# An Inexact General Descent Method with Applications in Differential Equation-Constrained Optimization

Humberto Gimenes Macedo[1,2*] and Luís Felipe Bueno[1]

[1]Universidade Federal de São Paulo (UNIFESP), Av. Cesare Mansueto Giulio Lattes, 1201 — Eugênio de Melo, São José dos Campos, SP, Brazil.
[2]Instituto Tecnológico de Aeronáutica (ITA), Praça Marechal Eduardo Gomes, 50 — Vila das Acácias, São José dos Campos, SP, Brazil.

*Corresponding author(s). E-mail(s): gimenes.humberto@unifesp.br;
Contributing authors: lfelipebueno@gmail.com;

## Abstract

In many applications, gradient evaluations are inherently approximate, motivating the development of optimization methods that remain reliable under inexact first-order information. A common strategy in this context is adaptive evaluation, whereby coarse gradients are used in early iterations and refined near a minimizer. This is particularly relevant in differential equation–constrained optimization (DECO), where discrete adjoint gradients depend on iterative solvers. Motivated by DECO applications, we propose an inexact general descent framework and establish its global convergence theory under two step-size regimes. For bounded step sizes, the analysis assumes that the error tolerance in the computed gradient is proportional to its norm, whereas for diminishing step sizes, the tolerance sequence is required to be summable. The framework is implemented through inexact gradient descent and an inexact BFGS-like method, whose performance is demonstrated on a second-order ODE inverse problem and a two-dimensional Laplace inverse problem using discrete adjoint gradients with adaptive accuracy. Across these examples, adaptive inexact gradients consistently reduced optimization time relative to fixed tight tolerances, while incorporating curvature information further improved overall efficiency.

**Keywords:** Inexact optimization, Gradient-based methods, DE-constrained optimization, Discrete adjoint method

# 1 Introduction

Conventionally, convergence analyses of gradient-based algorithms assume the availability of exact information, encompassing both zeroth-order and first-order information. Nevertheless, in various settings, the acquisition of such information is subject to error. These errors can arise, for example, when the objective function lacks an analytic representation, exhibits stochasticity, is affected by finite-precision errors from floating-point arithmetic, or constitutes a smoothed approximation of an inherently non-smooth function. Moreover, errors can also stem from the tolerance-limited accuracy of iterative methods used to compute functional values, gradients, or broader search directions. This phenomenon is especially prevalent in differential equation-constrained optimization (DECO) problems. In this case, the objective function evaluation requires one or more numerical simulations, typically implemented through iterative procedures. Likewise, when the discrete adjoint method is used, computing the gradient normally involves the iterative solution of the adjoint equation. Consequently, in such contexts, both the objective function and its gradient are routinely subject to inexactness. Viewed more broadly, inexactness in numerical analysis arises along two complementary channels, namely (a) inexact solves, where subproblems or linear and nonlinear systems are solved only approximately, and (b) inexact information, where objective, constraint, and/or their gradients evaluations are themselves approximate.

Research on nonlinear equations has predominantly focused on channel (a). Foundational work introduced and developed inexact Newton methods to reduce the computational cost associated with solving Newton equations, showing that exact solves are often unnecessary when the current iterate is far from the solution [1–3]. This line of study has since been extended to quasi-Newton methods, where Hessian approximations replace the exact Jacobian and the resulting system is solved only approximately. Applications of such techniques to sparse nonlinear systems of equations have been explored [4], and subsequent studies have addressed issues of global convergence and constraints such as non-negativity [5, 6]. Recent investigations continue to advance the theory and practice of such inexact methods [7, 8].

From an optimization perspective, both channels (a) and (b) have been systematically investigated. Natural extensions of inexact Newton-type techniques to optimization arise from viewing first-order optimality systems as nonlinear equations [9]. In unconstrained problems, this leads to inexact Newton and quasi-Newton schemes, while for constrained problems, it gives rise to inexact Sequential Quadratic Programming (SQP) methods. Among these, inexact SQP methods have been extensively studied due to their broad applicability across diverse classes of problems. These methods interpret the resolution of first-order optimality conditions as a sequence of quadratic subproblems solved only approximately, thereby introducing inexactness primarily through channel (a) [10–12]. Inexactness of type (a) also arises when approximately solving subproblems across several optimization frameworks, including higher-order methods [13], interior-point strategies [14], augmented Lagrangian formulations [15], and in the broader class of inexact restoration techniques [16].

Regarding chanel (b), trust-region and inexact restoration frameworks have been rcently developed to manage accuracy in optimization problems where inexactness is intrinsic, such as finite-sum minimization, in which the objective function is typically approximated via data subsampling to reduce computational cost [17–20]. Because such approaches enable the scalable training of large-scale AI models, this has become an active and rapidly growing area of research within the optimization community.

On the other hand, the investigation of the effect of inexactness in traditional methods, which constitute the foundation of many modern optimization algorithms, dates back to a few decades earlier. The analysis by Polyak [21] examined how gradient inaccuracies affect the steepest descent method in smooth, convex, unconstrained settings, and Bertsekas and Tsitsiklis [22] later established convergence guarantees when the error is bounded by the step size times a gradient-dependent term. The accelerated framework of Nesterov [23] was extended by d'Aspremont [24] to inexact gradients without loss of optimal complexity.

Devolder et al. [25] formalized the inexact first-order oracle and analyzed primal, dual, and accelerated methods with oracle errors in smooth and convex regimes. The analysis revealed that the accelerated gradient method accumulates errors linearly with the number of iterations, whereas the standard gradient method is not affected by this issue. Building on these findings, Devolder et al. [26] proposed the intermediate gradient method, which integrates the strengths of both prior approaches. It achieves a convergence rate comparable to the accelerated gradient method while maintaining the robustness of the standard gradient method in the presence of inexact oracles. The scope of the intermediate gradient method was subsequently broadened to address problems involving stochastic oracles in [27] and Ghadimi and Lan [28]. This analysis was further expanded to include constrained optimization problems [29].

Khanh et al. [30] developed an inexact reduced gradient method for nonconvex, continuously differentiable problems, both with and without Lipschitz continuous gradients, establishing convergence under various step-size strategies. This research direction was further explored in Khanh et al. [31], where a new inexact gradient method for smooth functions was introduced.

Inexact trust-region and multilevel SQP algorithms have also been developed for DECO applications, where both the state and adjoint equations, as well as the quadratic subproblems, are solved only approximately, with adaptive mechanisms controlling inexactness and discretization accuracy [32, 33]. In the same context, Brown and Nadarajah [34, 35, 36] analyzed how inexactness in discrete adjoint computations impacts gradient methods. They argue that when the state and adjoint equations are solved with an adaptive tolerance proportional to the gradient norm, the inexact gradient method achieves the same convergence rate as its exact counterpart. In these works there is a connection between inaccuracies in solving systems of equations and in the gradient of the functions of the optimization problem.

As we can notice, inexact gradient-based algorithms have become increasingly important, driven by both theoretical challenges and practical demands. In response, we introduce a novel inexact general descent method and demonstrate its applications to DECO problems, a domain where inexactness is inherent and remains a central focus of research. Our approach provides convergence proofs for an inexact version of classical general directions strategies under both bounded and diminishing step-size rules. The analysis focuses on these more restricted step-size strategies because, in DECO settings, line-search procedures are not readily applicable due to inexactness in the objective function.

Among the works discussed so far, the most closely related are [22], [25], [36], and [30], yet our contributions differ in several key aspects. Bertsekas and Tsitsiklis [22] formulate the inexact conditions for the general direction using the unknown exact gradients, whereas our method applies these conditions directly with the available inexact gradients. The focus of Devolder et al. [25] is convex, first-order (gradient) settings with fixed search directions, rather than flexible direction schemes for general smooth problems, and it does not address DECO applications. An unconventional strategy

is adopted in Brown and Nadarajah [36], where the convergence is done assuming a strong connection between the iterates generated by an exact algorithm and its inexact counterpart. In contrast, we follow the traditional approach by directly proving the convergence of the inexact methods. The analysis in Brown and Nadarajah [36] also leaves the step-size rules unspecified and restricts some results to the gradient method. Finally, the search direction in Khanh et al. [30] is also limited to being proportional to the inexact gradient, whereas our approach admits fully general directions. Algorithmically, our method is also simpler, arising as a straightforward extension of classical general descent methods.

The remainder of the paper is organized as follows. Section 2 reviews the DECO setting, its applications, standard solution approaches, and alternative formulations. Section 3 introduces two inverse DECO case studies, specifying the objectives and the discretizations of the governing equations. Section 4 states preliminary mathematical facts and then develops the proposed inexact general descent framework with a convergence analysis. Section 5 details the adaptive test-and-tighten scheme used to compute discrete adjoint gradients with known and controlled accuracy. Section 6 reports numerical experiments on the two case studies and evaluates the effectiveness of the methods. Finally, Section 7 summarizes the main findings and outlines directions for future research.

## 2 DECO Framework and Formulations

DECO combines high-fidelity numerical simulations with optimization algorithms to solve problems governed by differential equations [37, 38]. Applications include inverse problems, design optimization, and optimal control. Inverse problems seek to identify unknown parameters in the governing equations from limited measurements of the system state, such as estimating thermal conductivity from sparse temperature data. Design optimization aims to enhance system performance through systematic modifications of geometry, configuration, or operating conditions. A classical example is aerodynamic shape optimization, where the goal may be to reduce drag or maximize the lift-to-drag ratio of an aircraft wing. Optimal control focuses on determining control inputs that steer a system toward a desired state while satisfying physical and operational constraints and optimizing a performance criterion.

To address such applications, two principal approaches to DECO have been developed: optimize–then–discretize (OD) and discretize–then–optimize (DO) [39]. In the OD approach, one derives continuous optimality conditions, such as the Karush–Kuhn–Tucker (KKT) conditions, and subsequently discretizes them for numerical implementation. By contrast, the DO approach first discretizes the governing equations and then derives the optimality conditions directly from the discrete system. These two approaches are not equivalent in general and can lead to distinct solutions [40].

Within the DO approach, the control variables $\mathbf{z} \in \mathbb{R}^m$ and state variables $\mathbf{y} \in \mathbb{R}^n$ are coupled through the discrete governing equations

$$\mathbf{R}(\mathbf{z}, \mathbf{y}) = \mathbf{0}_{\mathbb{R}^n}, \tag{1}$$

where $\mathbf{R} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ denotes the residual vector of the discretized system. The equations in (1) are also known as state equations. Depending on how (1) is enforced, two distinct formulations arise [41, 42].

In the simultaneous analysis and design (SAND) formulation, both the state and control variables are treated as unknowns, and the state equation is imposed as a

constraint. This yields the following full-space optimization problem:

$$\text{minimize} \quad F(\mathbf{z}, \mathbf{y}), \qquad \mathbf{z} \in \mathbb{R}^m, \ \mathbf{y} \in \mathbb{R}^n,$$
$$\text{subject to} \quad \mathbf{R}(\mathbf{z}, \mathbf{y}) = \mathbf{0}_{\mathbb{R}^n},$$

where $F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable objective function. Although SAND provides a unified framework, its application to high-fidelity simulations is often hindered by the computational cost and memory demands of treating both the state and control variables simultaneously [43].

Despite the state equation may not admit a unique solution in general [44], under suitable regularity conditions the implicit function theorem guarantees local uniqueness of the state for a given control [45]. In this context, the state variables can be eliminated, which leads to the nested analysis and design (NAND) formulation. In this work, we focus on residual functions that are affine in the state variable and associated with well-conditioned, nonsingular matrices, so that a global mapping can be defined. Thus, we can find a mapping

$$\mathbf{u} : \mathbb{R}^m \to \mathbb{R}^n \quad \text{such that} \quad \mathbf{R}\big(\mathbf{z}, \mathbf{u}(\mathbf{z})\big) = \mathbf{0}_{\mathbb{R}^n},$$

and, by substituting $\mathbf{y} = \mathbf{u}(\mathbf{z})$ into the objective function, we obtain a reduced-space optimization problem

$$\underset{\mathbf{z} \in \mathbb{R}^m}{\text{minimize}} \quad \widetilde{F}(\mathbf{z}) = F\left(\mathbf{z}, \mathbf{u}(\mathbf{z})\right), \tag{2}$$

where $\widetilde{F} : \mathbb{R}^m \to \mathbb{R}$ denotes the reduced objective function, assumed to be $L$-smooth and bounded from below in this work. This formulation closely resembles the classical Generalized Reduced Gradient (GRG) methods [46, 47] when the state variables are computed exactly. In practice, however, this is rarely the case, since the state dependence on the controls is obtained through numerical solvers, introducing inexactness into both the objective and its gradient evaluations. SAND and NAND are therefore fundamentally distinct and need not produce equivalent solutions [48].

