AIAA Aviation
16-20 June 2014, Atlanta, GA
15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference

Optimization Algorithm for Systems Governed by Chaotic Dynamics

Anthony Ashley* and Jason E. Hicken†
Rensselaer Polytechnic Institute, Troy, New York, 12180

We describe an algorithm for optimizing time-averaged objective functions that depend on a chaotic state variable. Such problems are ubiquitous in engineering design. They are challenging, because of the sensitive dependence of the state to perturbations in the design. One consequence of this sensitive dependence is that increasing the averaging period, which improves the accuracy of the objective, causes the gradient to diverge. To overcome this issue, the proposed algorithm uses an ensemble objective in a Newton-Krylov trust-region framework. The ensemble objective averages a set of objective functions, each of which uses a reduced time-averaging period and independent state variable; this independence permits each simulation and adjoint computation to be carried out in parallel. The novel aspect of the proposed method is the use of the ensemble objective within a Newton-Krylov algorithm; the latter helps avoid some of the issues presented by objectives governed by chaotic state variables. We demonstrate the proposed ensemble-Newton-Krylov algorithm on an optimization problem governed by the Lorenz dynamical system.

I. Introduction

Many systems and subsystems found in aerospace applications are governed by nonlinear physics that exhibit chaotic behavior; such behavior can be found, for example, in an engine combustor and turbulent boundary layers. These systems are challenging to design, because their dynamics are complex and often defy intuition. This motivates the current work, which investigates optimization tools that can help engineers design systems subject to chaotic processes. By largely automating the design of chaotic systems, the hope is that their performance will improve and their cost will decrease.

Broadly, the proposed approach is a form of simulation-based optimization. In this methodology, the system to be optimized is modeled using a high-fidelity approximation — typically a partial differential equation (PDE) — and this model is then incorporated into an optimization algorithm. The approach is sometimes called PDE-constrained optimization, since it is an optimization subject to a PDE constraint. Our target application is high-fidelity, physics-based multi-disciplinary design optimization (MDO). However, the method proposed should be applicable to a wide range of PDE-constrained optimization problems that arise in engineering design, such as aerodynamic shape optimization.

Aerodynamic shape optimization using the adjoint variables, as pioneered by Jameson [1], is now well understood and mature for systems governed by steady flows; see, for example, Refs. [2–4]. Consequently, during the last decade, researchers have focused on extending the adjoint methodology to unsteady flows. In the context of time-periodic flows, Duta, Giles, and Campobasso [5] developed an adjoint method based on linear harmonic analysis. Similarly, Nadarajah and Jameson [6] extended the adjoint approach for nonlinear frequency-domain methods. These methods are highly efficient, but, since they assume periodicity, they are not suitable for chaotic flows.

Adjoint methods for more general (non-periodic) flows have also been developed; see, for example, Refs. [7–11]. In principle, these methods can be applied to arbitrary unsteady simulations to compute the gradient of an objective; however, as described below, the usefulness of this gradient is questionable when the flow is chaotic and the objective is a time-averaged quantity. Note that most numerical results

*Graduate Student, Department of Mechanical, Aerospace, and Nuclear Engineering, Student Member AIAA
†Assistant Professor, Department of Mechanical, Aerospace, and Nuclear Engineering, Member AIAA

for the (conventional) unsteady adjoint presented in the literature are based on periodic flows or short-time horizons.

Lea, Allen, and Haine [12] were among the first researchers to study sensitivity analysis of chaotic systems. For the Lorenz differential equation (described below), they showed that the standard adjoint approach produces divergent sensitivities as the averaging period tends to infinity. As a potential remedy, they proposed using an ensemble of adjoints [13]: an average of adjoints from problems with distinct initial conditions, but shorter time intervals. This ensemble-adjoint approach prevented divergence, but the resulting gradient was slow to converge to the true gradient. Indeed, a subsequent study by Eyink, Haine, and Lea [14] provided evidence that the ensemble adjoint does not have a finite variance and, consequently, converges slower than a typical Monte Carlo estimate, i.e. slower than $N^{-1/2}$. In the subsequent sections we will show that, despite this slow convergence, the ensemble adjoint still provides useful information.

The key innovation of the proposed method is to synthesize two methods: the ensemble objective and Newton-Krylov optimization. The ensemble objective is the source of the ensemble adjoint investigated by Lea, Allen, and Haine [12]: it is a ensemble of objectives produced by sampling initial conditions. In contrast with previous approaches, the proposed method incorporates the ensemble objective and its sensitivities into a reduced-space Newton-Krylov (NK) algorithm. In this context, reduced-space refers to treating the PDE state variables, e.g. velocity and pressure, as implicit functions of the design variables. We refer to the proposed algorithm as ensemble Newton Krylov, or ENK for short.

Reduced-space NK optimization methods offer excellent algorithmic scaling, because they make use of the reduced Hessian. Historically, the reduced Hessian has been considered too costly and difficult to use in high-fidelity MDO: the cost to form the Hessian is proportional to $m+1$ PDE solves, where m is the number of design variables. This cost does not arise in NK methods, because they do not need to form the Hessian explicitly. Instead, the Krylov iterative method relies on Hessian-vector products, and these products can be formed using two second-order adjoints [15–18]. To the best of our knowledge, the use of second-order adjoints for ensemble objectives in chaotic optimization problems has not been explored previously.

The research most closely related to the current work is that of Wang and his collaborators [19,20]. In [20], Wang, Hu, and Blonigan propose the least-squares sensitivity approach to find gradients of objectives that depend on chaotic differential equations. Their approach relies on the fact that long-time averages for ergodic systems should not depend on the initial condition. Thus, they permit the initial condition to vary as part of the sensitivity calculation. The least-squares sensitivity approach has shown promise, although further investigation is necessary for large-scale PDE applications and high-dimensional design problems.

The paper begins by introducing the Lorenz dynamical system and using this simple system of ODEs to illustrate the difficulties associated with optimizing objectives that depend on chaotic systems. We then discuss possible approaches to deal with these difficulties, and this discussion leads to the the proposed ensemble-Newton-Krylov method. Subsequently, we describe the major steps in the algorithm and finally demonstrate the effectiveness of the ENK method on the Lorenz system. We conclude with a summary and future directions.

