1})\right)\big). \quad (28)\end{aligned}$$

Here $\mathcal{S}_{\ell,n} = dX_n/dp_\ell, 1 \le \ell \le m$, denote the solution sensitivities (also known as trajectory sensitivities). One can verify that this approach leads to the same formulation as with the traditional forward approach when using the same theta method and step size for solving the continuous sensitivity equation.

With the solution sensitivities, the total derivative of $\psi(X_N)$ can be computed by using

$$\frac{d\psi}{dp_\ell}(X_N) = \frac{\partial\psi}{\partial X}(X_N)\mathcal{S}_{\ell,N} + \frac{\partial\psi}{\partial p_\ell}(X_N). \quad (29)$$

Let q be the integral term in (10). The total derivative of q to parameters p is given as

$$\frac{dq}{dp_\ell} = \int_{t_0}^{t_F}\left(\frac{\partial r}{\partial X}(t,X)\mathcal{S}_\ell + \frac{\partial r}{\partial p_\ell}(t,X)\right)dt. \quad (30)$$

This integral must be calculated with the same time-stepping algorithm and sequence of time steps in the discrete approaches such that the derivative computed sticks tightly to the numerical procedure that is used to evaluate the objective function.

Note that for each parameter p_ℓ there is one variable $\mathcal{S}_\ell$ carrying the sensitivity information and one linear system arising from (28) to be solved at each time step. Thus, the computational cost of the forward approach is determined mainly by the number of parameters to which the sensitivities are desired.

The initial values for $\mathcal{S}_\ell$ follow directly from the condition (5). Since X consists of both differential variables and algebraic variables, $\mathcal{S}_\ell$ can also be separated into $\mathcal{S}_\ell^x$ and $\mathcal{S}_\ell^y$ corresponding to sensitivities associated with differential and algebraic parts of the solution respectively. Differentiating (5) yields the following relationship:

$$g_x\mathcal{S}_\ell^x + g_y\mathcal{S}_\ell^y + g_{p_\ell} = 0. \quad (31)$$

Given the value of $\mathcal{S}_\ell^x$ and the assumption that g_y is invertible, $\mathcal{S}_\ell^y$ could be solved from (31).

If the trajectory sensitivities to initial values are desired, we can also treat the initial values in the same way as parameters, and the derivatives to p_ℓ such as F_{p_ℓ}, $\partial\psi/\partial p_\ell$, $\partial r/\partial p_\ell$ and g_{p_ℓ} in (28)-(31) should be zeros.

We also present the sensitivity transfer equation used in forward method for completeness. Details on the derivation can be found in [31] and [1]. The jump conditions are

$$\mathcal{S}_{\ell,N_1}^{(2)} = \left(\mathbf{I} + \left(\frac{\partial X_{N_1}^{(2)}}{\partial t} - \frac{\partial X_{N_1}^{(1)}}{\partial t}\right)\frac{\frac{d\gamma}{dX}(X_{N_1}^{(1)})}{\frac{d\gamma}{dX}(X_{N_1}^{(1)})\frac{\partial X_{N_1}^{(1)}}{\partial t}}\right)\mathcal{S}_{\ell,N_1}^{(1)}.$$

References

[1] I. Hiskens and M. Pai, "Trajectory sensitivity analysis of hybrid systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 2, 2000.

[2] M. Zima and G. Andersson, "Stability assessment and emergency control method using trajectory sensitivities," *2003 IEEE Bologna Power Tech Conference Proceedings,*, vol. 2, 2003.

[3] K. N. Shubhanga and A. M. Kulkarni, "Determination of effectiveness of transient stability controls using reduced number of trajectory sensitivity computations," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 473–482, 2004.

[4] G. Hou and V. Vittal, "Trajectory sensitivity based preventive control of voltage instability considering load uncertainties," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2280–2288, nov 2012.

[5] A. Zamora-Cardenas and C. R. Fuerte-Esquivel, "Multi-parameter trajectory sensitivity approach for location of series-connected controllers to enhance power system transient stability," *Electric Power Systems Research*, vol. 80, no. 9, pp. 1096–1103, 2010.