With the formulation established, attention turns to solution strategies. Approaches fall broadly into two classes: gradient-free and gradient-based methods [49]. Gradient-free approaches rely solely on function evaluations and are effective for nonsmooth or noisy objectives, but their high computational cost typically restricts them to small- or medium-scale problems [50–52]. Gradient-based methods, by contrast, exploit derivative information, scale efficiently to high-dimensional control spaces, and are thus the preferred choice for large-scale DECO [53, 54]. Their performance, however, depends critically on the accurate and efficient computation of sensitivities. Several techniques are available for this purpose, including finite differences, complex-step differentiation [55], adjoint methods [56], and algorithmic differentiation (AD) [57]. Among these, adjoint methods are particularly advantageous in large-scale settings because their cost is essentially independent of the number of control variables [58]. In practice, hybrid adjoint–AD strategies are increasingly adopted to automate derivative computation and further enhance efficiency [59].

As noted above, adjoint methods enable scalable gradient-based optimization. In fact, the discrete adjoint method arises naturally when deriving the first-order necessary optimality conditions for (2). Specifically, applying the chain rule to $\widetilde{F}$ yields

$$\boldsymbol{\nabla} \widetilde{F}(\mathbf{z}) \equiv \boldsymbol{\nabla}_{\mathbf{z}} F(\mathbf{z}, \mathbf{u}(\mathbf{z})) - \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{z}} \bigg|_{(\mathbf{z}, \mathbf{u}(\mathbf{z}))} \right]^{\top} \boldsymbol{\psi} = \mathbf{0}_{\mathbb{R}^m}, \tag{3}$$

5

where $\frac{\partial \mathbf{R}}{\partial \mathbf{z}} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^{n \times m}$ and $\boldsymbol{\nabla}_{\mathbf{z}} F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ represent the partial Jacobian and gradient of $\mathbf{R}$ and $F$ with respect to $\mathbf{z}$, respectively, and the adjoint variable $\boldsymbol{\psi} \in \mathbb{R}^n$ is obtained by solving the so called adjoint equation

$$\mathbf{R}_\psi(\boldsymbol{\psi}; \mathbf{z}, \mathbf{u}(\mathbf{z})) \equiv \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{y}} \bigg|_{(\mathbf{z}, \mathbf{u}(\mathbf{z}))} \right]^\top \boldsymbol{\psi} - \boldsymbol{\nabla}_{\mathbf{y}} F(\mathbf{z}, \mathbf{u}(\mathbf{z})) = \mathbf{0}_{\mathbb{R}^n}. \tag{4}$$

In this context, $\frac{\partial \mathbf{R}}{\partial \mathbf{y}} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^{n \times n}$ and $\boldsymbol{\nabla}_{\mathbf{y}} F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ denote the partial Jacobian and gradient with respect to $\mathbf{y}$. Since the state and adjoint equations are enforced in the exact NAND formulation, the optimizer needs only to drive (3) to zero.

In summary, the discrete adjoint method provides an efficient and scalable way to compute gradients in DECO problems. However, in practical implementations, the state equation is often not solved exactly, especially during the early iterations when high accuracy is unnecessary. Consequently, the state $\mathbf{u}(\mathbf{z})$ is not computed exactly, and the quantities involved in the gradient and adjoint equations (3)-(4) become inexact. In addition, the adjoint equation may itself be solved only approximately, introducing another source of error. As a result, the gradient $\boldsymbol{\nabla} \widetilde{F}(\mathbf{z})$ is evaluated at inexact state and adjoint variables, leading to an inexact gradient. These observations indicate that inexact gradient-based methods naturally arise as an appropriate and effective framework for differential-equation-constrained optimization (DECO) problems.

## 3 Case Studies

This section presents two study cases, each formulated as an inverse problem consistent with the NAND formulation (2). In the first case, which is detailed in Subsection 3.1, the governing equation is a second-order ODE. The second case, described in Subsection 3.2, uses the Laplace equation as its governing equation. These particular governing equations were chosen because they lead to state and adjoint equations with a sparse tridiagonal structure. Such systems can be efficiently solved by iterative methods terminated when the residuals meet a specified tolerance, giving explicit control of the state and adjoint residuals and, consequently, adjoint-based gradients of known accuracy, as discussed in Section 5. The controls, which are either boundary data or coefficients, are adjusted to fit measurements at selected locations. In both study cases, the associated state equation is linear in the state variable and can be expressed in the following affine form:

$$\mathbf{R}(\mathbf{z}, \mathbf{y}) = \mathbf{A}(\mathbf{z}) \, \mathbf{y} - \mathbf{b}(\mathbf{z}), \tag{5}$$

where $\mathbf{A} : \mathbb{R}^m \to \mathbb{R}^{n \times n}$ is a matrix-valued function determined by the governing equation and the finite-difference stencil used for discretization, and $\mathbf{b} : \mathbb{R}^m \to \mathbb{R}^n$ is a vector-valued function that incorporates boundary conditions and may also include source terms.

### 3.1 Case Study 1: Inverse Problem with Parametrized Coefficients and Dirichlet Boundary Conditions

In the first case study, the governing equation consists of a boundary value problem (BVP) defined by a second-order ODE with Dirichlet boundary conditions and a

polynomial source term, as follows:

$$z_0 \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + z_1 \frac{\mathrm{d}u}{\mathrm{d}x} + z_2 u = \underbrace{z_3 + z_4 x + z_5 x^2}_{p(x)}, \qquad x \in (0, 1),$$

$$u(0) = z_6, \qquad u(1) = z_7,$$

(6)

where $\mathbf{z} = \begin{bmatrix} z_0 & z_1 & \cdots & z_7 \end{bmatrix} \in \mathbb{R}^8$ denotes the vector of control variables, with $z_0 \neq 0$. This ODE commonly appears in the analysis of second-order oscillatory systems, including mechanical (mass-spring) and electrical (RLC) systems.

Discretizing (6) with a Cartesian mesh of $M > 0$ interior points, mesh spacing $\Delta x = 1/(M+1)$, and second-order central difference operators, leads to a state equation in which $\mathbf{A} : \mathbb{R}^8 \to \mathbb{R}^{(M+2) \times (M+2)}$ and $\mathbf{b} : \mathbb{R}^8 \to \mathbb{R}^{(M+2)}$ are defined as follows:

$$\mathbf{A}(\mathbf{z}) = \begin{bmatrix} 1 & \mathbf{0}_{\mathbb{R}^M}^\top & 0 \\ \bar{a}\,\mathbf{e}_1 & \mathrm{tridiag}(\bar{a}, \bar{b}, \bar{c}) & \bar{c}\,\mathbf{e}_M \\ 0 & \mathbf{0}_{\mathbb{R}^M}^\top & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{b}(\mathbf{z}) = \begin{bmatrix} z_6 \\ p(x_1) \\ p(x_2) \\ \vdots \\ p(x_{M-1}) \\ p(x_M) \\ z_7 \end{bmatrix},$$

respectively, with

$$\bar{a} = \left( \frac{z_0}{\Delta x^2} - \frac{z_1}{2\Delta x} \right), \quad \bar{b} = \left( -\frac{2z_0}{\Delta x^2} + z_2 \right), \quad \text{and} \quad \bar{c} = \left( \frac{z_0}{\Delta x^2} + \frac{z_1}{2\Delta x} \right).$$

Here, $\mathbf{e}_1$ and $\mathbf{e}_M$ denote the first and last canonical basis vectors of $\mathbb{R}^M$. If $\mathbf{A}(\mathbf{z})$ is strictly diagonally dominant, the Levy–Desplanques theorem ensures the matrix is non-singular, thus guaranteeing a unique solution to the state equation. For this problem, given $z_1 \neq 0$ and $M > 0$, it is sufficient to choose $z_0 \geq \frac{|z_1|}{2(M+1)}$ and any $z_2 < 0$ to obtain strict diagonal dominance. The other coefficients do not affect this property. Thus, $\mathbf{u} : \mathbb{R}^8 \to \mathbb{R}^{M+2}$ is defined as $\mathbf{u}(\mathbf{z}) = [\mathbf{A}(\mathbf{z})]^{-1} \mathbf{b}(\mathbf{z})$.

Regarding the objective function and the reference values, we proceed as follows. First, a reference control vector $\mathbf{z}_{\mathrm{ref}} \in \mathbb{R}^8$ is defined and used to generate the corresponding reference state values. Specifically, the analytical solution of (6), denoted by $u(x; \mathbf{z})$, is evaluated at $\mathbf{z}_{\mathrm{ref}}$, and the reference values are set as

$$u_k^* = u(x_k; \mathbf{z}_{\mathrm{ref}}), \quad \forall k \in S,$$

where $S \subset \{0, 1, \ldots, M+1\}$ is a subset of mesh indices obtained by uniformly sampling $n_P \in (0, M+1)$ points. The objective function $\widetilde{F} : \mathbb{R}^8 \to \mathbb{R}$ is then defined as the sum of squared residuals between the computed solution and the reference values, both evaluated at the indices in $S$:

$$\widetilde{F}(\mathbf{z}) = \sum_{k \in S} [u_k(\mathbf{z}) - u_k^*]^2 + \alpha\,[u_0(\mathbf{z}) - 1]^2, \tag{7}$$

where $u_k : \mathbb{R}^8 \to \mathbb{R}$ denotes the $k$-th component of $\mathbf{u}(\mathbf{z})$, and $F : \mathbb{R}^{M+10} \to \mathbb{R}$ is defined as $F(\mathbf{z}, \mathbf{y}) = \sum_{k \in S}[y_k - u_k^*]^2$. The last term in (7), weighted by the parameter $\alpha > 0$,

penalizes deviations of $u_0(\mathbf{z})$, the leading coefficient, from unity, ensuring consistency with the normalized form of the second-order ODE.

For completeness, Figure 1 illustrates the solution of a specific instance of this inverse problem on a representative grid ($M = 64$). The optimized curve aligns precisely with the analytical solution at the sampled reference points.
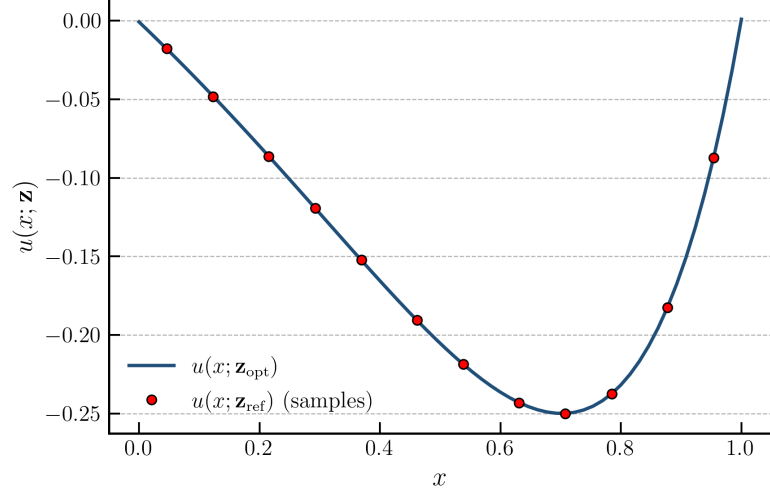


**Fig. 1**: Optimized solution curve with reference points.

## 3.2 Case Study 2: Inverse Problem with Parametrized Boundary Conditions

In the second case study, the governing equation is a BVP defined by Laplace's equation on a unit square, subject to Dirichlet boundary conditions, as follows:

$$
\begin{aligned}
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 0, \quad (x, y) \in \Omega, \\
u(x, y) - u_0(x, y; \mathbf{z}) &= 0, \quad (x, y) \in \partial\Omega,
\end{aligned}
\tag{8}
$$

where $\mathbf{z} = \begin{bmatrix} z_0 & z_1 & z_2 & \cdots & z_\ell \end{bmatrix}^T \in \mathbb{R}^{\ell+1}$, with $\ell > 1$, and $\Omega = \{(x, y) \in \mathbb{R}^2 \mid x, y \in (0, 1)\}$ is the unit square domain with boundary $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2 \cup \partial\Omega_3 \cup \partial\Omega_4$. These boundary subsets are defined by

$$
\begin{aligned}
\partial\Omega_1 &= \{(x, 0) \in \mathbb{R}^2 \mid x \in (0, 1)\}, \quad \partial\Omega_2 = \{(x, 1) \in \mathbb{R}^2 \mid x \in (0, 1)\}, \\
\partial\Omega_3 &= \{(0, y) \in \mathbb{R}^2 \mid y \in [0, 1]\}, \quad \partial\Omega_4 = \{(1, y) \in \mathbb{R}^2 \mid y \in [0, 1]\}.
\end{aligned}
$$

The function $u_0 : \partial\Omega \to \mathbb{R}$ defines the shape of the Dirichlet boundary conditions. It is expressed as

$$
u_0(x, y; \mathbf{z}) = \begin{cases} 0, & (x, y) \in \partial\Omega_1 \cup \partial\Omega_2 \cup \partial\Omega_3, \\ f(y; \mathbf{z}), & (x, y) \in \partial\Omega_4, \end{cases}
$$

where $f : [0, 1] \to \mathbb{R}$ is a polynomial function given by

$$f(y; \mathbf{z}) = z_0 + z_1 y + z_2 y^2 + \cdots + z_\ell y^\ell.$$

It is important to note that (8) arises when modeling steady-state temperature distributions in thermodynamic systems.