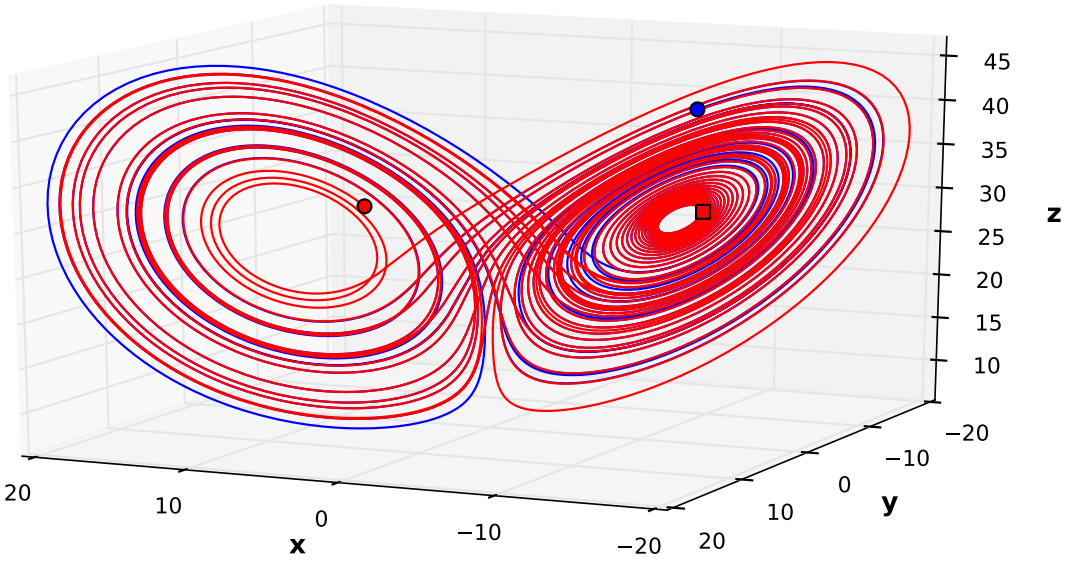
II. The Challenge of Optimizing Chaotic Systems

In order to illustrate the implications of chaotic dynamics for optimization, we follow reference [12] and consider the Lorenz dynamical system. The Lorenz system is a coupled set of nonlinear ordinary differential equations and a canonical example of chaotic behavior. The system is defined by

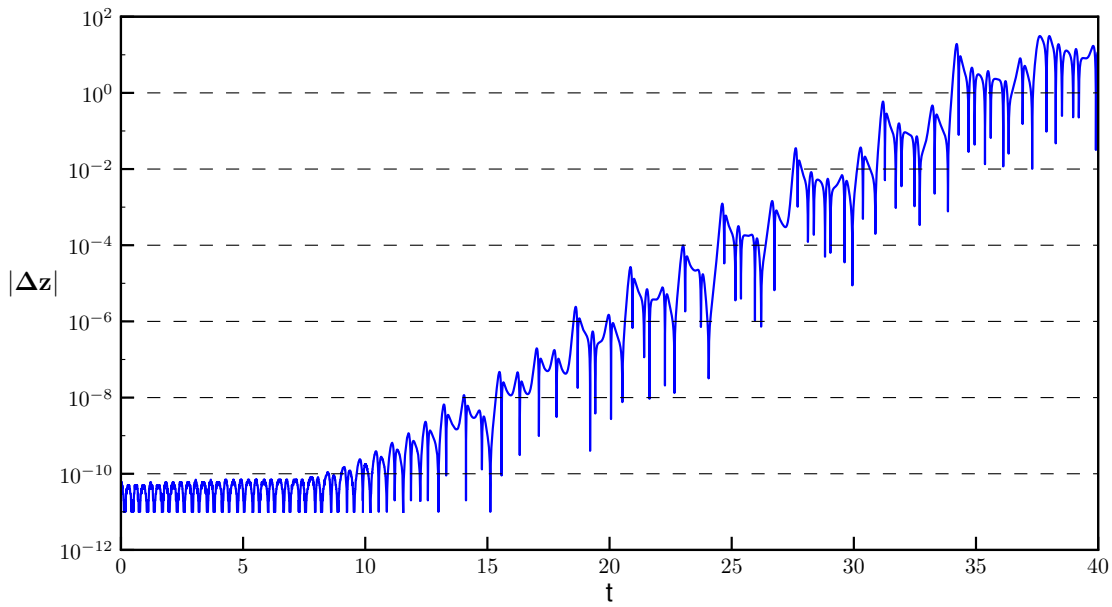
$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sigma(y-x) \\ x(\rho-z)-y \\ xy-\beta z \end{pmatrix}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{1}$$

where $\boldsymbol{x} = (x, y, z)^T$ are the state variables and (σ, ρ, β) are the parameters. Unless stated otherwise, we set $\sigma = 10$ and $\beta = 8/3$ and use ρ as a control variable.

Figure 1 shows two trajectories of the Lorenz system that demonstrate the sensitive nature of the state to its initial condition. Indeed, the initial conditions of the two trajectories satisfy $\|\Delta\boldsymbol{x}_0\| \leq 10^{-10}$, yet their final locations are O(100) units apart. The exponential divergence of the trajectories is clearly observed by plotting the absolute difference in their z coordinates, $|\Delta z(t)|$, which is shown in Figure 1(b). This exponential divergence is characteristic of chaotic systems and has been observed in unsteady fluid simulations; see, for example, [21].



(a) two trajectories of the Lorenz system



(b) difference in z coordinate

Figure 1. A plot of two trajectories obeying the Lorenz system is shown in Figure (a). Despite starting with almost identical initial conditions — denoted by the red square — the trajectories rapidly diverge. The final locations of the two trajectories are marked with the blue and red circles. Figure (b) plots the absolute difference in the two trajectories' z coordinate; note the logarithmic scale along the $|\Delta z|$ axis.

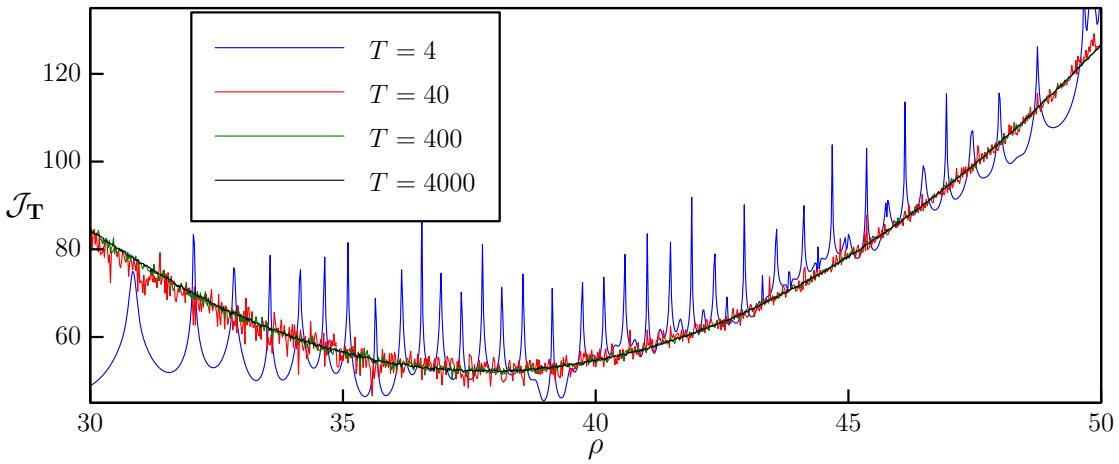


Figure 2. The Lorenz objective $\mathcal{J}_T(\rho)$ versus ρ for increasing values of the integration period T. As T increases, $\mathcal{J}_T$ becomes a better approximation to the ideal objective $\mathcal{J}$, but the "roughness" of $\mathcal{J}_T$ increases.

