Engineering, 2012, 4, 390-393
doi:10.4236/eng.2012.47051 Published Online July 2012 (http://www.SciRP.org/journal/eng)

Optimality Conditions and Algorithms for Direct Optimizing the Partial Differential Equations

Victor K. Tolstykh
Computer Science Department, Donetsk National University, Donetsk, Ukraine
Email: tvk@dongu.donetsk.ua

Received April 20, 2012; revised May 17, 2012; accepted May 26, 2012

ABSTRACT

New form of necessary conditions for optimality (NCO) is considered. They can be useful for design the direct infinite-dimensional optimization algorithms for systems described by partial differential equations (PDE). Appropriate algorithms for unconstrained minimizing a functional are considered and tested. To construct the algorithms, new form of NCO is used. Such approach demonstrates fast uniform convergence at optimal solution in infinite-dimensional space.

Keywords: Optimization; Gradient; Necessary Conditions for Optimality; Partial Differential Equations; Infinite-Dimensional Algorithms

1. Introduction

Three classes of optimization problems for PDE are known, e.g., [1]: optimal control, parameter identification, and optimal design. To solve its in general case are used optimization algorithms in infinite-dimensional spaces, and finite-dimensional spaces. In the last case the algorithms are applied after transformation a desired parameter-function into a finite-dimensional space. We shall consider direct optimization [2,3], *i.e.* immediately minimization an objective functional $J(u)$ by infinite-dimensional methods on the basis of the gradient ∇J. Here $\nabla J(u;\tau)$ is a Frechet derivative, which is a linear functional. It depends on desirable parameter u and space-time variable τ.

It is well known classical NCO for unconstrained optimization problems:

$$\left\|\nabla J\left(u_*\right)\right\|_{U^*(S)} = 0 \tag{1}$$

where $u_*(\tau) \in U(S)$ is an optimum value of a desired parameter, $U(S)$ is a space of desired parameters defined on S, $U^*(S)$ is an adjoint space.

Because of computing errors the NCO (1) is never implemented. Approximate value of (1) is used sometimes for estimating a relative minimization of $J(u)$ in linear search problems. Sometimes approximate value of (1) is used as a completion criterion for optimization. No one uses NCO (1) for choosing a minimization direction in optimization algorithms.

We will consider NCO in a new form. It can be used for choosing a minimization direction for direct optimization algorithms.

2. Necessary Condition and Optimization Algorithm

2.1. Algorithm

For direct minimization approach the solution arg min $J(u)$ is searched on the basis of the algorithm

$$\begin{aligned} &u^{k+1}(\tau) = u^k(\tau) + b^k p\left(u^k;\tau\right), \\ &\tau \in S, \quad k = 0, 1, 2, \cdots, \end{aligned} \tag{2}$$

where direction $p\left(u^k;\tau\right) \equiv p^k \in U^*(S)$ is a linear functional representing the anti-gradient of the objective functional, here $p^k = -\nabla J^k$, or the conjugate gradients, e.g. Polak-Ribière (CG-PR) $p^k = -\nabla J^k + \beta^k p^{k-1}$, $\beta^k = \left\langle \nabla J^k, \left(\nabla J^k - \nabla J^{k-1}\right)\right\rangle \Big/ \left\|\nabla J^{k-1}\right\|^2$, b^k is a step-size.

Unfortunately, the optimizing by the algorithm (2) is not always possible. Even for a quadratic J there are no grounds of convergence for infinite-dimensional algorithm (2).

Let's replace (2) by the following algorithm:

$$\begin{aligned} &u^{k+1}(\tau) = u^k(\tau) + b^k \alpha^k(\tau) p\left(u^k;\tau\right), \\ &\tau \in S, \quad k = 0, 1, 2, \cdots, \end{aligned} \tag{3}$$

where $\alpha^k(\tau)$ is a function which regulates a convergence $u^k \to u_*$ on each iteration.

2.2. Necessary Condition

How correctly to set a function $\alpha^k(\tau)$ in (3)? Let's require: the algorithm (3) has to provide almost everywhere on S (a.e. S) convergence in an adjoint space U^*. Thus instead of integral NCO (1) we must to introduce the following NCO.

Theorem. *Let* $J(u)$ *be a smooth unconstrained functional, and it has a strict minimum at* u_*. *Then in some neighborhood of* u_* *the sequence* $u^k \to u_*$ *exists such, that*

$$\nabla J\left(u^k;\tau\right) \to 0 \quad \text{a.e.}\, S\,. \tag{4}$$

The singularity of introduced NCO (4) is that it is imposed on the gradient in vicinity of a minimum u_* instead of not exactly at u_* as it is presented in (1). Therefore the condition (4) can be used for constructing minimization steps near u_*. We are going to use new NCO (4) to set a function $\alpha^k(\tau)$ for algorithm (3).