The BVP described in (8) can be discretized using a uniform two-dimensional Cartesian mesh with $M > 0$ interior points in each direction and mesh spacing $h = 1/(M+1)$. Since Laplace's equation is elliptic, central difference operators are suitable for approximating the derivatives. By organizing the algebraic equations resulting from the discretization in a row-wise fashion, we obtain a state equation for which $\mathbf{A} : \mathbb{R}^{\ell+1} \to \mathbb{R}^{(M+2)^2 \times (M+2)^2}$ and $\mathbf{b} : \mathbb{R}^{\ell+1} \to \mathbb{R}^{(M+2)^2}$ are defined as follows:

$$\mathbf{A}(\mathbf{z}) = \left[\begin{array}{c|ccccc|c} \mathbf{I} & & & & & & \\ \hline \mathbf{B} & \mathbf{T} & \mathbf{B} & & & & \\ & \mathbf{B} & \mathbf{T} & \mathbf{B} & & & \\ & & \mathbf{B} & \mathbf{T} & \mathbf{B} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{B} & \mathbf{T} & \mathbf{B} \\ & & & & & \mathbf{B} & \mathbf{T} & \mathbf{B} \\ \hline & & & & & & \mathbf{I} \end{array}\right], \quad \text{and} \quad \mathbf{b}(\mathbf{z}) = \left[\begin{array}{c} \mathbf{0}_{(M+1)(M+2) \times 1} \\ \hline f(0; \mathbf{z}) \\ f(h; \mathbf{z}) \\ \vdots \\ f((M+1)h; \mathbf{z}) \end{array}\right],$$

respectively, where

$$\mathbf{T} = \begin{bmatrix} 1 & \mathbf{0}_{\mathbb{R}^M}^\top & 0 \\ \mathbf{e}_1 & \text{tridiag}(1, -4, 1) & \mathbf{e}_M \\ 0 & \mathbf{0}_{\mathbb{R}^M}^\top & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 & \mathbf{0}_{\mathbb{R}^M}^\top & 0 \\ \mathbf{0}_{\mathbb{R}^M} & \mathbf{I}_{M \times M} & \mathbf{0}_{\mathbb{R}^M} \\ 0 & \mathbf{0}_{\mathbb{R}^M}^\top & 0 \end{bmatrix}$$

are in $\mathbb{R}^{(M+2) \times (M+2)}$. Here, $\mathbf{I}_{M \times M}$ is the identity matrix of order $M$. It should be noted that if the boundary conditions are addressed separately, $\mathbf{A}$ can be simplified to a symmetric, negative definite block tridiagonal matrix. This property guarantees that $\mathbf{A}$ is non-singular. As a result, we can define $\mathbf{u} : \mathbb{R}^{\ell+1} \to \mathbb{R}^{(M+2)^2}$ by $\mathbf{u}(\mathbf{z}) = [\mathbf{A}(\mathbf{z})]^{-1} \mathbf{b}(\mathbf{z})$ as in the first case study.

Turning to the construction of the objective function and the reference values for this problem, we proceed as follows. Initially, we define $\Omega_d \subset \Omega$ as the set of desired point coordinates used to evaluate each reference value. To construct this set, we employ the Latin hypercube sampling technique, which enables the generation of a uniformly distributed set of points across the domain [60]. Next, these points are mapped to the corresponding mesh points to build $\Omega_d$, and we outline the set of its mesh indices as

$$\mathcal{I}(\Omega_d) = \{(j, k) \in \mathcal{I} \times \mathcal{I} \mid (jh, kh) \in \Omega_d\}, \quad \text{where} \quad \mathcal{I} = \{0, 1, \ldots, M+1\}.$$

For the evaluation of the reference values, we consider the analytical solution of (8) at a reference vector of control variables $\mathbf{z}_{\text{ref}} \in \mathbb{R}^{\ell+1}$, denoted by $u(x, y; \mathbf{z}_{\text{ref}})$, obtained by solving the problem with $f(y; \mathbf{z}_{\text{ref}})$. This analytical solution is then used to compute the reference state values $u_d^{j,k}$ at the points in $\Omega_d$, so that

$$u_d^{j,k} = u(jh, kh; \mathbf{z}_{\text{ref}}), \quad \forall (j, k) \in \mathcal{I}(\Omega_d).$$

Then, we finally define the objective function $\widetilde{F} : \mathbb{R}^{\ell+1} \to \mathbb{R}$ as

$$\widetilde{F}(\mathbf{z}) = \sum_{(j,k)\in\mathcal{I}(\Omega_d)} \left[ u_{j,k}(\mathbf{z}) - u_d^{j,k} \right]^2,$$

where $u_{j,k} : \mathbb{R}^{\ell+1} \to \mathbb{R}$ denotes the $(j,k)$-th component of $\mathbf{u}(\mathbf{z})$, and $F :$
$\mathbb{R}^{\ell+1+(M+2)^2} \to \mathbb{R}$ is defined as $F(\mathbf{z}, \mathbf{y}) = \sum\limits_{(j,k)\in\mathcal{I}(\Omega_d)} \left[ y_{j,k} - u_d^{j,k} \right]^2.$

For completeness, Figure 2 illustrates the solution of a specific instance of this inverse problem in a representative grid ($M = 64$). The optimized surface aligns precisely with the analytical solution at the sampled reference points.
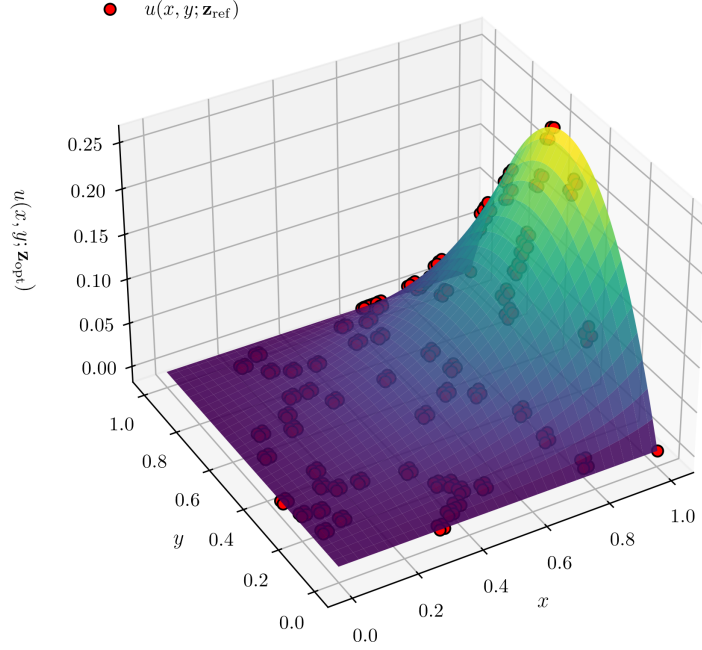


**Fig. 2**: Optimized solution surface with reference points.

# 4 Convergence Theory

This section begins with preliminaries that establish the necessary background for the analysis, which will be restricted to the case of smooth functions. We then introduce the proposed algorithm and outline its key properties. Following this, we determine conditions under which we carry out a convergence analysis.

## 4.1 Mathematical Preliminaries

It is well known that smooth functions can be upper-bounded by a quadratic polynomial [61]. In a series of works, Devolder et al. [25], Devolder [62, 63] demonstrated a similar result for inexact gradients, as shown in the following lemma. The proof provided here follows Claim 4.1 in [64].

**Lemma 4.1** *Let $\widetilde{F} : \mathbb{R}^m \to \mathbb{R}$ be an L-smooth function, and $\mathbf{z} \in \mathbb{R}^m$ a specified vector of control variables. Suppose that the inexact gradient of $\widetilde{F}$, denoted by $\overline{\boldsymbol{\nabla}}\widetilde{F} : \mathbb{R}^m \to \mathbb{R}^m$, satisfies*

$$\left\| \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}) - \boldsymbol{\nabla}\widetilde{F}(\mathbf{z}) \right\|_2 \leq \delta_{\mathbf{z}}, \quad \forall \mathbf{z} \in \mathbb{R}^m, \tag{9}$$

*where $\boldsymbol{\nabla}\widetilde{F} : \mathbb{R}^m \to \mathbb{R}^m$ denotes the exact gradient of $\widetilde{F}$, and $\delta_{\mathbf{z}} \in \mathbb{R}_{\geq 0}$ indicates the error tolerance at $\mathbf{z}$. Then,*

$$\widetilde{F}(\mathbf{x}) \leq \widetilde{F}(\mathbf{z}) + \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z})^T(\mathbf{x} - \mathbf{z}) + \frac{L'}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \delta_{\mathbf{z}}', \quad \forall \mathbf{x}, \mathbf{z} \in \mathbb{R}^m,$$

*where $L' = L + 1$ and $\delta_{\mathbf{z}}' = \delta_{\mathbf{z}}^2/2$.*

*Proof* Since $\widetilde{F}$ is $L$-smooth, it follows that

$$\widetilde{F}(\mathbf{x}) \leq \widetilde{F}(\mathbf{z}) + \boldsymbol{\nabla}\widetilde{F}(\mathbf{z})^T(\mathbf{x} - \mathbf{z}) + \frac{L}{2}\|\mathbf{x} - \mathbf{z}\|_2^2, \quad \forall \mathbf{x} \in \mathbb{R}^m. \tag{10}$$

The gradient term in (10) can be rewritten as

$$\boldsymbol{\nabla}\widetilde{F}(\mathbf{z})^T(\mathbf{x} - \mathbf{z}) = \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z})^T(\mathbf{x} - \mathbf{z}) + \left[\boldsymbol{\nabla}\widetilde{F}(\mathbf{z}) - \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z})\right]^T(\mathbf{x} - \mathbf{z}). \tag{11}$$

Now, define $\mathbf{w} = \boldsymbol{\nabla}\widetilde{F}(\mathbf{z}) - \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z})$ and $\mathbf{v} = \mathbf{x} - \mathbf{z}$. Since $\|\mathbf{w} - \mathbf{v}\|_2^2 \geq 0$ by the non-negativity property of norms, we can deduce that

$$\left[\boldsymbol{\nabla}\widetilde{F}(\mathbf{z}) - \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z})\right]^T(\mathbf{x} - \mathbf{z}) \leq \frac{1}{2}\left[\left\|\boldsymbol{\nabla}\widetilde{F}(\mathbf{z}) - \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z})\right\|_2^2 + \|\mathbf{x} - \mathbf{z}\|_2^2\right]. \tag{12}$$

Substituting (12) in (11) yields the following upper bound for the gradient term:

$$\boldsymbol{\nabla}\widetilde{F}(\mathbf{z})^T(\mathbf{x} - \mathbf{z}) \leq \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z})^T(\mathbf{x} - \mathbf{z}) + \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \frac{1}{2}\left\|\boldsymbol{\nabla}\widetilde{F}(\mathbf{z}) - \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z})\right\|_2^2. \tag{13}$$

Applying (13) to (10), and subsequently utilizing (9) within the result, yields the desired inequality. $\square$

## 4.2 Inexact General Descent Method

This section introduces an *Inexact General Descent Method* (IGDM) to address smooth unconstrained problems. The method is characterized by the following update rule:

$$\mathbf{z}_{k+1} = \mathbf{z}_k + t_k\mathbf{s}_k, \quad \forall k \in \mathbb{N}_0, \tag{14}$$

where $(t_k)_{k \in \mathbb{N}_0}$ represents a positive step size sequence, and $\mathbf{s}_k \in \mathbb{R}^m$ indicates an inexact general direction. The directional conditions we use here originate from classical descent methods such as in [22], but we replace the exact gradients with their inexact counterparts. We assume that there exist constants $c_1' > 0$ and $c_2' > 0$ such that $\mathbf{s}_k$ satisfies the following conditions:

$$c_1'\left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2^2 \leq -\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)^T\mathbf{s}_k, \quad \forall k \in \mathbb{N}_0, \tag{15a}$$

and

$$\|\mathbf{s}_k\|_2 \leq c_2'\left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2, \quad \forall k \in \mathbb{N}_0. \tag{15b}$$

Having introduced the update rule, the search direction, and the error condition for the inexact gradient, we can consolidate the procedure of the IGDM in Algorithm 1.

For this, we consider that the inexact gradient error $\mathbf{e}_k = \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) - \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_k)$ is such that

$$\|\mathbf{e}_k\|_2 \leq \delta_k, \quad \forall k \in \mathbb{N}_0, \tag{16}$$

where $(\delta_k)_{k \in \mathbb{N}_0}$ is a non-negative error tolerance sequence.

---

**Algorithm 1:** INEXACT GENERAL DESCENT METHOD

---

**Data:** An initial point $\mathbf{z}_0 \in \mathbb{R}^m$
**Result:** A sequence $(\mathbf{z}_k)_{k \in \mathbb{N}_0} \subset \mathbb{R}^m$ whose limit points, if they exist, are stationary

**Step 1.** Set $k \leftarrow 0$.
**Step 2.** Choose $\delta_k \geq 0$.
**Step 3.** Compute $\overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \in \mathbb{R}^m$ such that

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) - \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_k) \right\|_2 \leq \delta_k.$$

**Step 4.** Compute $\mathbf{s}_k \in \mathbb{R}^m$ so that (15) holds.
**Step 5.** Choose $t_k > 0$ by a specific rule.
**Step 6.** Set $\mathbf{z}_{k+1} \leftarrow \mathbf{z}_k + t_k \mathbf{s}_k$.
**Step 7.** Set $k \leftarrow k + 1$ and **go to Step 2**.