For our model optimization problem, we consider an "inverse" design in which we attempt to minimize the variance of z with respect to a nominal time-averaged value z_{targ}. The ideal objective is

$$\mathcal{J}(\rho) = \lim_{T\to\infty} \frac{1}{2T} \int_0^T \left(z(t,\rho) - z_{\text{targ}}\right)^2 \, dt,$$

where $z_{\text{targ}} = 35$ is the nominal average. This limit cannot be evaluated in practice, so an approximation must be made. One possibility is to truncate the limit at a finite value of T and define

$$\mathcal{J}_T(\rho) = \frac{1}{2T} \int_0^T \left(z(t,\rho) - z_{\text{targ}}\right)^2 \, dt.$$

Figure 2 plots the approximate objective, $\mathcal{J}_T$, over a range of ρ and four values of T. We observe that $\mathcal{J}_T$ has large fluctuations that reduce as the period T increases. The fluctuations in $\mathcal{J}_T$ reflect the sensitive dependence on the initial conditions; more precisely, a perturbation in the design variable, $\rho + \delta\rho$, causes a perturbation in the state during its early evolution, and this leads to a divergence between the states $z(\rho)$ and $z(\rho + \delta\rho)$.

As T is increased, $\mathcal{J}_T$ converges toward $\mathcal{J}$ and the fluctuations reduce in magnitude; however, the number of fluctuations over a fixed interval of ρ grows as T increases. This growth hints at the difficulty associated with the sensitivity analysis of $\mathcal{J}_T$. Indeed, as $T \to \infty$ the gradient of $\mathcal{J}_T$ diverges [12], and we cannot use conventional sensitivity analysis to achieve any useful reduction in $\mathcal{J}_T$. An alternative approach is necessary.

III. Proposed Approach

Figure 2 suggests that the gradient of $\mathcal{J}$, i.e. the "true" gradient, might be approximated using a finite-difference approximation applied to $\mathcal{J}_T$. This approach has been pursued in the climate modeling community [12], but there are two objections to using finite-difference approximations. The first is the difficulty in finding a suitable step size, which requires trial and error. The second, more serious, objection is that the cost of finite-difference approximations scale with the number of design variables. For high-fidelity MDO problems, this cost is usually prohibitive.

The small-scale fluctuations of $\mathcal{J}_T$ are reminiscent of a noisy or discontinuous function. In light of this observation, we may be tempted to use gradient-free methods to optimize chaotic objectives. Gradient-free methods are suitable for small design problems, but they are computationally expensive when applied to high-dimensional design problems, like those typically encountered in high-fidelity MDO. Conn, Scheinberg, and Vicente [22] indicate that "it is usually not reasonable to try and optimize problems with more than a few tens of variables..." with gradient-free methods. The most efficient gradient-free methods can achieve linear convergence rates [22], but even linear convergence rates are not satisfactory for PDE-constrained optimization.

While gradient-free methods are not appropriate in our context, they do offer some guidance on how we might proceed. Many gradient-free methods construct a surrogate model — sometimes called a response surface — of the objective function, and use this surrogate to guide the optimization. This is the inspiration for the method we pursue.

The challenge with using a surrogate model is the potential cost. For example, a high-fidelity optimization involving 10^4 design variables would require on the order of 5×10^7 PDE simulations to build a simple quadratic model by sampling the function. For "well-behaved" smooth functions, we can reduce this cost significantly by sampling the gradient, as is done in quasi-Newton methods like the Broyden-Fletcher-Goldberg-Shanno (BFGS) update [23]. Unfortunately, as we have seen, the objective $\mathcal{J}_T$ in our target applications is not well-behaved, so we cannot use derivative information to build a surrogate for this objective.

Rather than building a surrogate of $\mathcal{J}_T$ directly, the proposed method uses a surrogate based on a modified objective function. For the modified objective function, we will use an ensemble objective [12], which consists of an ensemble of time-averaged objective functions:

$$\langle \mathcal{J} \rangle \equiv \frac{1}{N} \sum_{i=1}^{N} \mathcal{J}_{i,\tau}. \tag{2}$$

For example, for the Lorenz objective we would replace $\mathcal{J}_T$ with

$$\langle \mathcal{J} \rangle(\rho) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2\tau} \int_0^\tau \left(z_i(t,\rho) - z_{\text{targ}} \right)^2 \, dt. \tag{3}$$

The subscript i indicates that each $\mathcal{J}_{i,\tau}$ uses an independent state produced by an *unique initial condition*. Moreover, note that the individual objectives that form the ensemble $\langle \mathcal{J} \rangle$ use a shortened time domain $\tau < T$. This reduced time domain has important implications.

- If τ is sufficiently small, two states whose initial conditions are nearby will not diverge significantly as they evolve, and, consequently, the adjoint of $\mathcal{J}_{i,\tau}$ will be well behaved. This permits the construction of potentially useful gradient and Hessian information for $\mathcal{J}_{i,\tau}$ and, consequently, $\langle \mathcal{J} \rangle$.
- If $T = N\tau$, for example, the cost of each $\mathcal{J}_{i,\tau}$ in the ensemble objective will be N times cheaper than the objective $\mathcal{J}_T$. Although there are N times as many of these objectives, the states that they depend on are independent; therefore, the objectives $\mathcal{J}_{i,\tau}$ — *and their gradients* — can be computed in parallel providing a significant reduction in time.

We emphasize that the ensemble objective, while differentiable, does not eliminate the fluctuations observed in $\mathcal{J}_T$. This apparent contradiction is explained by the so-called spin-up period. When the design variables are updated, the system should be run for an initial spin-up period before gathering statistics; this ensures that the statistics reflect the long-time behavior of the system. In contrast, when computing the sensitivities of the ensemble objective, the initial conditions are frozen and the spin-up period is ignored.