[6] A. Nasri, R. Eriksson, and M. Ghandhari, "Using trajectory sensitivity analysis to find suitable locations of series compensators for improving rotor angle stability," *Electric Power Systems Research*, vol. 111, pp. 1–8, 2014.

[7] A. Bidram, M. Hamedani-golshan, and A. Davoudi, "Capacitor design considering first swing stability of distributed generations," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1941–1948, nov 2012.

[8] T. B. Nguyen, S. Member, and M. a. Pai, "Dynamic security-constrained rescheduling of power systems using trajectory sensitivities," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 848–854, 2003.

[9] J. Ma, D. Han, R. M. He, Z. Y. Dong, and D. J. Hill, "Reducing identified parameters of measurement-based composite load model," *IEEE Transactions on Power Systems*, vol. 23, no. 1, pp. 76–83, 2008.

[10] I. Hiskens, "Nonlinear dynamic model evaluation from disturbance measurements," *IEEE Transactions on Power Systems*, vol. 16, no. 4, pp. 702–710, 2001.

[11] T. B. Nguyen, M. a. Pai, and I. a. Hiskens, "Sensitivity approaches for direct computation of critical parameters in a power system," *International Journal of Electrical Power and Energy Systems*, vol. 24, no. 5, pp. 337–343, 2002.

[12] D. Chatterjee and A. Ghosh, "Transient stability assessment of power systems containing series and shunt compensators," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1210–1220, aug 2007.

[13] ——, "TCSC control design for transient stability improvement of a multi-machine power system using trajectory sensitivity," *Electric Power Systems Research*, vol. 77, no. 5-6, pp. 470–483, 2007.

[14] I. Hiskens and B. Gong, "MPC-based load shedding for voltage stability enhancement," *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.

[15] B. Sapkota and V. Vittal, "Dynamic VAr planning in a large power system using trajectory sensitivities," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 461–469, 2010.

[16] S. Q. Yuan and D. Z. Fang, "Robust PSS parameters design using a trajectory sensitivity approach," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 1011–1018, 2009.

[17] S. M. Baek, J. W. Park, and I. A. Hiskens, "Optimal tuning for linear and nonlinear parameters of power system stabilizers in hybrid system modeling," *IEEE Transactions on Industry Applications*, vol. 45, no. 1, pp. 87–97, 2009.

[18] I. A. Hiskens, M. A. Pai, and T. B. Nguyen, "Bounding uncertainty in power system dynamic simulations," in *2000 IEEE Power Engineering Society, Conference Proceedings*, vol. 2, 2000, pp. 1533–1537.

[19] I. A. Hiskens and J. Alseddiqui, "Sensitivity, approximation, and uncertainty in power system dynamic simulation," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1808–1820, 2006.

[20] I. Hiskens and M. Pai, "Power system applications of trajectory sensitivities," *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.02CH37309)*, vol. 2, no. 4, pp. 1–6, 2002.

[21] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed. Society for Industrial and Applied Mathematics, 2008.

[22] G. Hou and V. Vittal, "Cluster computing-based trajectory sensitivity analysis application to the WECC system," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 502–509, feb 2012.

[23] P. Mitra, V. Vittal, P. Pourbeik, and A. Gaikwad, "Load sensitivity studies in power systems with non-smooth load behavior," *IEEE Transactions on Power Systems*, vol. pp, no. 99, 2016.

[24] Y. Sun, Y. Xinlin, and H. F. Wang, "Approach for optimal power flow with transient stability constraints," *IEE Proceedings-Generation, Transmission and Distribution,*, vol. 151, no. 1, pp. 8–18, jan 2004.

[25] M. Gunzburger, *Perspectives in Flow Control and Optimization*. Society for Industrial and Applied Mathematics, 2002.

[26] L. C. Wilcox, G. Stadler, T. Bui-Thanh, and O. Ghattas, "Discretely exact derivatives for hyperbolic PDE-constrained optimization problems discretized by the discontinuous galerkin method," *Journal of Scientific Computing*, vol. 63, no. 1, pp. 138–162, aug 2014.