---

As will be shown in the next subsection, the method converges provided the sequence $(\delta_k)_{k \in \mathbb{N}_0}$ satisfies additional conditions that depend on the step–size rule.

## 4.3 Convergence Analysis

We now turn to the convergence analysis of the IGDM. We begin by presenting several auxiliary results that form the foundation of the proofs. By combining the IGDM update rule with its associated directional conditions, we introduce the following assumption, which will be used throughout the subsequent results:

*Assumption A.1* Let $\widetilde{F} : \mathbb{R}^m \to \mathbb{R}$ be an $L$-smooth function. Given an initial point $\mathbf{z}_0 \in \mathbb{R}^m$, consider the sequence $(\mathbf{z}_k)_{k \in \mathbb{N}}$ generated by (14) employing a positive sequence of step sizes $(t_k)_{k \in \mathbb{N}}$, and a general directions $(\mathbf{s}_k)_{k \in \mathbb{N}}$ that fulfills (15a) and (15b) for a pair of constants $c_1' > 0$ and $c_2' > 0$.

Following the classical gradient descent method, our analysis is built upon a key relation between the change in the objective function value across successive iterations and the (inexact) gradient at the current point. This relation, which will be instrumental in establishing the convergence theorems for the step-size strategies considered later, is formalized in the following lemma.

**Lemma 4.2** *Suppose that Assumption A.1 is satisfied. If* (16) *holds, then*

$$\widetilde{F}(\mathbf{z}_{k+1}) \leq \widetilde{F}(\mathbf{z}_k) - t_k \left( c_1' - (c_2')^2 \frac{L' t_k}{2} \right) \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2^2 + \delta_k', \quad \forall k \in \mathbb{N}_0, \tag{17}$$

*where $L' = L + 1$ and $\delta_k' = \delta_k^2 / 2$.*

*Proof* Since $\widetilde{F}$ is an $L$-smooth function and (16) holds, we can apply Lemma 4.1 with $\mathbf{x} = \mathbf{z}_{k+1}$, $\mathbf{z} = \mathbf{z}_k$, and $\delta_{\mathbf{z}_k}' = \delta_k'$. Proceeding in this way, we obtain

$$\widetilde{F}(\mathbf{z}_{k+1}) \leq \widetilde{F}(\mathbf{z}_k) + \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k)^T (\mathbf{z}_{k+1} - \mathbf{z}_k) + \frac{L'}{2} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^2 + \delta_k'$$

$$\leq \widetilde{F}(\mathbf{z}_k) + t_k \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k)^T \mathbf{s}_k + \frac{L'}{2} t_k^2 \|\mathbf{s}_k\|_2^2 + \delta_k'. \tag{18}$$

Using (15a) and (15b) in (18) yields (17). $\qquad \square$

*Remark 4.1* Note that Lemma 4.2 does not inherently guarantee the monotonic descent property of $\left(\widetilde{F}(\mathbf{z}_k)\right)_{k\in\mathbb{N}_0}$, even when $t_k < \frac{c_1'}{(c_2')^2}\frac{2}{L}$ is satisfied for all $k \in \mathbb{N}_0$, owing to the presence of an additional error term in (17). This limitation was previously documented by [25].

When the sequence $(\delta_k)_{k\in\mathbb{N}_0}$ is defined as a quantity proportional to the gradient or the inexact gradient norm, a common practice in the literature [64], it proves useful in ensuring the descent property. The subsequent result formalizes this observation.

**Lemma 4.3** *Suppose that Assumption A.1 and* (16) *hold. If there exists* $\beta \in \left[0, \frac{1}{\sqrt{L'}}\frac{c_1'}{c_2'}\right)$ *such that the sequence* $(\delta_k)_{k\in\mathbb{N}_0}$ *satisfies*

$$\delta_k \le \beta\left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2, \quad \forall k \in \mathbb{N}_0, \tag{19}$$

*then*

*(a)*

$$\widetilde{F}(\mathbf{z}_{k+1}) \le \widetilde{F}(\mathbf{z}_k) - (c_2')^2\frac{L'}{2}\left(t_k - \bar{t}_1\right)\left(\bar{t}_2 - t_k\right)\left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2^2, \quad \forall k \in \mathbb{N}_0,$$

*where*

$$\bar{t}_1 = \frac{c_1' - \sqrt{\Delta}}{(c_2')^2 L'}, \quad and \quad \bar{t}_2 = \frac{c_1' + \sqrt{\Delta}}{(c_2')^2 L'}, \quad with \quad \Delta = (c_1')^2 - L'(c_2')^2\beta^2,$$

*(b) and*

$$0 \le \bar{t}_1 < \bar{t}_2 \le \frac{c_1'}{(c_2')^2}\frac{2}{L'}. \tag{20}$$

*Proof* Combining (17), the definition of $\delta_k'$ in Lemma 4.2 and (19) yields

$$\widetilde{F}(\mathbf{z}_{k+1}) \le \widetilde{F}(\mathbf{z}_k) - \left(-(c_2')^2\frac{L'}{2}t_k^2 + c_1' t_k - \frac{\beta^2}{2}\right)\left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2^2$$

$$\le \widetilde{F}(\mathbf{z}_k) - (c_2')^2\frac{L'}{2}(t_k - \bar{t}_1)(\bar{t}_2 - t_k)\left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2^2,$$

which completes the proof of item (a). Now, observe that $\beta \in \left[0, \frac{1}{\sqrt{L'}}\frac{c_1'}{c_2'}\right)$ implies that $0 < \Delta \le (c_1')^2$. So the roots $\bar{t}_1$ and $\bar{t}_2$ satisfies (20). $\qquad\square$

*Remark 4.2* Lemma 4.3 guarantees monotonic descent if $t_k \in (\bar{t}_1, \bar{t}_2)$ for all $k$. The size of this interval depends on the error: larger $\beta$ (greater error) shrinks the interval, while smaller $\beta$ (less error) widens it. This limits how small the step size can be, unlike in the exact case.

IGDM convergence can be analyzed using either Lemma 4.2 or Lemma 4.3. With Lemma 4.2, convergence is shown for bounded or diminishing step sizes if $(\delta_k)_{k\in\mathbb{N}_0}$ is quadratically summable or summable, respectively. Without extra conditions, monotonic descent isn't guaranteed. In contrast, Lemma 4.3 guarantees monotonic descent, but only for bounded step sizes, since diminishing step sizes strategy certainly would not result in steps within an interval with a positive lower bound.

### 4.3.1 Convergence under a Bounded Step Size

This subsection proves that, provided the step sizes remain bounded by some specific values, any limit point of the sequence is a stationary point.

**Theorem 4.3** *Suppose that Assumption A.1 holds and that $\widetilde{F}$ is bounded below. Assume further that (16) is satisfied by a non-negative sequence $(\delta_k)_{k \in \mathbb{N}_0}$ which fulfills (19) with $\beta \in \left[0, \frac{1}{\sqrt{L'}} \frac{c_1'}{c_2'}\right)$. Let $\bar{t}_1$ and $\bar{t}_2$ be as in Lemma 4.3, and let $\gamma \in \left(0, \frac{\bar{t}_2 - \bar{t}_1}{2}\right]$. If $t_k \in [\bar{t}_1 + \gamma, \bar{t}_2 - \gamma]$ for all $k \in \mathbb{N}_0$, then*

(a) $\widetilde{F}(\mathbf{z}_{k+1}) \leq \widetilde{F}(\mathbf{z}_k), \quad \forall k \in \mathbb{N}_0.$

(b) $\lim_{k \to \infty} \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) = \mathbf{0}_{\mathbb{R}^m}, \quad and \quad \lim_{k \to \infty} \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_k) = \mathbf{0}_{\mathbb{R}^m}.$

*Furthermore, every limit point of $(\mathbf{z}_k)_{k \in \mathbb{N}_0}$ is a stationary point of $\widetilde{F}$.*

*Proof* Combining item (a) of Lemma 4.3 with the fact that $t_k \in [\bar{t}_1 + \gamma, \bar{t}_2 - \gamma]$, we have that

$$\widetilde{F}(\mathbf{z}_{k+1}) \leq \widetilde{F}(\mathbf{z}_k) - \frac{L'(c_2')^2 \gamma^2}{2} \left\|\overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k)\right\|_2^2 \leq \widetilde{F}(\mathbf{z}_k), \quad \forall k \in \mathbb{N}_0. \tag{21}$$

Summing from 0 to $K - 1$, with $K \geq 1$, results in

$$\widetilde{F}(\mathbf{z}_K) \leq \widetilde{F}(\mathbf{z}_0) - \frac{L'(c_2')^2 \gamma^2}{2} \sum_{k=0}^{K-1} \left\|\overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k)\right\|_2^2. \tag{22}$$

Given that $\widetilde{F}$ is bounded below, it follows from (22) that

$$\sum_{k=0}^{\infty} \left\|\overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k)\right\|_2^2 < \infty. \tag{23}$$

Consequently,

$$\lim_{k \to \infty} \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) = \mathbf{0}_{\mathbb{R}^m}. \tag{24}$$

Combining (16), (19) and (24) yields

$$\lim_{k \to \infty} \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_k) = \mathbf{0}_{\mathbb{R}^m}. \tag{25}$$

This concludes the proof of item (b). By continuity, every limit point of $(\mathbf{z}_k)_{k \in \mathbb{N}_0}$ is a stationary point of $\widetilde{F}$. □

### 4.3.2 Convergence under Diminishing Step Size

We now analyze convergence under diminishing step sizes. In contrast to the bounded (constant) step-size regime, which often requires tuning using problem-specific quantities such as a Lipschitz constant, we focus on a positive sequence $(t_k)_{k \in \mathbb{N}_0}$ that satisfies

$$\sum_{k=0}^{\infty} t_k^2 < \infty, \tag{26}$$

and

$$\sum_{k=0}^{\infty} t_k = \infty. \tag{27}$$

These classical conditions guarantee sustained progress while steps vanish, thereby mitigating the need for precise knowledge of problem constants.

For this step size strategy, our proof relies on the general argument presented in [65], when analyzing the exact version of the method. In this context, the following technical auxiliary lemma is crucial.

**Lemma 4.4** *Suppose that Assumption A.1 holds, and that* (16) *is satisfied by a non-negative, summable sequence* $(\delta_k)_{k \in \mathbb{N}_0}$. *If*

$$\sum_{k=0}^{\infty} t_k \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2^2 < \infty, \tag{28}$$

*and*

$$\liminf_{k \to \infty} \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2 = 0, \tag{29}$$

*then*

$$\lim_{k \to \infty} \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2 = 0. \tag{30}$$

*Proof* In order to derive a contradiction, suppose that (30) fails to hold. Consequently, there exists $\epsilon > 0$ such that

$$\limsup_{k \to \infty} \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2 \geq \epsilon. \tag{31}$$

Since (29) and (31) hold, the sequence of inexact gradient norms exhibit an oscillatory pattern, which allows the partition of the sequence into segments characterized by relatively small or large norms. Accordingly, define $(m_j)_{j \in \mathbb{N}_0}$ and $(n_j)_{j \in \mathbb{N}_0}$ as sequences of indices such that

$$m_j < n_j < m_{j+1}, \quad \forall j \in \mathbb{N}_0,$$

and

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2 > \frac{\epsilon}{3}, \quad \text{for } m_j \leq k < n_j, \tag{32}$$

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2 \leq \frac{\epsilon}{3}, \quad \text{for } n_j \leq k < m_{j+1}. \tag{33}$$

Since $(\delta_k)_{k \in \mathbb{N}_0}$ is summable, given $\frac{\epsilon}{12} > 0$ there exist a sufficiently large constant $j_1 \in \mathbb{N}_0$ such that

$$\sum_{k=m_{j_1}}^{\infty} \delta_k < \frac{\epsilon}{12}. \tag{34}$$

Analogously, based on (28), it is possible to determine a sufficiently large constant $\mathbb{N}_0 \ni j_2 > j_1$ such that

$$\sum_{k=m_{j_2}}^{\infty} t_k \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2^2 < \frac{\epsilon^2}{18L}. \tag{35}$$

Now, note that

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_{k+1}) - \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2 \leq \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_{k+1}) - \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_{k+1}) \right\|_2 + \left\| \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_{k+1}) - \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_k) \right\|_2$$
$$+ \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) - \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_k) \right\|_2$$
$$\leq \delta_{k+1} + L \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2 + \delta_k. \tag{36}$$

Consequently, for every $j \in \mathbb{N}_0$, where $j \geq j_2$, and each $m \in \mathbb{N}_0$ satisfying $m_j \leq m \leq n_j - 1$, it follows from (34) and (36) that

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_{n_j}) - \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_m) \right\|_2 \leq \sum_{k=m}^{n_j-1} \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_{k+1}) - \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2$$

$$\leq \sum_{k=m}^{n_j-1} \delta_{k+1} + L \sum_{k=m}^{n_j-1} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2 + \sum_{k=m}^{n_j-1} \delta_k$$

$$= \frac{\epsilon}{6} + L \sum_{k=m}^{n_j-1} t_k \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2. \tag{37}$$

By exploiting (32) and (35), the right-hand side of (37) simplifies to

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_{n_j}) - \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_m) \right\|_2 \leq \frac{\epsilon}{6} + \frac{3L}{\epsilon} \sum_{k=m}^{n_j-1} t_k \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2^2$$

$$\leq \frac{\epsilon}{6} + \frac{3L}{\epsilon} \frac{\epsilon^2}{18L}$$

$$= \frac{\epsilon}{3}. \tag{38}$$

Based on (33) and (38), it can be established that

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_m) \right\|_2 \leq \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_m) - \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_{n_j}) \right\|_2 + \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_{n_j}) \right\|_2$$

$$\leq \frac{\epsilon}{3} + \frac{\epsilon}{3}$$

$$= \frac{2\epsilon}{3}, \quad \forall j \geq j_2, \quad m_j \leq m \leq n_j - 1. \tag{39}$$

By combining (33) and (39), it is immediate that

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_m) \right\|_2 \leq \frac{2\epsilon}{3}, \quad \forall j \geq j_2, \quad m_j \leq m < m_{j+1}$$

which means that

$$\left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_m) \right\|_2 \leq \frac{2\epsilon}{3}, \quad \forall m \geq m_{j_2}.$$

However, this contradicts (31). Therefore, (30) must holds. $\qquad \square$

Utilizing Lemma 4.4, we can complete the convergence proof for the diminishing step-size scenario, as detailed in the theorem below.