A drawback of the ensemble objective is that its gradient is slow to converge to the true gradient. For example, for the Lorenz system based on $\tau = 2$, Eyink *et al.* [14] estimate that $N \approx 6.4 \times 10^{14}$ samples would be required for 1% accuracy in the derivative; clearly this is impractical. Fortunately, the Lorenz problem is not completely analogous to the applications we have in mind, and we believe the estimates in reference [14] may be overly pessimistic in this context.

One reason to be optimistic is that aerodynamic forces and moments are space-time integrals. The gradient of the Lorenz ensemble objective is slow to converge, because of rare samples, i.e. rare trajectories, that differ significantly from the mean. Rare samples such as these are likely to exist at each spatial location in a (turbulent) flow; however, spatial averaging will effectively increase the size of the ensemble and, hopefully, reduce the influence of these rare events.

In addition, while the ensemble objective may not be able to find the true optimum of chaotic problems in the strict mathematical sense, we hypothesize that the ensemble objective is sufficiently accurate to improve the baseline design for engineering purposes. This hypothesis is confirmed below in the case of the Lorenz problem.

III.A. Building the Surrogate Model: Newton-Krylov and the ensemble objective

This section describes how the surrogate model for chaotic objectives is constructed; however, before proceeding, we need to introduce some notation. In the optimization community x is often used to denote the (design) variables, while in the PDE community x denotes a spatial coordinate. To avoid confusion, we will use $\chi \in \mathbb{R}^m$ to represent the vector of design/control variables. The physics will be modeled using a discretized PDE, which, in residual form, is given by

$$\mathcal{R}_i(\chi, u_i) = 0, \qquad \forall i = 1, 2, \ldots, N,$$

where $u_i \in \mathbb{R}^n$ is the discrete state, or primal, solution corresponding to the initial condition $u_{0,i}$. The subscript i on $\mathcal{R}_i$ reflects the fact that the initial conditions are also incorporated into the discrete residual.

The objective for the optimization problem is the ensemble objective, $\langle \mathcal{J} \rangle$, defined earlier. Thus, our optimization problem can be stated succinctly as

$$\begin{aligned} &\text{minimize} && \langle \mathcal{J} \rangle(\chi) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{J}_{i,\tau}\left(\chi, u_i(\chi)\right), \\ &\text{with respect to} && \chi \in \mathbb{R}^m. \end{aligned} \tag{Chaos-Opt}$$

Note that Chaos-Opt is formulated as a reduced-space problem, since we have written the state as a function of the control. Furthermore, Chaos-Opt is not completely general, since constraints are not included; constraints will be addressed in future work.

The ensemble objective $\langle \mathcal{J} \rangle$ requires a surrogate, because, while differentiable for fixed initial conditions, it remains a complex, nonlinear function of (potentially) thousands of design variables. The proposed surrogate model is a quadratic approximation of the ensemble objective that is used in a trust-region framework to determine a design update. In other words, the design update will be given by the solution to the following trust-region subproblem:

$$\min_{p} \quad m_k(p) = \langle \mathcal{J} \rangle(\chi_k) + g^T p + \frac{1}{2} p^T \mathsf{H} p, \qquad \text{s.t.} \quad \|p\| \leq \Delta,$$

where $g \equiv \nabla_\chi \langle \mathcal{J} \rangle(\chi_k)$ is the gradient of the ensemble objective and $\mathsf{H} \equiv \nabla^2_{\chi\chi} \langle \mathcal{J} \rangle(\chi_k)$ is its Hessian. The efficient computation of these derivatives, or their action, is a critical aspect of the proposed approach; thus, the remainder of this section is devoted to explaining how these computations will be performed.

The gradient g will be computed efficiently using the conventional adjoint approach applied to each $\mathcal{J}_{i,\tau}$ individually. Indeed,

$$g = \nabla_\chi \langle \mathcal{J} \rangle = \frac{1}{N} \sum_{i=1}^{N} \nabla_\chi \mathcal{J}_{i,\tau},$$

where each $\nabla_\chi \mathcal{J}_{i,\tau}$ can be evaluated using *independent* adjoints $\psi_i \in \mathbb{R}^n$, which are the solution of

$$\mathcal{S}_i(\chi, u_i, \psi_i) = \left[\nabla_{u_i} \mathcal{J}_{i,\tau}(\chi, u_i)\right]^T + \left[\nabla_{u_i} \mathcal{R}_i(\chi, u_i)\right]^T \psi_i = 0.$$

The adjoints are independent because the states u_i and objectives $\mathcal{J}_{i,\tau}$ are independent. This independence can be exploited to compute the gradients $\nabla_\chi \mathcal{J}_{i,\tau}$ in parallel with no communication, i.e. a so-called "embarrassingly parallel" computation.

In addition to the gradient, the Hessian of $\langle \mathcal{J} \rangle$ also appears in the trust-region subproblem. In engineering literature, the Hessian is often approximated using quasi-Newton methods like BFGS. The use of quasi-Newton methods is motivated by the perceived cost and difficulty of evaluating second-order derivatives. In particular, computing the (reduced) Hessian explicitly requires the solution of $m + 1$ linear systems the size of the original PDE [24, 25] for each optimization iteration.

Despite their popularity, we anticipate two potential issues with using quasi-Newton methods in the optimization of high-dimensional chaotic systems. First, the initial quasi-Newton approximation to the Hessian is typically the identity matrix, or a scalar multiple of it. This leads to a steepest descent direction on the first iteration with no information regarding the curvature of $\langle \mathcal{J} \rangle$. Figure 3 illustrates why this steepest descent direction might cause problems. The optimization algorithm, using either a line-search or trust-region approach, must determine how far to step along this direction. If the function is smooth, a