[27] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, K. Rupp, B. Smith, S. Zampini, H. Zhang, and H. Zhang, "PETSc Users Manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.7, 2016.

[28] H. Zhang and A. Sandu, "FATODE: a library for forward, adjoint, and tangent linear integration of ODEs," *SIAM Journal on Scientific Computing*, vol. 36, no. 5, pp. C504–C523, oct 2014.

[29] I. A. Hiskens, "Power system modeling for inverse problems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 3, pp. 539–551, 2004.

[30] K. E. Brenan, S. L. V. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1996.

[31] S. Galán, W. F. Feehery, and P. I. Barton, "Parametric sensitivity functions for hybrid discrete/continuous systems," *Applied Numerical Mathematics*, vol. 31, no. 1, pp. 17–47, 1999.

[32] S. Abhyankar, B. Smith, H. Zhang, and A. Flueck, "Using PETSc to develop scalable applications for next-generation power grid," in *Proceedings of the 1st International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. ACM, 2011.

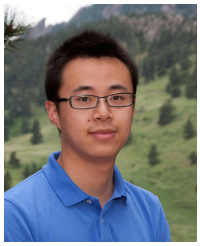
[33] A. Griewank and A. Walther, "Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation," *ACM Transactions on Mathematical Software*, vol. 26, no. 1, pp. 19–45, 2000.

[34] S. Abhyankar, V. Rao, and M. Anitescu, "Dynamic security constrained optimal power flow using finite difference sensitivities," in *2014 IEEE PES General Meeting — Conference & Exposition*. IEEE, 2014, pp. 1–5.

[35] R. Zimmerman and C. Murillo-Sanchez, "MATPOWER 5.1 users manual," Power Systems Engineering Research Center, Tech. Rep., 2015.

[36] P. Sauer and M. A. Pai, *Power System Dynamics and Stability*. Prentice Hall Inc., 1998.

[37] N. Petra, C. Petra, Z. Zhang, E. Constantinescu, and M. Anitescu, "A bayesian approach for parameter estimation with uncertainty for dynamic power systems," *to appear in IEEE Transactions on Power Systems*, 2016.



Hong Zhang received his Ph.D. degree in computer science from Virginia Tech, Blacksburg, in 2014. He is currently a postdoc appointee in the Mathematics and Computer Science Division at Argonne National Laboratory. His research interests include sensitivity analysis, time-stepping algorithms and their applications to power systems. He is a codeveloper of the PETSc library.

Shrirang Abhyankar received his Ph.D. degree (2011) in electrical engineering from Illinois Institute of Technology, Chicago. He is currently a computational engineer in the Energy Sciences Division at Argonne National Laboratory. His research interests include scalable algorithms for large-scale transient stability analysis, combined electromechanical and electromagnetic simulation, and cosimulation of transmission-distribution dynamics. He is a codeveloper of the PETSc library.

Emil M. Constantinescu received his Ph.D. degree in computer science from Virginia Tech, Blacksburg, in 2008. He is currently a computational mathematician in the Mathematics and Computer Science Division at Argonne National Laboratory, and he is on the editorial board of SIAM Journal on Scientific Computing. His research interests include numerical analysis of time-stepping algorithms and their applications to energy systems.

Mihai Anitescu is a senior computational mathematician at Argonne National Laboratory and professor of statistics at the University of Chicago. He obtained his Ph.D. in applied mathematical and computational sciences from the University of Iowa in 1997. He was the recipient of the Wilkinson Fellowship in Scientific Computing in 1997, of a DOE INCITE Award in 2010, of the 2013 COIN-OR cup, and of the 2013 COAP best paper prize award. He specializes in the areas of numerical optimization, computational science, numerical analysis, and uncertainty quantification. He coauthored more than 100 peer-reviewed papers in scholarly journals, book chapters, and conference proceedings. He is on the editorial board of Mathematical Programming A and B, SIAM Journal on Optimization, SIAM Journal on Scientific Computing, and SIAM/ASA Journal in Uncertainty Quantification; and he is a senior editor for Optimization Methods and Software.