**Theorem 4.4** *Suppose that Assumption A.1 holds and that $\widetilde{F}$ is bounded below. Assume further that (16) is satisfied by a non-negative, summable sequence $(\delta_k)_{k \in \mathbb{N}_0}$. If the sequence $(t_k)_{k \in \mathbb{N}_0}$ is diminishing in accordance with (26) and (27), then*

$$\lim_{k \to \infty} \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) = \mathbf{0}_{\mathbb{R}^m}, \quad and \quad \lim_{k \to \infty} \boldsymbol{\nabla} \widetilde{F}(\mathbf{z}_k) = \mathbf{0}_{\mathbb{R}^m}.$$

*Furthermore, every limit point of $(\mathbf{z}_k)_{k \in \mathbb{N}_0}$ is a stationary point of $\widetilde{F}$.*

*Proof* By Lemma 4.2, we have that

$$\widetilde{F}(\mathbf{z}_{k+1}) \leq \widetilde{F}(\mathbf{z}_k) - t_k \left( c_1' - (c_2')^2 \frac{L' t_k}{2} \right) \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2^2 + \delta_k', \quad \forall k \in \mathbb{N}_0. \tag{40}$$

Since $(t_k)_{k \in \mathbb{N}_0}$ is quadratically summable, it converges to zero. Thus, for a given $\widetilde{c} \in (0, c_1')$ there exists a constant $k_1 \in \mathbb{N}$ such that

$$t_k \leq \frac{2(c_1' - \widetilde{c})}{(c_2')^2 L'}, \quad \forall k \geq k_1.$$

Rearranging terms yields

$$-\widetilde{c} \geq - \left( c_1' - (c_2')^2 \frac{L' t_k}{2} \right), \quad \forall k \geq k_1. \tag{41}$$

Using (41) in (40), we arrive at

$$\widetilde{F}(\mathbf{z}_{k+1}) \leq \widetilde{F}(\mathbf{z}_k) - \widetilde{c}t_k \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2^2 + \delta_k', \quad \forall k \geq k_1. \tag{42}$$

By summing (42) over $k = k_1, \ldots, K - 1$, with $K - 1 > k_1$, we obtain

$$\widetilde{F}(\mathbf{z}_K) \leq \widetilde{F}(\mathbf{z}_{k_1}) + \frac{1}{2} \sum_{k=k_1}^{K-1} \delta_k^2 - \widetilde{c} \sum_{k=k_1}^{K-1} t_k \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2^2. \tag{43}$$

As $(\delta_k)_{k \in \mathbb{N}_0}$ is nonnegative and summable, it follows that it is also quadratically summable. Thus, using the fact that $\widetilde{F}$ is bounded below, we obtain that

$$\sum_{k=k_1}^{\infty} t_k \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2^2 < \infty. \tag{44}$$

Now, suppose for contradiction that there exist a constant $\epsilon > 0$, and $k_2 \in \mathbb{N}$ such that

$$\left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2 \geq \epsilon, \quad \forall k \geq k_2.$$

Thus, since $(t_k)_{k \in \mathbb{N}_0}$ is not summable, we have

$$\sum_{k=k_2}^{\infty} t_k \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2^2 \geq \epsilon^2 \sum_{k=k_2}^{\infty} t_k = \infty.$$

However, this contradicts condition (44). Hence,

$$\liminf_{k \to \infty} \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2 = 0. \tag{45}$$

By virtue of conditions (44) and (45), Lemma 4.4 ensures that

$$\lim_{k \to \infty} \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) = \mathbf{0}_{\mathbb{R}^m}. \tag{46}$$

From the triangle inequality and (16), we obtain

$$\left\|\boldsymbol{\nabla}\widetilde{F}(\mathbf{z}_k)\right\|_2 \leq \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2 + \left\|\boldsymbol{\nabla}\widetilde{F}(\mathbf{z}_k) - \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2$$
$$\leq \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2 + \delta_k. \tag{47}$$

As both terms on the right-hand side of (47) approach zero as $k \to \infty$, the proof is completed.
$\square$

## 4.4 Computation of the General Direction

Having established the convergence analysis of Algorithm 1, attention now turns to its practical implementation. Fortunately, a procedure can be formulated for selecting a general direction $\mathbf{s}_k$ that inherently fulfills the conditions in (15). The following theorem formalizes this result. The statement parallels the sufficient-descent results for Newton-type directions in Forsgren et al. [66], adapted here to allow inexact gradients on the right-hand side and to ensure positive definiteness via a uniformly bounded spectrum.

**Theorem 4.5** *Let $\widetilde{F} : \mathbb{R}^m \to \mathbb{R}$ be a differentiable function. Given an initial point $\mathbf{z}_0 \in \mathbb{R}^m$, consider the sequence $(\mathbf{z}_k)_{k \in \mathbb{N}}$ generated by (14) using a positive sequence of step sizes $(t_k)_{k \in \mathbb{N}_0}$, and a general direction $\mathbf{s}_k \in \mathbb{R}^m$ that satisfies*

$$\mathbf{H}_k \mathbf{s}_k = -\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k), \quad \forall k \in \mathbb{N}_0,$$

*where $\mathbf{H}_k \in \mathbb{R}^{m \times m}$ is a real symmetric with a uniformly bounded spectrum matrix, which means that there exist constants $\mu, \eta > 0$ such that*

$$0 < \mu \leq \lambda_i(\mathbf{H}_k) \leq \eta, \quad \forall i \in \{1, \ldots, m\}, \forall k \in \mathbb{N}_0,$$

*where $\lambda_i(\mathbf{H}_k)$ denotes the ith eigenvalue of $\mathbf{H}_k$. Then the conditions (15a) and (15b) are valid for some pair of constants $c_1' > 0$ and $c_2' > 0$.*

*Proof* See Lemma 2.3 in [66]. $\square$

17

# 5 Adjoint-Based Gradient Error

The convergence results in Subsection 4.3 are based on the assumption of a bounded inexact-gradient error (16). This condition holds for several sensitivity schemes, including finite differences, linear interpolation, and Gaussian and spherical smoothing [67]. Importantly, this requirement is also met by the discrete adjoint method, which we use in this work to evaluate the sensitivities of the objective functions in our study cases. Below, we outline why the adjoint-based gradient meets the bounded-error condition, drawing on the framework of [36], which analyzes how inexact state and adjoint solves propagate to gradient inaccuracies in the discrete adjoint method.

Suppose that a state equation of the form given by (5) is approximately solved in each iteration as follows:

$$\left\| \mathbf{R}\left( \mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)} \right) \right\|_2 \leq \tau_{\mathcal{R},k}, \quad \forall k \in \mathbb{N}_0, \tag{48}$$

where $\overline{\mathbf{u}(\mathbf{z}_k)} \in \mathbb{R}^n$ is an approximation for $\mathbf{u}(\mathbf{z}_k)$, and $\tau_{\mathcal{R},k} > 0$ is an adaptive state tolerance. Similarly, given these approximate states, assume that the adjoint equation in (4) is solved inexactly, namely,

$$\left\| \mathbf{R}_\psi\left( \overline{\boldsymbol{\psi}}_k; \mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)} \right) \right\|_2 \leq \tau_{\psi,k}, \quad \forall k \in \mathbb{N}_0, \tag{49}$$

where $\overline{\boldsymbol{\psi}}_k \in \mathbb{R}^n$ denotes an approximation of the exact solution of the adjoint equation with inexact state variables, and $\tau_{\psi,k} > 0$ is an adaptive adjoint tolerance. Using both the inexact state and adjoint variables obtained from (48) and (49), we obtain the following inexact gradient from (3):

$$\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) = \boldsymbol{\nabla}_{\mathbf{z}} F\left( \mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)} \right) - \left[ \left. \frac{\partial \mathbf{R}}{\partial \mathbf{z}} \right|_{\left( \mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)} \right)} \right]^\top \overline{\boldsymbol{\psi}}_k, \quad \forall k \in \mathbb{N}_0. \tag{50}$$

For linear state equations as described by (5), it can be shown from the theoretical framework established by [36] that the inexact gradient presented in (50) satisfies

$$\left\| \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) - \boldsymbol{\nabla}\widetilde{F}(\mathbf{z}_k) \right\|_2 \leq C' \tau_{\mathcal{R},k} + C'' \tau_{\psi,k}, \quad \forall k \in \mathbb{N}_0, \tag{51}$$

where $C' > 0$ and $C'' > 0$ are constants that depend on the Lipschitz continuity and boundedness of the partial Jacobians of $F$ and $\mathbf{R}$ with respect to both the control and state variables. Thus, if we define the state and adjoint tolerances as

$$\tau_{\mathcal{R},k} \leq \gamma_1 \left\| \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) \right\|_2, \quad \text{and} \quad \tau_{\psi,k} \leq \gamma_2 \left\| \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) \right\|_2, \quad \forall k \in \mathbb{N}_0, \tag{52}$$

where $\gamma_1 \in \left( 0, \dfrac{\beta}{2C'} \right]$ and $\gamma_2 \in \left( 0, \dfrac{\beta}{2C''} \right]$ are defined constants for a given $\beta > 0$, it follows from (51) and (52) that

$$\begin{aligned}
\left\| \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) - \boldsymbol{\nabla}\widetilde{F}(\mathbf{z}_k) \right\|_2 &\leq \left( C' \frac{\beta}{2C'} + C'' \frac{\beta}{2C''} \right) \left\| \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) \right\|_2 \\
&\leq \beta \left\| \overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) \right\|_2, \quad \forall k \in \mathbb{N}_0.
\end{aligned} \tag{53}$$

The established condition corresponds with that outlined in (19) if $\beta$ is small enough. Consequently, we can ensure the convergence of the IGDM by employing a suitable constant step size, with gradients computed inexactly through the discrete adjoint method.

To enforce (53), the inexact gradient at iteration $k \geq 0$ is computed via a discrete-adjoint test-and-tighten scheme (Algorithm 2) with a state–adjoint nesting. Because the target bounds in (51) depend on the (unknown) inexact gradient, we start with trial tolerances and tighten them geometrically as needed. Given $\mathbf{z}_k$, the outer loop solves the state equation (48) up to a trial target $\tau_{\mathcal{R},k}^{(i)}$. An inner loop solves the adjoint equation (49) up to $\tau_{\psi,k}^{(j)}$ and compute the inexact gradient, and it is repeated tightening $\tau_{\psi,k}^{(j)}$ until the adjoint residual meets the bound proportional to $\gamma_2$ times the inexact gradient. Then we check if the state residual meets the required bound, proportional to $\gamma_1$ times the inexact gradient. If the criterion is satisfied, the approximate value of the state variables is set, otherwise $\tau_{\mathcal{R},k}^{(i)}$ is tightened and the outer loop repeats. The procedure terminates whenever the exact gradient is not null, ensuring that both bounds in (48) and (49) are satisfied simultaneously for

$$\tau_{\mathcal{R},k} = \gamma_1 \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2, \quad \text{and} \quad \tau_{\psi,k} = \gamma_2 \left\| \overline{\boldsymbol{\nabla}} \widetilde{F}(\mathbf{z}_k) \right\|_2, \tag{54}$$

which guarantees (53). The trial tolerances $\tau_{\mathcal{R},k}^{(i)}$ and $\tau_{\psi,k}^{(j)}$ are recognized only as iterates used to discover these levels.