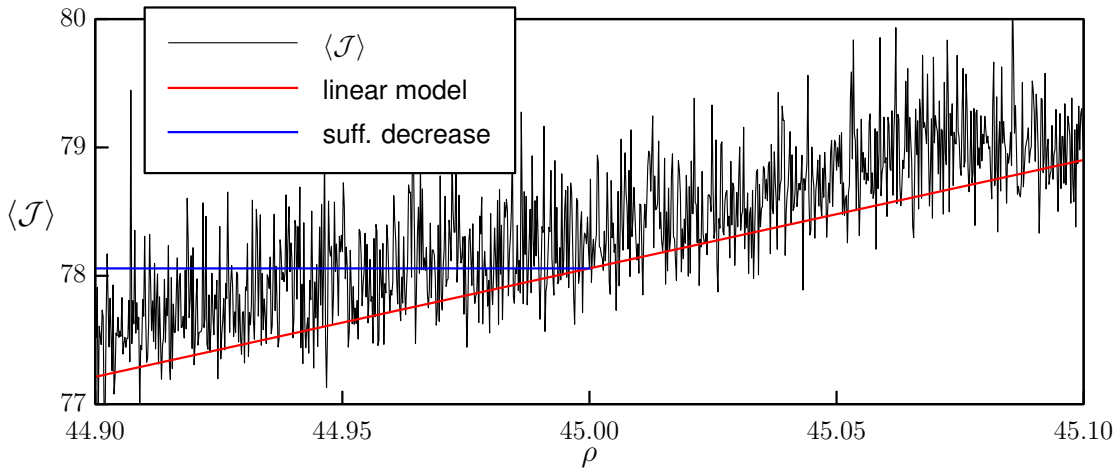


Figure 3. Linear model (i.e. steepest descent) of the objective $\langle \mathcal{J} \rangle$ and the sufficient decrease condition for $c_1 = 10^{-4}$. In this example, the linear model at $\rho = 29$ is based on a value of $\langle \mathcal{J} \rangle$ that is low relative to the true mean; consequently, the sufficient decrease condition may be difficult to satisfy, because many fluctuations of $\langle \mathcal{J} \rangle$ are above this line.

line-search approach can safely use the following sufficient-decrease condition [23] to accept or reject a step length:

$$\mathcal{J}(\chi_k + \alpha p) \leq \mathcal{J}(\chi_k) + c_1 g^T p,$$

where p is the step direction, α is the step length, and $c_1 \in (0, 1)$ is a parameter. In practice c_1 is chosen to be relatively small, e.g. $c_1 = 10^{-4}$, which is why the sufficient decrease condition appears as a horizontal line in Figure 3. For chaotic problems, like the Lorenz problem, the sufficient-decrease condition will not be reliable, because even a small step may be rejected. Similar problems will arise in a trust-region framework if the trust radius is too small initially.

The second serious issue with quasi-Newton methods is scalability. Conventional optimization algorithms frequently resort to limited-memory BFGS for large-scale problems [26–28], and this leads to linear convergence rates [23, 29]. This means that the number of optimization iterations grows linearly with the number of design variables. Such linear growth is unacceptable for high-dimensional design problems with costly objective functions and gradients, whether they are based on chaotic systems or not.

The proposed approach seeks to ameliorate both of these issues by retaining the Hessian in the trust-region subproblem. The Hessian provides curvature information, even during the first optimization iteration, and this reduces the likelihood that a small step length will be chosen and subsequently rejected because of fluctuations in $\langle \mathcal{J} \rangle$. Issues with algorithmic scalability are also reduced, because, by including the Hessian, the underlying optimization algorithm becomes Newton's method. We emphasize that the asymptotic convergence rate of Newton's method will not be quadratic in this case, given the fluctuations in $\langle \mathcal{J} \rangle$ and its derivatives. Nevertheless, second-order information for the ensemble objective should help the optimization algorithm make significant progress toward a local minimum of the ideal objective $\mathcal{J}$. This is demonstrated for the Lorenz problem below.

To further justify the use of the Hessian in the trust-region framework, we must address the computational cost of including this second-order information. The key is to recognize that we do not need to compute H explicitly. If we solve the trust-region subproblem using a Hessian-free Newton-Krylov approach [30], we need only provide products of the form $\mathsf{H}w$, where $w \in \mathbb{R}^m$ is an arbitrary vector in the design space. Unlike the explicit Hessian, which requires the solution of $m+1$ PDEs, these Hessian-vector products can be evaluated using only two second-order adjoint PDEs [15–18, 31]. The general form of the equations governing these second-order adjoints is

$$\left(\nabla_{u_i} \mathcal{R}_i\right) z_i = -\left(\nabla_{\psi_i} g^T\right) w, \tag{4}$$

$$\left(\nabla_{u_i} \mathcal{R}_i\right)^T \lambda_i = -\left(\nabla_{u_i} g^T\right) w - \left(\nabla_{u_i} \mathcal{S}^T\right) z_i, \tag{5}$$

where $z_i \in \mathbb{R}^n$ and $\lambda_i \in \mathbb{R}^n$ denote the second-order adjoints for the i^{th} objective $\mathcal{J}_{i,\tau}$. See Ref. [31] for additional details on the solution of equations (4) and (5).

As with the state u_i and first-order adjoint ψ_i, the second-order adjoints can be computed independently. More precisely, while the z_i must be solved before the λ_i, each (z_i, λ_i) pair is independent and can be solved in parallel. This is vital to the scalability of the proposed approach.

Once the z_i and λ_i have been determined, the desired Hessian-vector product can be computed from[a]

$$\mathsf{H}w = \nabla_\chi \left(g^T w\right) + \sum_{i=1}^{N} \left[\left(\nabla_\chi \mathcal{R}\right)^T \lambda_i + \left(\nabla_\chi \mathcal{S}\right)^T z_i\right].$$

This final product can be evaluated using an all-reduce operation across the N terms.

III.B. The Ensemble-Newton-Krylov Algorithm

The proposed method for chaotic-PDE-constrained optimization is outlined in Algorithm 1. The algorithm is essentially a Newton-Krylov method applied to the ensemble objective; hence the name ensemble Newton Krylov (ENK). In the list below, we describe the major steps in the proposed ENK algorithm.

Initialization: Lines 1 and 2 are the initialization phase of the algorithm. Using the initial design χ_0, the ensemble of states is computed. Each state u_i will be based on a unique initial condition, and the PDE solver will be run for an initial "spin-up" period to reach stationarity. For example, the solver may be run for 3τ time units, with only the last τ units used to define the ensemble objective. Each u_i can be computed independently in parallel if sufficient resources exist.

Gradient and Convergence: The ensemble of first-order adjoints is computed on line 4 followed by the gradient evaluation on line 5. Convergence criteria are checked on line 6. In the optimization of deterministic objective functions, we would typically terminate the algorithm when the norm of the gradient is reduced below a specified tolerance. In the present context the gradient, g, of the ensemble objective is not necessarily an accurate representation of the gradient of the ideal objective $\mathcal{J}$. Appropriate convergence criterion for ENK remains an outstanding issue that we continue to investigate.

Trust-Region Subproblem: If the algorithm has not converged, a candidate step p is determined by solving the trust-region subproblem on line 7. This subproblem will be solved using a Krylov iterative solver like Steihaug-Toint CG, with the required Hessian-vector products computed using second-order adjoints.