As we can see, the test-and-tighten scheme allows us to ensure that the inexact gradient error satisfies (53). This differs from the approach in [36], where the authors employ a purely heuristic strategy, fixing the tolerances in the solution of the state and adjoint equations to be proportional to the inexact gradient norm at the previous iterate.

---

**Algorithm 2:** ADAPTIVE INEXACT STATE–ADJOINT RESIDUAL ALGORITHM

---

**Data:** Current iterate $\mathbf{z}_k$; previous gradient norm $\left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_{k-1})\right\|_2$; initial tolerances $\tau_{\mathcal{R}}^{(0)}$, $\tau_{\psi}^{(0)}$; lower bounds $\underline{\tau_{\mathcal{R}}}$, $\underline{\tau_{\psi}}$; and parameters $\gamma_1 \in \left(0, \dfrac{\beta}{2C'}\right]$, $\gamma_2 \in \left(0, \dfrac{\beta}{2C''}\right]$.

**1** Set $i \leftarrow 0$ and define the state tolerance as

**2** **if** $k = 0$ **then**

**3** $\quad\left\lfloor\ \tau_{\mathcal{R},k}^{(i)} \leftarrow \tau_{\mathcal{R}}^{(0)}\right.$

**4** **else**

**5** $\quad\left\lfloor\ \tau_{\mathcal{R},k}^{(i)} \leftarrow \min\left(0.5 \cdot \gamma_1 \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_{k-1})\right\|_2, \tau_{\mathcal{R}}^{(0)}\right)\right.$

**6** **repeat**

**7** $\quad$ Compute $\overline{\mathbf{u}(\mathbf{z}_k)} \in \mathbb{R}^n$ by solving $\left\|\mathbf{R}\left(\mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)}\right)\right\|_2 \leq \tau_{\mathcal{R},k}^{(i)}$

**8** $\quad$ Set $j \leftarrow 0$ and the adjoint tolerance

**9** $\quad$ **if** $k = 0$ **then**

**10** $\quad\quad\left\lfloor\ \tau_{\psi,k}^{(j)} \leftarrow \tau_{\psi}^{(0)}\right.$

**11** $\quad$ **else**

**12** $\quad\quad\left\lfloor\ \tau_{\psi,k}^{(j)} \leftarrow \min\left(0.5 \cdot \gamma_2 \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_{k-1})\right\|_2, \tau_{\psi}^{(0)}\right)\right.$

**13** $\quad$ **repeat**

**14** $\quad\quad$ Compute $\overline{\boldsymbol{\psi}}_k \in \mathbb{R}^n$ by solving $\left\|\mathbf{R}_{\psi}\left(\overline{\boldsymbol{\psi}}_k; \mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)}\right)\right\|_2 \leq \tau_{\psi,k}^{(j)}$

**15** $\quad\quad$ $\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k) \leftarrow \boldsymbol{\nabla}_{\mathbf{z}}F\left(\mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)}\right) - \left[\left.\dfrac{\partial \mathbf{R}}{\partial \mathbf{z}}\right|_{\left(\mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)}\right)}\right]^{\top} \overline{\boldsymbol{\psi}}_k$

**16** $\quad\quad$ $j \leftarrow j + 1$

**17** $\quad\quad$ Update: $\tau_{\psi,k}^{(j)} \leftarrow 0.1 \cdot \gamma_2 \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2$

**18** $\quad$ **until** $\left\|\mathbf{R}_{\psi}\left(\overline{\boldsymbol{\psi}}_k; \mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)}\right)\right\|_2 \leq \gamma_2 \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2$

**19** $\quad$ $i \leftarrow i + 1$

**20** $\quad$ Update: $\tau_{\mathcal{R},k}^{(i)} \leftarrow 0.5 \cdot \gamma_1 \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2$

**21** **until** $\left\|\mathbf{R}\left(\mathbf{z}_k, \overline{\mathbf{u}(\mathbf{z}_k)}\right)\right\|_2 \leq \gamma_1 \left\|\overline{\boldsymbol{\nabla}}\widetilde{F}(\mathbf{z}_k)\right\|_2$

---

# 6 Numerical Results

This section presents numerical experiments involving the DECO problems described in Section 3. The objective is to assess the computational savings obtained by using inexact adjoint gradients with adaptive tolerances, as opposed to fixed, restrictive tolerances. In addition, we aim to determine whether incorporating a BFGS-like update for additional flexibility in the search direction provides further computational benefits within the inexact framework. To achieve these objectives, we implemented two instances of the Inexact General Descent Method (IGDM) in C++, computing inexact gradients with the adaptive procedure in Algorithm 2, which keeps the gradient error tolerance proportional to the inexact gradient. Since convergence has been demonstrated under this error tolerance condition for bounded step sizes, we adopted a constant, data-aware step size that scales inversely with the number of sampled reference points. The specific step-size rule used in each problem setup is discussed in the following subsections.

In the first instance, we consider an Inexact Gradient Descent (IGD) method defined by the search direction

$$\mathbf{s}_k^{\mathrm{IGD}} = -\overline{\boldsymbol{\nabla}}\,\widetilde{F}(\mathbf{z}_k), \quad k \in \mathbb{N}_0.$$

In this configuration, the inexact adjoint gradients are computed using Algorithm 2 with adaptive tolerance parameters $\gamma_1 = 60$ and $\gamma_2 = 3$. These parameters were chosen through preliminary grid searches and demonstrated stable performance across problem sizes, and they are consistently used in both case studies presented in this section. For reference, we also implemented a conventional Gradient Descent (GD) method, in which adjoint gradients are computed using fixed tight tolerances $\tau_{\mathcal{R},k} = \tau_{\psi,k} = 10^{-9}$ for all $k \in \mathbb{N}_0$, and termination occurs when the norm of the exact gradient falls below the same threshold.

To investigate whether curvature information could provide additional savings beyond the inexact gradient scheme, we implemented a second instance of IGDM that updates an inverse-Hessian approximation through a BFGS-type recursion. Starting with $\mathbf{H}_0^{-1} = \mathbf{I}_{m \times m}$, this approximation is updated at each iteration by

$$\mathbf{H}_{k+1}^{-1} = \left(\mathbf{I}_{m \times m} - \rho_k \mathbf{p}_k \mathbf{y}_k^\top\right)\mathbf{H}_k^{-1}\left(\mathbf{I}_{m \times m} - \rho_k \mathbf{y}_k \mathbf{p}_k^\top\right) + \rho_k \mathbf{p}_k \mathbf{p}_k^\top,$$

where

$$\rho_k = \frac{1}{\mathbf{y}_k^\top \mathbf{p}_k}, \qquad \mathbf{p}_k = \mathbf{z}_{k+1} - \mathbf{z}_k, \qquad \mathbf{y}_k = \overline{\boldsymbol{\nabla}}\,\widetilde{F}(\mathbf{z}_{k+1}) - \overline{\boldsymbol{\nabla}}\,\widetilde{F}(\mathbf{z}_k).$$

The corresponding search direction is then given by

$$\mathbf{s}_k^{\mathrm{IBFGS}} = -\mathbf{H}_k^{-1}\overline{\boldsymbol{\nabla}}\,\widetilde{F}(\mathbf{z}_k).$$

IGDM with this choice of direction yields an inexact BFGS-like method (IBFGS). Due to the inexact nature of the gradient, we apply a few stabilization measures to ensure robustness. First, the adaptive tolerance parameters $\gamma_1$ and $\gamma_2$ are tuned separately for each case study to maintain reliable curvature updates and mitigate instabilities. In addition, a Powell-type damping is employed to preserve the symmetry and positive definiteness of the inverse-Hessian approximations [68].

As noted before, both IGD and IBFGS were implemented under the same adaptive tolerance framework given by Algorithm 2. A practical aspect of this implementation concerns the choice of its input parameters. The trial targets start from fixed initial tolerances $\tau_{\mathcal{R}}^{(0)} > 0$ and $\tau_{\psi}^{(0)} > 0$ and are progressively tightened during the iterations. To prevent oversolving as these targets decrease, we enforce lower bounds $\underline{\tau}_{\mathcal{R}}$ and $\underline{\tau}_{\psi}$, both set to $10^{-9}$ in this work. Regarding the coefficients $\gamma_1$ and $\gamma_2$, we selected their values empirically based on short preliminary runs, since their theoretical upper bounds depend on Lipschitz-type constants that are typically unknown or unreliable in practice.

The numerical experiments for both instances of IGDM are presented separately in Subsection 6.1 and Subsection 6.2, corresponding to the two case studies considered. All computations were performed on a system equipped with an Intel Core i7-8550U CPU running at $1.80\,\mathrm{GHz}$, 8 GB of RAM, and a $512\,\mathrm{GB}$ SSD, using Ubuntu 24.04.2 LTS.

## 6.1 Case Study 1: Second-Order Inverse Problem

For the second-order inverse problem described in Subsection 3.1, both the state and adjoint equations were solved using in-house C++ implementations based on a successive over-relaxation (SOR) scheme. Since the corresponding linear systems are Toeplitz tridiagonal, we used a relaxation factor close to its optimal value in the SOR iterations. For the inverse reconstruction, we uniformly sampled $n_P = 12$ mesh indices to form the set $S$. At these locations, we evaluated the analytical solution $u(x; \mathbf{z})$ with the reference control vector $\mathbf{z}_{\text{ref}} = \begin{bmatrix} 1 & -3 & -4 & 1 & 1 & 5 & 0 & 0 \end{bmatrix}^\top$ to obtain the reference state values. To ensure consistency across mesh refinements, we anchored the sampling locations so that approximately the same physical coordinates were used to compute the reference values for all tested values of $M$. Preliminary tests indicated that the optimizers operated effectively across these mesh sizes using a step size $t = 1.15/n_P = 9.583 \times 10^{-2}$. The initial guess was $\mathbf{z}_0 = \begin{bmatrix} 1.5 & 2.0 & -7.0 & 0.2 & -0.4 & 0.1 & 1.0 & -0.2 \end{bmatrix}^\top$, and the stopping criterion was defined as the inexact gradient norm falling below the tolerance $\epsilon = 1 \times 10^{-6}$. These settings were applied consistently across all methods. It is worth noting that all optimizations successfully converged.

To evaluate the numerical performance under these settings, we analyzed wall-clock optimization times and the associated computational savings across different mesh resolutions. For each grid size $M$ from 16 to 128, we measured total optimization time for GD and IGD and presented the corresponding runtime curves, comparing the results across small, medium, and large instances. In addition, we reported the percentage-reduction curve of IGD relative to GD for $M = 64$ through 128, where the gap between the methods became more pronounced, and included a horizontal line indicating the average percentage reduction over this interval. We then used the IGD curve obtained from this first comparison as a baseline to benchmark the IBFGS method, applying the same range of problem sizes and reporting optimization time as a function of $M$. Finally, we also included the percentage-reduction curve of IBFGS relative to IGD, together with the average reduction over the range, which quantified the computational savings attributable to curvature information in the inexact framework.

The results of these comparisons are summarized in Figure 3, which presents wall-clock optimization time (seconds) as a function of $M$ for GD, IGD, and IBFGS, together with the corresponding percentage-reduction curves. On average, IGD reduced runtime by $45.55\,\%$ relative to GD, while IBFGS achieved an average reduction of $87.69\,\%$ relative to IGD. Thus, IBFGS consistently outperformed IGD, as expected. Aside from a few outliers, the pointwise reductions clustered near these averages across $M$, indicating robust behavior. Further exploratory repeats with the same settings, not shown here, confirmed the same qualitative behavior. For IBFGS, we used $\gamma_1 = 1 \times 10^{-3}$ and $\gamma_2 = 1 \times 10^{-3}$. With these settings, curvature safeguards, the descent-check fallback, and bad-curvature events were rare across $M$, and no resets to the identity were observed in the reported runs. We also verified that the counts of state and adjoint solves closely tracked the iteration count across $M$, indicating minimal overhead from the adaptive tolerance mechanism prescribed by Algorithm 2.
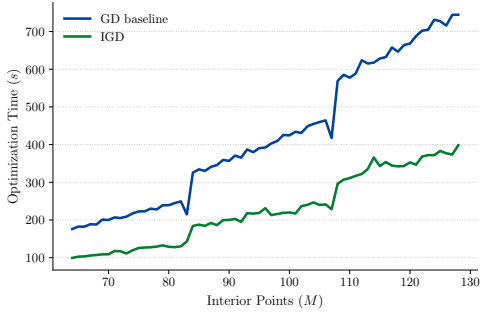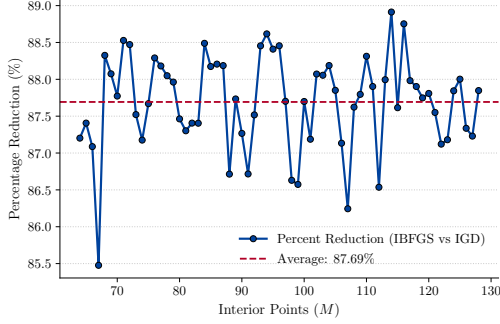
(a) IGD compared with GD baseline.
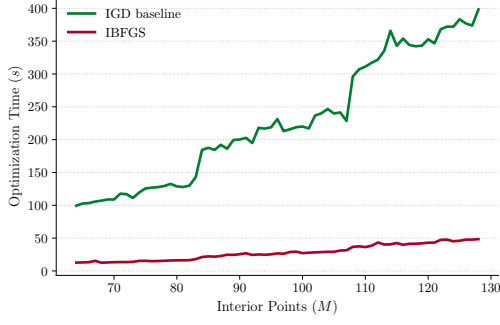


(b) IBFGS compared with IGD baseline.
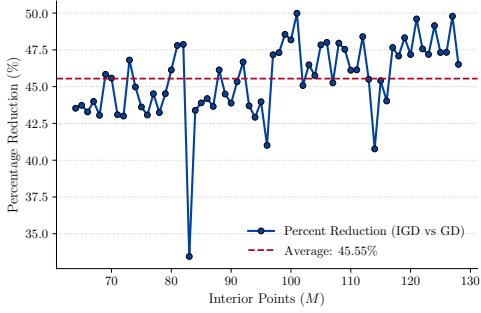


(c) Percent reduction: IGD relative to GD.