Provisional Update: The design variables are provisionally updated to χ_{k+1} in line 8, and the states u_i are updated to reflect this new design. Since χ_{k+1} will alter the dynamics of the system, the PDE solver must once again be run for a spin-up period (line 9) before computing the states u_i (line 10).

Check Actual Reduction: The ratio of the actual reduction in the objective to the predicted objective reduction, r_k, is computed on line 11. If this ratio is deemed too small or negative (line 12), the quadratic surrogate did not model the objective well over the radius Δ; in this case we revert to the previous design on line 13.

Update Trust-Region Radius: Line 15 begins a standard update of the trust-region radius based on r_k. If the trust-region constraint is active and the ratio r_k is sufficiently large, then the trust-region constraint is hindering progress and the radius Δ is increased. Conversely, if the ratio is small or negative, then the trust-region radius is decreased. See, for example, Ref. [23].

IV. Results

We use the Lorenz model problem to illustrate and investigate the ENK algorithm. Although this model problem is small — only three state variables and at most three parameters — it exhibits the characteristics of large-scale chaotic processes.

For the following two examples, the Hessian is evaluated explicitly using finite-differencing applied to the gradient. A Krylov iterative method is not necessary here and would converge in at most m iterations anyway, where m is the dimension of the design space.

[a] If memory is an issue, the contribution of z_i and λ_i to the Hessian-vector product can also be computed "on-the-fly."

Algorithm 1: Ensemble Newton Krylov (ENK) optimization algorithm

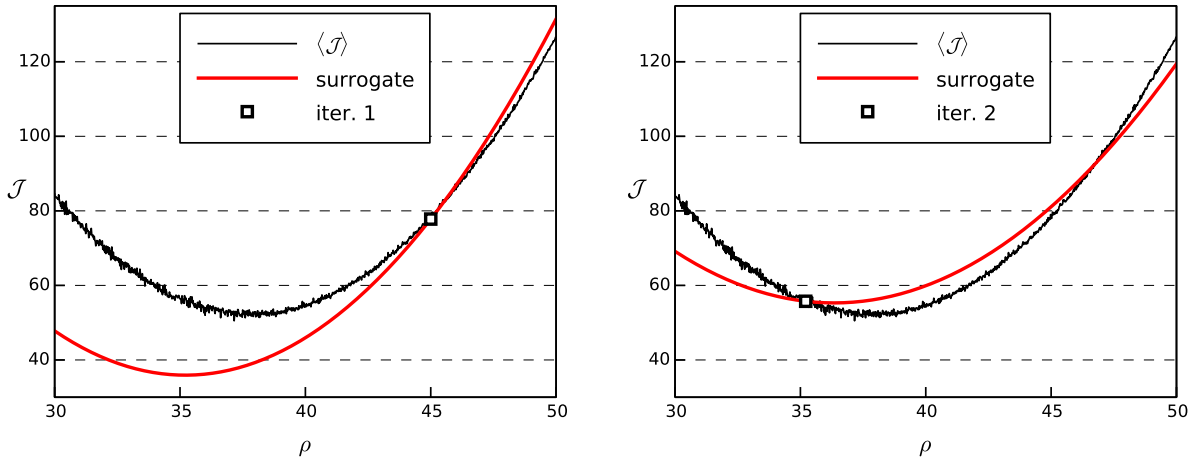
choose an initial design χ_0
solve for states, $\{u_i(\chi_0)\}_{i=1}^N$
for $k = 0, 1, 2, \ldots$ **do**
    solve for the (first-order) adjoints, $\{\psi_i(\chi_{k+1})\}_{i=1}^N$
    compute reduced gradient, g, of $\langle\mathcal{J}\rangle$
    (exit if convergence criteria passed)
    Using, e.g. Steihaug-Toint CG, solve the trust-region subproblem:

$$\min_p \quad m_k(p) = \langle\mathcal{J}\rangle(\chi_k) + g^T p + \frac{1}{2} p^T \mathsf{H} p, \qquad \text{s.t.} \quad \|p\| \leq \Delta$$

    set provisional update $\chi_{k+1} = \chi_k + p$
    run solvers for a spin-up period τ_{spin}
    solve for (independent) states $\{u_i(\chi_{k+1})\}_{i=1}^N$
    compute the ratio of the actual reduction to the predicted reduction:

$$r_k = \frac{\langle\mathcal{J}\rangle(\chi_k) - \langle\mathcal{J}\rangle(\chi_{k+1})}{m_k(0) - m_k(p)}$$

    if $r_k < \eta$ **then** (update is unacceptable)
        revert to previous design: $\chi_{k+1} = \chi_k$
    end
    if $r_k < \frac{1}{4}$ **then** (update trust-region radius)
        $\Delta \leftarrow \frac{1}{4}\Delta$
    else
        if $r_k > \frac{3}{4}$ *and* $\|p\| = \Delta$ **then**
            $\Delta \leftarrow \min(2\Delta, \Delta_{\max})$.
        end
    end
    if $\Delta < \Delta_{\min}$ **then**
        exit
    end
end



(a) Model at iteration 1

(b) Model at iteration 2

Figure 4. ENK iterations applied to the Lorenz model problem. In the first iteration, ρ is updated based the minimum of the quadratic model. The new quadratic model is then built at this updated location for the second iteration.

IV.A. One-dimensional Design Problem

We begin by considering the one-dimensional problem of minimizing the objective (3). Figure 4 depicts two iterations of the ENK method beginning at $\rho_0 = 45$. At each iteration, the quadratic model at ρ_k overlays the ensemble objective $\langle \mathcal{J} \rangle$, computed here using $N = 800$ and $\tau = 0.5$; thus, the ensemble objective has the same cost as using $\mathcal{J}_T$ with $T = 400$. Despite errors in the quadratic model, ENK is able to improve the objective value.

To investigate the average behavior of ENK on this problem, we applied the algorithm to 100 randomly generated initial guesses ρ_0 selected from the interval $[25, 75]$. The initial trust radius was set to $\Delta_0 = 10$ and the maximum trust radius was set to $\Delta_{\max} = 20$. A spin-up integration period of $\tau_{\text{spin}} = 1.0$ was adopted for each sample. Computational cost of each iteration is measured using integration time; for example, the cost of solving for the state variables is the number of samples times the integration period of each sample, $N\tau = 400$, and, similarly, the cost of the spin-up is $N\tau_{\text{spin}} = 800$.

Figure 5(a) plots the value of the design variable ρ versus the cost metric. The red line is an estimate of $\mathsf{argmin}_\rho \mathcal{J}(\rho)$ based on the minimum value observed from the $T = 4000$ data (see Figure 2). Figure 5(b) is a similar plot of the objective versus the cost; note that a logarithmic scale is used for $\langle \mathcal{J} \rangle$. In all cases, the algorithm terminates because the minimum allowable trust radius, $\Delta_{\min} = \Delta_0(0.5^9) \approx 0.0195$, was reached.

In general, the ENK algorithm improves the objective and moves the design toward the optimal value. However, there is consistent bias in the "optimal" value of ρ determined by the algorithm, due to the errors in the ensemble gradient and Hessian.

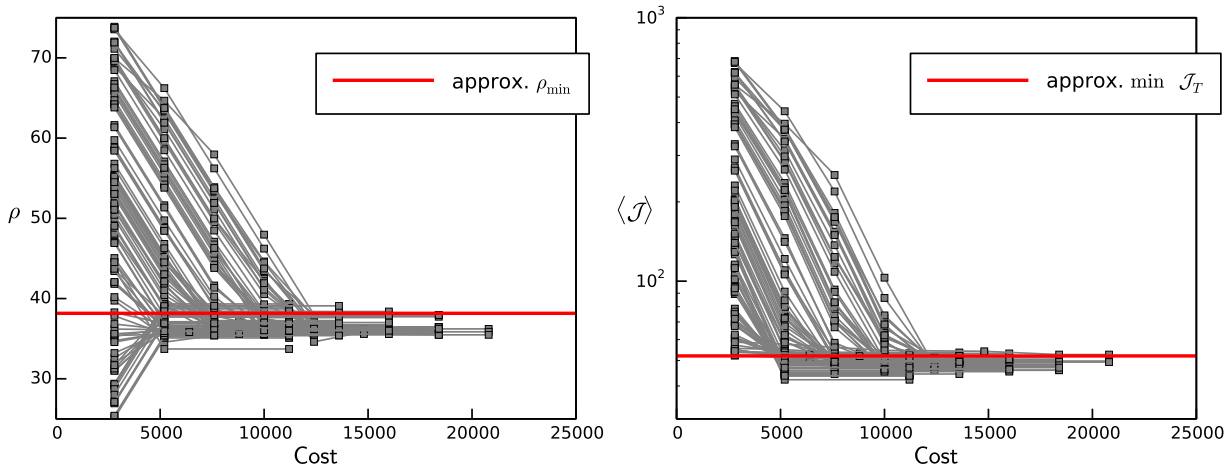
IV.B. Two-dimensional Design Problem

A two-dimensional design problem was also considered, in which both ρ and β were the design variables. The objective function, (6), was similar to the one-dimensional case; however, it was necessary to include an additional term, $20/\beta$, to ensure the existence of a local optimum.

$$\langle \mathcal{J} \rangle(\rho, \beta) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2\tau} \int_0^\tau \left(z_i(t, \rho, \beta) - z_{\text{targ}} \right)^2 \, dt + \frac{20}{\beta}, \tag{6}$$

The ENK method was applied to this two-dimensional objective. Parameters for the ENK algorithm 1 remained similar between the one-dimensional and the two-dimensional case. The starting trust radius was $\Delta_0 = 1.0$, while the maximum trust radius was $\Delta_{\max} = 3.0$. One notable difference was the use of a scaled infinite norm to define the trust radius. Specifically, we used the norm

$$\|p\| \equiv \max\left(5|\rho|, |\beta|\right).$$



(a) ρ versus Cost

(b) $\langle \mathcal{J} \rangle$ versus Cost

Figure 5. Progress of the iterates from 100 randomly initialized starting values for ρ.

The scaling was necessary to account for the disparity in the curvatures in the ρ and β directions.

The application of the ENK method to the two-dimensional objective function is illustrated in Figure 6. These results were generated with $N = 800$ number of intervals and an interval length of $\tau = 0.5$. A number of starting points were selected for the application of the ENK algorithm in the interval $\rho = [30, 50]$ and $\beta = [0.25, 5.0]$. Similar to the one-dimensional case, the ENK algorithm successfully moves the iterates toward the optimal value. However, nearly all cases fail to converge near the approximate minimum of $(\rho, \beta) = (37.68, 1.61)$.

V. Summary and Conclusion

Many engineering design problems seek to optimize an objective that is a time-averaged quantity. Such optimization problems are particularly challenging when the time-averaged objective depends on state variables governed by chaotic dynamical systems, because the conventional adjoint evaluation of the gradient — which is the preferred approach for large-scale design problems — diverges as the averaging period tends to infinity.

To address this issue, we have proposed the ensemble-Newton-Krylov optimization algorithm. This algorithm replaces the single time-averaged objective with an ensemble of time-averaged objectives that use shorter averaging periods and distinct initial conditions. The use of distinct initial conditions means that the state variable for each objective in the ensemble is independent, as are any adjoints that are needed. This leads to a highly parallel and scalable method.

Our preliminary results suggest that the proposed ensemble-Newton-Krylov algorithm yields effective steps that improve the objective. However, while the gradient of the ensemble objective is highly parallelizable, it is slow to converge to the gradient of the infinite-period objective. This results in inaccurate derivative information that contaminates the surrogate model. Future research will focus on improving the quality of the surrogate model and determining appropriate convergence criteria for the ENK algorithm.

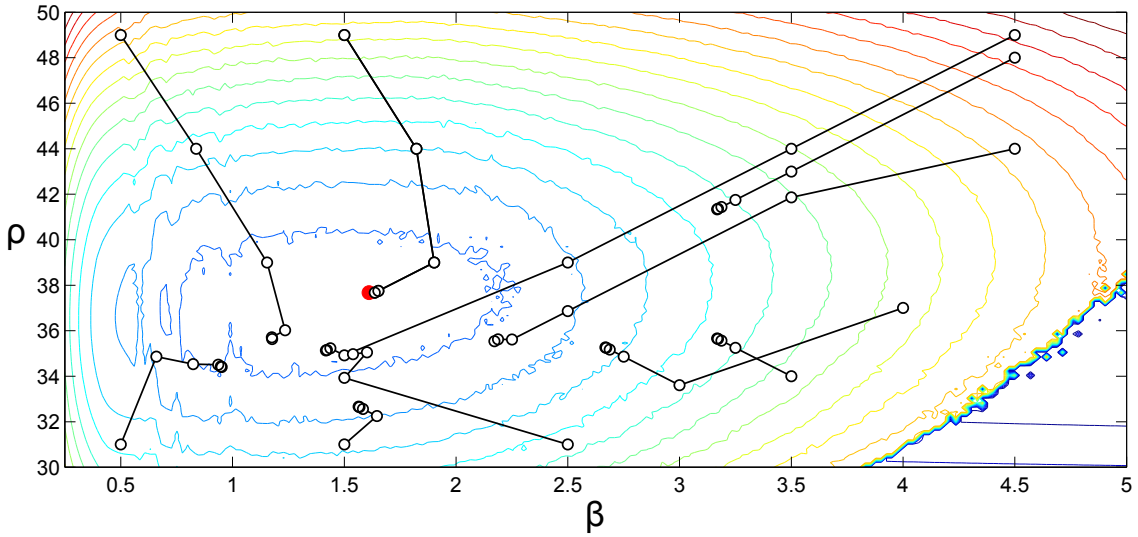


Figure 6. Contour plot of the ensemble objective $\langle \mathcal{J} \rangle(\rho, \beta)$. The approximate location of the minimum, denoted by the red marker, is $(\rho, \beta) = (37.68, 1.61)$. Several optimization sequences, each with a different starting value for the design variables, are plotted. Most sequences fail to reach the approximate minimum, although all sequences improve upon their initial objective value.