(d) Percent reduction: IBFGS relative to IGD.

**Fig. 3**: Optimization time versus interior points $M$ with percentage-reduction curves. IGD uses $\gamma_1 = 60$, $\gamma_2 = 3$, and IBFGS uses $\gamma_1 = 1 \times 10^{-3}$, $\gamma_2 = 1 \times 10^{-3}$. All methods use constant step size $t = 9.583 \times 10^{-2}$ and stopping tolerance $\epsilon = 1 \times 10^{-6}$. Horizontal lines show the mean reduction for $64 \leq M \leq 128$.

## 6.2 Case Study 2: Laplace Inverse Problem

For the Laplace inverse problem described in Subsection 3.2, the state equation was solved using an alternating direction implicit (ADI) scheme, where the tridiagonal subsystems in each coordinate direction were handled via Thomas' algorithm. The adjoint equation was solved using restarted GMRES with sparse matrix–vector products and SOR preconditioning. In contrast to Case Study 1, where the number of sampled measurements was fixed, here we scaled the number of samples with the grid size to provide more data to fit and to increase the effective complexity of the reconstruction as the mesh was refined. This design is meant to test whether the algorithm can preserve its efficiency under these more demanding conditions. Specifically, for each grid size $M$ we drew $n_s = 4(M+1)+2$ mesh indices by Latin hypercube sampling to form $\mathcal{I}(\Omega_d)$, and at these locations we evaluated the analytical solution at $\mathbf{z}_{\mathrm{ref}} = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}^\top$ to obtain the reference state values. Since $n_s$ increases with $M$, we used the grid-dependent step size $t = 3.5/(n_s - 2)$, which empirically maintained stability and yielded consistent efficiency across mesh sizes. The initial guess was $\mathbf{z}_0 = \begin{bmatrix} 0.5 & 1.0 & 0.25 \end{bmatrix}^\top$, and the stopping criterion was set to the inexact gradient norm falling below $\epsilon = 10^{-6}$. These configurations were uniformly applied across all methods.

The same comparison procedure described in Case Study 1 was adopted here to evaluate performance across methods and mesh resolutions. The results are summarized in Figure 4, which presents wall-clock optimization time as a function of $M$ for GD, IGD, and IBFGS, together with the corresponding percentage-reduction curves. The qualitative pattern resembled that of Case Study 1, with one key difference: the saving for IGD over GD was much larger here, averaging 90.21 %, whereas the additional saving of IBFGS over IGD averaged 79.61 %. Although the latter was smaller than in Case Study 1, it remained substantial. These percentage reductions translated into larger absolute time savings than in Case Study 1, particularly for the GD-to-IGD comparison. For IBFGS, we set $\gamma_1 = 1 \times 10^{-1}$ and $\gamma_2 = 1 \times 10^{-1}$, following the same rationale as in the previous case. Pointwise reductions remained close to the averages across $M$, confirming robust behavior as before. Safeguard triggers and descent-check fallbacks were infrequent, no identity resets were observed, and the counts of state and adjoint solves closely tracked the iteration counts across $M$, again indicating minimal overhead from the adaptive tolerance mechanism prescribed by Algorithm 2.
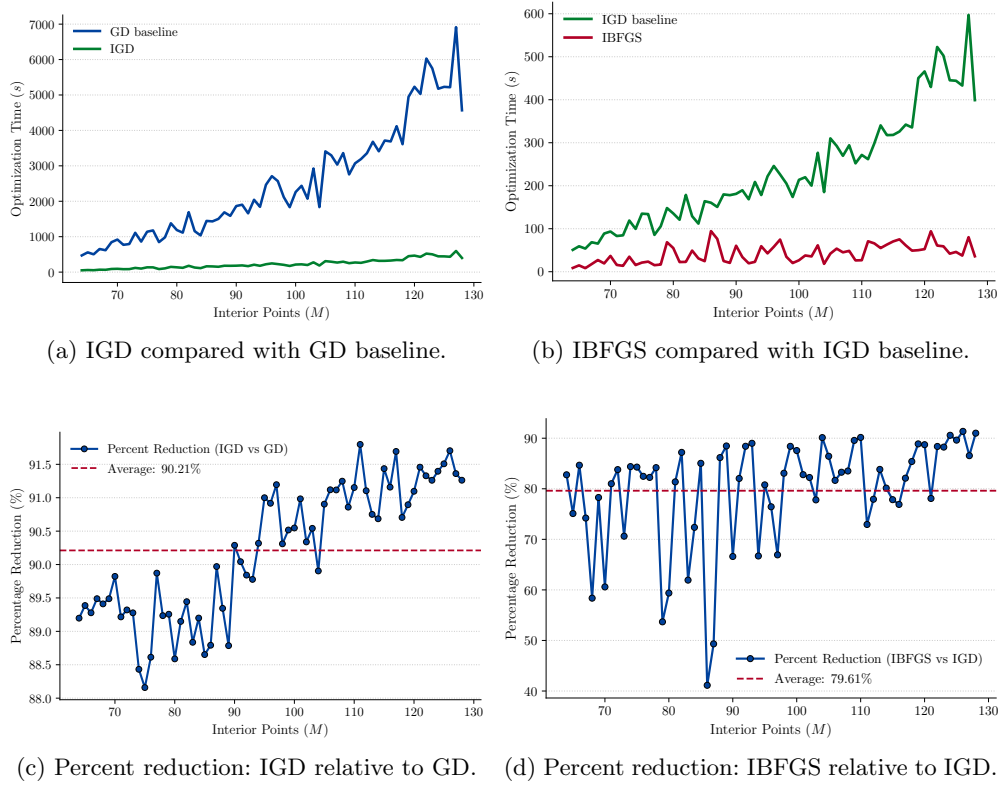


(a) IGD compared with GD baseline.

(b) IBFGS compared with IGD baseline.

(c) Percent reduction: IGD relative to GD.

(d) Percent reduction: IBFGS relative to IGD.

**Fig. 4**: Wall-clock optimization time as a function of interior points $M$, together with percentage-reduction curves. IGD uses $\gamma_1 = 60$ and $\gamma_2 = 3$. IBFGS uses $\gamma_1 = 1 \times 10^{-1}$ and $\gamma_2 = 1 \times 10^{-1}$. All methods use stopping tolerance $\epsilon = 1 \times 10^{-6}$. Horizontal lines indicate the mean reduction for $64 \leq M \leq 128$.

# 7 Final Remarks and Future Research

Motivated by DECO applications, this work introduced a novel inexact-gradient-based framework with general inexact search directions for smooth unconstrained optimization. A comprehensive convergence analysis was developed for this framework under both bounded and diminishing step-size rules. For bounded step sizes, we identified an admissible interval that ensures convergence while guaranteeing a monotonic-descent property when the error-tolerance sequence is proportional to the norm of the computed inexact gradient. For diminishing step sizes, convergence was achieved under summable error-tolerance sequences, although monotonic descent could not be ensured in this case. Furthermore, the proposed framework combines theoretical rigor and practical implementability by allowing flexible search directions and by defining acceptance conditions that depend directly on the inexact gradient rather than on its unavailable exact counterpart.

Two instances of the framework were implemented and tested with constant step sizes: the Inexact Gradient Descent (IGD) and the Inexact BFGS-like (IBFGS) methods. These were applied to a second-order inverse problem and to a two-dimensional Laplace inverse problem. In both applications, gradients were computed via an adaptive adjoint test-and-tighten scheme, which adjusts the state and adjoint tolerances dynamically to produce gradients with controlled accuracy. The use of adaptive inexact gradients led to significant runtime reductions compared to fixed-tolerance baselines, while incorporating curvature information through the IBFGS scheme provided additional computational savings. Both IGD and IBFGS exhibited consistent performance across mesh sizes, with negligible overhead from the adaptive tolerance mechanism. For IBFGS, the implemented safeguards effectively maintained a well-conditioned inverse-Hessian approximation.

Despite these encouraging results, several limitations remain. The study cases considered residual functions that are linear in the state variables, ensuring globally well-defined state mappings. A natural continuation is to address nonlinear residuals, where such mappings may fail to exist globally. The present framework also focuses solely on unconstrained problems, and its extension to include equality and inequality constraints remains an open and practically relevant challenge. Moreover, the numerical experiments were restricted to the bounded step-size regime because only in this setting we could prove convergence for error tolerances proportional to the inexact gradient norm. Finally, the current implementation does not employ a line-search procedure, as inexact objective evaluations make its direct use nontrivial.

Future research will primarily focus on overcoming these limitations. We plan to broaden the theoretical analysis to nonlinear residual and constrained problems and to design line-search or trust-region mechanisms that incorporate precision-control explicitly. In addition, an important theoretical direction involves developing conditions that allow the use of diminishing step sizes even when error tolerances remain proportional to the inexact gradient. Beyond these extensions, further research avenues include the use of Inexact Restoration techniques to globalize the optimization process, allowing the algorithm to follow infeasible paths with respect to the state variable.

## Statements and Declarations

**Competing interests.** The authors have no relevant financial or non-financial interests to disclose.

**Supplementary Information.** Animations illustrating the two study cases are available at: https://youtu.be/X905iImwfmE and https://youtu.be/onk9BWbxR3s.

# References

[1] Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact Newton methods. SIAM J. Numer. Anal. **19**(2), 400–408 (1982) https://doi.org/10.1137/0719025

[2] Eisenstat, S.C., Walker, H.F.: Globally convergent inexact Newton methods. SIAM J. Optim. **4**(2), 393–422 (1994) https://doi.org/10.1137/0804022

[3] Eisenstat, S.C., Walker, H.F.: Choosing the forcing terms in an inexact Newton method. SIAM J. Sci. Comput. **17**(1), 16–32 (1996) https://doi.org/10.1137/09 17003

[4] Bergamaschi, L., Moret, I., Zilli, G.: Inexact quasi-Newton methods for sparse systems of nonlinear equations. Future Gener. Comput. Syst. **18**(1), 41–53 (2001) https://doi.org/10.1016/S0167-739X(00)00074-1

[5] Birgin, E.G., Krejić, N., Martínez, J.M.: Globally convergent inexact quasi-Newton methods for solving nonlinear systems. Numer. Algorithms **32**(2), 249–260 (2003) https://doi.org/10.1023/A:1024013824524

[6] Arias, C.A., Martínez, H.J., Pérez, R.: Global inexact quasi-Newton method for nonlinear system of equations with constraints. Appl. Numer. Math. **150**, 559–575 (2020) https://doi.org/10.1016/j.apnum.2019.11.002

[7] Arias, C.A., Gómez, C.: Inexact free derivative quasi-Newton method for large-scale nonlinear system of equations. Numer. Algorithms **94**(3), 1103–1123 (2023) https://doi.org/10.1007/s11075-023-01529-6

[8] Piétrus, A., Santos, P.S.M., Silva, G.N.: Inexact quasi-Newton methods under a relaxed metric regularity assumption. Numer. Algorithms (2024) https://doi.or g/10.1007/s11075-024-01979-6

[9] Martínez, J.M.: Algorithms for solving nonlinear systems of equations. In: Spedicato, E. (ed.) Algorithms for Continuous Optimization: The State of the Art, pp. 81–108. Springer, Dordrecht (1994). https://doi.org/10.1007/978-94-009-0369-2 _4

[10] Leibfritz, F., Sachs, E.W.: Inexact SQP interior point methods and large-scale optimal control problems. SIAM J. Control Optim. **38**(1), 272–293 (1999) https://doi.org/10.1137/S0363012996298795

[11] Byrd, R.H., Curtis, F.E., Nocedal, J.: An inexact SQP method for equality-constrained optimization. SIAM J. Optim. **19**(1), 351–369 (2008) https://doi.or g/10.1137/060674004

[12] Curtis, F.E., Johnson, T.C., Robinson, D.P., Wächter, A.: An inexact sequential quadratic optimization algorithm for nonlinear optimization. SIAM J. Optim. **24**(3), 1041–1074 (2014) https://doi.org/10.1137/130918320