References

[1]Jameson, A., "Aerodynamic design via control theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260.

[2]Anderson, W. K. and Bonhaus, D. L., "Airfoil design on unstructured grids for turbulent flows," *AIAA Journal*, Vol. 37, No. 2, Feb. 1999, pp. 185–191.

[3]Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., "Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1," *AIAA Journal*, Vol. 36, No. 1, Jan. 1999, pp. 51–60.

[4]Nemec, M. and Zingg, D. W., "Newton-Krylov algorithm for aerodynamic design using the Navier-Stokes equations," *AIAA Journal*, Vol. 40, No. 6, June 2002, pp. 1146–1154.

[5]Duta, M. C., Giles, M. B., and Campobasso, M. S., "The harmonic adjoint approach to unsteady turbomachinery design," *Int. J. Numer. Meth. Fluids*, Vol. 40, No. 3-4, Sept. 2002, pp. 323–332.

[6]Nadarajah, S. and Jameson, A., "Optimum Shape Design for Unsteady Three-Dimensional Viscous Flows Using a Nonlinear Frequency-Domain Method," *Journal of Aircraft*, Vol. 44, No. 5, Sept. 2007, pp. 1513–1527.

[7]Bewley, T. R., Moin, P., and Temam, R., "DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms," *Journal of Fluid Mechanics*, Vol. 447, No. -1, Nov. 2001, pp. 179–225.

[8]Nadarajah, S. and Jameson, A., "Optimal Control of Unsteady Flows Using a Time Accurate Method," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics, Sept. 2002.

[9]Mani, K. and Mavriplis, D. J., "Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes," *AIAA Journal*, Vol. 46, No. 6, June 2008, pp. 1351–1364.

[10]Rumpfkeil, M. P. and Zingg, D. W., "The optimal control of unsteady flows with a discrete adjoint method," *Optimization and Engineering*, Vol. 11, No. 1, 2010, pp. 5–22.

[11]Nielsen, E. J. and Diskin, B., "Discrete Adjoint-Based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids," *AIAA Journal*, Vol. 51, No. 6, April 2013, pp. 1355–1373.

[12]Lea, D. J., Allen, M. R., and Haine, T. W. N., "Sensitivity analysis of the climate of a chaotic system," *Tellus A*, Vol. 52, No. 5, Oct. 2000, pp. 523–532.

[13]Corti, S. and Palmer, T. N., "Sensitivity analysis of atmospheric low-frequency variability," *Q.J.R. Meteorol. Soc.*, Vol. 123, No. 544, Oct. 1997, pp. 2425–2447.

[14]Eyink, G. L., Haine, T. W. N., and Lea, D. J., "Ruelle's linear response formula, ensemble adjoint schemes and Lévy flights," *Nonlinearity*, Vol. 17, No. 5, Sept. 2004, pp. 1867+.

[15]Wang, Z., Navon, I. M., Dimet, F. X., and Zou, X., "The second order adjoint analysis: Theory and applications," *Meteorology and Atmospheric Physics*, Vol. 50, 1992, pp. 3–20.

[16]Akçelik, V., Biros, G., Ghattas, O., Hill, J., Keyes, D., and van Bloemen Waanders, B., "Parallel Algorithms for PDE-Constrained Optimization," *Parallel Processing for Scientific Computing*, edited by M. A. Heroux, P. Raghavan, and H. D. Simon, chap. 16, Society for Industrial and Applied Mathematics, Jan. 2006, pp. 291–322.

[17]Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S., *Optimization with PDE constraints*, Springer, 2009.

[18]Borzì, A. and Schulz, V., *Computational Optimization of Systems Governed by Partial Differential Equations*, Society for Industrial and Applied Mathematics, Jan. 2011.

[19]Wang, Q., "Forward and adjoint sensitivity computation of chaotic dynamical systems," *Journal of Computational Physics*, Vol. 235, Feb. 2013, pp. 1–13.

[20]Wang, Q., Hu, R., and Blonigan, P., "Sensitivity computation of periodic and chaotic limit cycle oscillations," Aug. 2013, arXiv:1204.0159v4.

[21]Lohner, R., Britto, D., Michailski, A., and Haug, E., "Butterfly-Effect for Massively Separated Flows," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, Jan. 2013.

[22]Conn, A. R., Scheinberg, K., and Vicente, L. N., *Introduction to Derivative-Free Optimization*, Society for Industrial and Applied Mathematics, Jan. 2009.

[23]Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer–Verlag, Berlin, Germany, 2nd ed., 2006.

[24]Griewank, A. and Walther, A., *Evaluating derivatives: principles and techniques of algorithmic differentiation*, Society for Industrial and Applied Mathematics, 2008.

[25]Ghate, D. and Giles, M., "Efficient Hessian Calculation Using Automatic Differentiation," *25th AIAA Applied Aerodynamics Conference*, No. AIAA–2007–4059, Miami, Florida, United States, June 2007.

[26]Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J., "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 23, No. 4, 1997, pp. 550–560.

[27]Wächter, A. and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, Vol. 106, No. 1, May 2006, pp. 25–57.

[28]Waltz, R. A. and Nocedal, J., "KNITRO user's manual," *Northwestern University, Evanston, Illinois, Technical Report OTC-2003/5*, 2003.

[29]Liu, D. C. and Nocedal, J., "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, Vol. 45, 1989, pp. 503–528.

[30]Steihaug, T., "The Conjugate Gradient Method and Trust Regions in Large Scale Optimization," *SIAM Journal on Numerical Analysis*, Vol. 20, No. 3, June 1983, pp. 626–637.

[31]Hicken, J., "Inexact Hessian-vector products in reduced-space differential-equation constrained optimization," *Optimization and Engineering*, accepted, 2014, pp. 1–34.