[13] Birgin, E.G., Gardenghi, J.L., Martínez, J.M., Santos, S.A., Toint, P.L.: Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. Math. Program. **163**(1), 359–368 (2017) https://doi.org/10.1007/s10107-016-1065-8

[14] Curtis, F.E., Schenk, O., Wächter, A.: An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. SIAM Journal on Scientific Computing **32**(6), 3447–3475 (2010) https://doi.org/10.1137/090747634 https://doi.org/10.1137/090747634

[15] Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On augmented lagrangian methods with general lower-level constraints. SIAM J. Optim. **18**(4), 1286–1309 (2008) https://doi.org/10.1137/060654797

[16] Bueno, L.F., Martínez, J.M.: On the complexity of an inexact restoration method for constrained optimization. SIAM Journal on Optimization **30**(1), 80–101 (2020) https://doi.org/10.1137/18M1216146 https://doi.org/10.1137/18M1216146

[17] Curtis, F.E., Scheinberg, K., Shi, R.: A stochastic trust region algorithm based on careful step normalization. INFORMS J. Optim. **1**(3), 200–220 (2019) https://doi.org/10.1287/ijoo.2018.0010

[18] Bellavia, S., Krejić, N., Morini, B.: Inexact restoration with subsampled trust-region methods for finite-sum minimization. Comput. Optim. Appl. **76**(3), 701–736 (2020) https://doi.org/10.1007/s10589-020-00196-w

[19] Bellavia, S., Krejić, N., Morini, B., Rebegoldi, S.: A stochastic first-order trust-region method with inexact restoration for finite-sum minimization. Comput. Optim. Appl. **84**(1), 53–84 (2023) https://doi.org/10.1007/s10589-022-00430-7

[20] Cao, L., Berahas, A.S., Scheinberg, K.: First- and second-order high-probability complexity bounds for trust-region methods with noisy oracles. Math. Program. **207**(1), 55–106 (2024) https://doi.org/10.1007/s10107-023-01999-5

[21] Polyak, B.T.: Introduction to Optimization. Translations Series in Mathematics and Engineering. Optimization Software, Publications Division, New York (1987)

[22] Bertsekas, D.P., Tsitsiklis, J.N.: Gradient convergence in gradient methods with errors. SIAM J. Optim. **10**(3), 627–642 (2000) https://doi.org/10.1137/S1052623497331063

[23] Nesterov, Y.: A method for solving the convex programming problem with convergence rate $o(1/k^2)$. In: Doklady Akademii Nauk SSSR, vol. 269, pp. 543–547 (1983). In Russian; English transl. in Soviet Math. Dokl. 27:372–376, 1983. https://cir.nii.ac.jp/crid/1370862715914709505

[24] d'Aspremont, A.: Smooth optimization with approximate gradient. SIAM J.

Optim. **19**(3), 1171–1183 (2008) https://doi.org/10.1137/060676386

[25] Devolder, O., Glineur, F., Nesterov, Y.: First-order methods of smooth convex optimization with inexact oracle. Math. Program. **146**(1), 37–75 (2014) https://doi.org/10.1007/s10107-013-0677-5

[26] Devolder, O., Glineur, F., Nesterov, Y.: Intermediate gradient methods for smooth convex problems with inexact oracle. Technical Report CORE-2013017, CORE, Université catholique de Louvain, Louvain-la-Neuve, Belgium (2013). Available at https://core.ac.uk/download/pdf/34118280.pdf (accessed 17 Mar 2025)

[27] Dvurechensky, P., Gasnikov, A.: Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. J. Optim. Theory Appl. **171**(1), 121–145 (2016) https://doi.org/10.1007/s10957-016-0999-6

[28] Ghadimi, S., Lan, G.: Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Math. Program. **156**(1), 59–99 (2016) https://doi.org/10.1007/s10107-015-0871-8

[29] Ghadimi, S., Lan, G., Zhang, H.: Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. Math. Program. **155**(1), 267–305 (2016) https://doi.org/10.1007/s10107-014-0846-1

[30] Khanh, P.D., Mordukhovich, B.S., Tran, D.B.: Inexact reduced gradient methods in nonconvex optimization. J. Optim. Theory Appl. **203**(3), 2138–2178 (2024) https://doi.org/10.1007/s10957-023-02319-9

[31] Khanh, P.D., Mordukhovich, B.S., Tran, D.B.: A new inexact gradient descent method with applications to nonsmooth convex optimization. Optim. Methods Softw. **39**, 1–29 (2024) https://doi.org/10.1080/10556788.2024.2322700

[32] Heinkenschloss, M., Ridzal, D.: An inexact trust-region SQP method with applications to PDE-constrained optimization. In: Kunisch, K., Of, G., Steinbach, O. (eds.) Numerical Mathematics and Advanced Applications, pp. 613–620. Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69777-0_73

[33] Ziems, J.C., Ulbrich, S.: Adaptive multilevel inexact SQP methods for PDE-constrained optimization. SIAM J. Optim. **21**(1), 1–40 (2011) https://doi.org/10.1137/080743160

[34] Brown, D.A., Nadarajah, S.: An adaptive constraint tolerance method for optimization algorithms based on the discrete adjoint method. In: Proceedings of the 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee, FL, USA (2018). https://doi.org/10.2514/6.2018-0414 . American Institute of Aeronautics and Astronautics (AIAA). AIAA Paper 2018–0414

[35] Brown, D.A., Nadarajah, S.: Inexactly constrained discrete adjoint approach for steepest-descent-based optimization algorithms. Numer. Algorithms **78**(3), 983–1000 (2018) https://doi.org/10.1007/s11075-017-0409-7

[36] Brown, D.A., Nadarajah, S.: Effect of inexact adjoint solutions on the discrete-adjoint approach to gradient-based optimization. Optim. Eng. **23**(3), 1643–1676 (2022) https://doi.org/10.1007/s11081-021-09681-5

[37] Hicken, J.E., Ashley, A.: Arnoldi-based sampling for high-dimensional optimization using imperfect data. In: Proceedings of the 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Dallas, TX, USA (2015). https://doi.org/10.2514/6.2015-2943 . American Institute of Aeronautics and Astronautics (AIAA). AIAA Paper 2015–2943

[38] Antil, H., Kouri, D.P., Lacasse, M.-D., Ridzal, D. (eds.): Frontiers in PDE-constrained Optimization. IMA Vol. Math. Appl., vol. 163. Springer, New York, NY (2018)

[39] Liu, J., Wang, Z.: Non-commutative discretize-then-optimize algorithms for elliptic PDE-constrained optimal control problems. J. Comput. Appl. Math. **362**, 596–613 (2019) https://doi.org/10.1016/j.cam.2018.07.028

[40] Collis, S.S., Heinkenschloss, M.: Analysis of the SUPG method for the solution of optimal control problems. arXiv Preprint (2024) https://doi.org/10.48550/arXiv.2411.09828 2411.09828

[41] Haftka, R.T.: Simultaneous analysis and design. AIAA J. **23**(7), 1099–1103 (1985) https://doi.org/10.2514/3.9043

[42] Cramer, E.J., J. E. Dennis, J., Frank, P.D., Lewis, R.M., Shubin, G.R.: Problem formulation for multidisciplinary optimization. SIAM J. Optim. **4**(4), 754–776 (1994) https://doi.org/10.1137/0804044

[43] Hicken, J.E., Alonso, J.J.: Comparison of reduced- and full-space algorithms for PDE-constrained optimization. In: Proceedings of the 51st AIAA Aerospace Sciences Meeting, Grapevine, TX, USA (2013). https://doi.org/10.2514/6.2013-1043 . American Institute of Aeronautics and Astronautics (AIAA). AIAA Paper 2013–1043

[44] Jameson, A.: Airfoils admitting non-unique solutions of the Euler equations. In: Proceedings of the 22nd Fluid Dynamics, Plasma Dynamics and Lasers Conference, Honolulu, HI, USA (1991). https://doi.org/10.2514/6.1991-1625 . American Institute of Aeronautics and Astronautics (AIAA). AIAA Paper 1991–1625

[45] Borggaard, J., Burns, J., Cliff, E., Schreck, S. (eds.): Computational Methods for Optimal Design and Control. Birkhäuser, Boston, MA (1998)

[46] Abadie, J.: Generalization of the wolfe reduced gradient method to the case of nonlinear constraints. Optimization, 37–47 (1969)

[47] Lasdon, L.S., Waren, A.D., Jain, A., Ratner, M.: Design and testing of a generalized reduced gradient code for nonlinear programming. ACM Trans. Math. Softw. **4**(1), 34–50 (1978) https://doi.org/10.1145/355769.355773

[48] Heinkenschloss, M., Vicente, L.N.: An interface optimization and application for the numerical solution of optimal control problems. ACM Trans. Math. Softw.

**25**(2), 157–190 (1999) https://doi.org/10.1145/317275.317278

[49] Zingg, D.W., Nemec, M., Pulliam, T.H.: A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. Eur. J. Comput. Mech. **17**(1-2), 103–126 (2008) https://doi.org/10.3166/remn.17.103-126

[50] Holst, T., Pulliam, T.: Aerodynamic shape optimization using a real-number-encoded genetic algorithm. In: Proceedings of the 19th AIAA Applied Aerodynamics Conference, Anaheim, CA, USA (2001). https://doi.org/10.2514/6.2001-2473 . American Institute of Aeronautics and Astronautics (AIAA). AIAA Paper 2001–2473

[51] Chernukhin, O., Zingg, D.W.: Multimodality and global optimization in aerodynamic design. AIAA J. **51**(6), 1342–1354 (2013) https://doi.org/10.2514/1.J051835

[52] He, Y., Agarwal, R.K.: Shape optimization of NREL S809 airfoil for wind turbine blades using a multi-objective genetic algorithm. In: Proceedings of the 32nd AIAA Applied Aerodynamics Conference, Atlanta, GA, USA (2014). https://doi.org/10.2514/6.2014-2845 . American Institute of Aeronautics and Astronautics (AIAA). AIAA Paper 2014–2845

[53] Secco, N.R., Jasa, J.P., Kenway, G.K.W., Martins, J.R.R.A.: Component-based geometry manipulation for aerodynamic shape optimization with overset meshes. AIAA J. **56**(9), 3667–3679 (2018) https://doi.org/10.2514/1.J056550

[54] Secco, N.R., Martins, J.R.R.A.: RANS-based aerodynamic shape optimization of a strut-braced wing with overset meshes. J. Aircraft **56**(1), 217–227 (2019) https://doi.org/10.2514/1.C034934

[55] Squire, W., Trapp, G.: Using complex variables to estimate derivatives of real functions. SIAM Rev. **40**(1), 110–112 (1998) https://doi.org/10.1137/S003614459631241X

[56] Bryson, A.E., Ho, Y.-C.: Applied Optimal Control: Optimization, Estimation and Control. Routledge, New York (1975)

[57] Wengert, R.E.: A simple automatic derivative evaluation program. Commun. ACM **7**(8), 463–464 (1964) https://doi.org/10.1145/355586.364791

[58] Kenway, G.K.W., Mader, C.A., He, P., Martins, J.R.R.A.: Effective adjoint approaches for computational fluid dynamics. Prog. Aerosp. Sci. **110**, 100542 (2019) https://doi.org/10.1016/j.paerosci.2019.05.002

[59] Mader, C.A., Martins, J.R.R.A., Alonso, J.J., Weide, E.: ADjoint: An approach for the rapid development of discrete adjoint solvers. AIAA J. **46**(4), 863–873 (2008) https://doi.org/10.2514/1.29123

[60] McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979) https://doi.org/10.1080/00401706.1979.10489755

[61] Beck, A.: First-Order Methods in Optimization. SIAM, Philadelphia, PA (2017)

[62] Devolder, O.: Stochastic first-order methods in smooth convex optimization. Discussion Paper 2011/70, CORE, Université catholique de Louvain, Louvain-la-Neuve, Belgium (2011). Available at http://hdl.handle.net/2078.1/105693 (accessed 16 Sep 2025)

[63] Devolder, O.: Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization. PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgium (2013). Available at http://hdl.handle.net/2078.1/105693 (accessed 16 Sep 2025)

[64] Vasin, A., Gasnikov, A., Dvurechensky, P., Spokoiny, V.: Accelerated gradient methods with absolute and relative noise in the gradient. Optim. Methods Softw. **38**(6), 1180–1229 (2023) https://doi.org/10.1080/10556788.2023.2212503

[65] Bertsekas, D.P.: Nonlinear Programming, 3rd edn. Athena Scientific, Belmont, MA (2016)

[66] Forsgren, A., Gill, P.E., Murray, W.: Computing modified newton directions using a partial cholesky factorization. SIAM J. Sci. Comput. **16**(1), 139–150 (1995) https://doi.org/10.1137/0916009

[67] Berahas, A.S., Cao, L., Choromanski, K., Scheinberg, K.: A theoretical and empirical comparison of gradient approximations in derivative-free optimization. Found. Comput. Math. **22**(2), 507–560 (2022) https://doi.org/10.1007/s10208-021-09513-z

[68] Powell, M.J.D.: Algorithms for nonlinear constraints that use Lagrangian functions. Math. Program. **14**(1), 224–248 (1978) https://doi.org/10.1007/BF01588967