The algorithm (3) with implementation of (4) allows us to solve *infinite-dimensional* optimization problems, under assumption that from a convergence a.e. S in an adjoint space U^* the similar convergence follows in a primal space U.

For a quantitative estimation of condition (4) let's introduce NCO-function

$$\eta^k(\tau) = \frac{\nabla J^{k-1}}{\left\|\nabla J^{k-1}\right\|}\operatorname{sign}\left\langle \nabla J^{k-1}, \nabla J^k \right\rangle - \frac{\nabla J^k}{\left\|\nabla J^k\right\|},\ k = 1, 2, \cdots$$

For this function, it is possible to write the NCO (4) in a more strong form

$$\left\|\eta^k\right\|_{U*} = 0 \quad \forall k > 0\,. \tag{5}$$

The NCO-Theorem with (5) instead of (4) requires decrease of function $\left|\nabla J\left(u^k;\tau\right)\right|$ not only a.e. S, but proportionally a.e. S for each iteration k under driving to $\min J$. The analogy in a finite-dimensional space for condition (5) denotes that the gradients vectors have to be collinear for all iterations up to u_* [4].

2.3. Implementation

The difficulty of practical implementation of method (3) is contained in a selection of function $\alpha^k(\tau)$ for satisfying the NCO (4) or (5). Consider one of methods for approximate implementing (5) on initial iterations.

We need to introduce a concept of template approximations. Let initial $u^0(\tau)$ and $\nabla J\left(u^0;\tau\right)$ known. Let's set the first approximation $u^1(\tau) = \varphi(\tau)$, where $\varphi(\tau)$ is a template function, for which the gradient $\nabla J(\varphi\,;\tau)$ satisfies to (5), *i.e.* proportionally decreases after the first iteration. Thus from (3) we can find, under $b^0 = 1$:

$$\alpha^0(\tau) = \left|\frac{\varphi(\tau) - u^0(\tau)}{\nabla J(u^0;\tau)}\right|,\ \ \nabla J\left(u^0;\tau\right) \neq 0\ \ \forall \tau \in S\,.$$

On the following iterations we set parameter $\alpha^k(\tau) = \alpha^0(\tau)$. In the given method from the researcher it is required to make some first experimental iterations for selecting an appropriate template function $\varphi(\tau)$, which satisfies to NCO (5).

We call your attention that the described method for $\alpha^k(\tau)$ can be applied to such u^0, that $\operatorname{sign}_\tau \nabla J\left(u^0;\tau\right) = \text{const}$, *i.e.* when $\nabla J\left(u^0;\tau\right) \neq 0$ for all $\tau \in S$.

2.4. Example

As an example we shall consider a one-dimensional linear parabolic heat equation in area $(t,x) \in \left[t_a, t_b\right] \times \left[x_0, x_1\right]$:

$$C\rho\frac{\partial T}{\partial t} - \lambda\frac{\partial^2 T}{\partial x^2} = 0, \tag{6}$$

$$\lambda\frac{\partial T}{\partial x}\bigg|_{x_0} = q,\ \ \lambda\frac{\partial T}{\partial x}\bigg|_{x_1} = u,\ \ T\big|_{t_a} = T_a$$

where $T(t,x)$ is a temperature, C, ρ, and λ is a thermal capacity, a density, and a thermal conduction accordingly. It is necessary to find a heat flow $u(t)$ on bound x_1 (set $S = \left(t_a, t_b\right) \times x_1$) that keeps a temperature T_* on other bound x_0 for given outflow q:

$$J(u) = \int_{t_a}^{t_b} \left(T - T_*\right)^2\Big|_{x_0}\,\mathrm{d}t \to \min \tag{7}$$

Applying the adjoint variables, we find the gradient

$$\nabla J(u;t) = -f(t,x) \quad \text{on} \quad S\,,$$

where $f(t,x)$ is a solution of the following adjoint problem

$$C\rho\frac{\partial f}{\partial t} + \lambda\frac{\partial^2 f}{\partial x^2} = 0,\ \ \lambda\frac{\partial f}{\partial x}\bigg|_{x_0} = 2\left(T - T_*\right)$$

$$\lambda\frac{\partial f}{\partial x}\bigg|_{x_1} = 0,\ \ f\big|_{t_b} = 0$$

The curve 1 on **Figure 1** illustrates unsuccessful attempt of solving the problem (6), (7) by infinite-dimensional algorithm (1) with direction p from method CG-PR. At initial approximation $u^0(t) = 400$ kJoule/(m^2·s) and optimal $u_*(t) = 350 + \left(t_a - t\right)350\big/\left(t_b - t_a\right)$ kJoule/(m^2·s) the gradient has very non-uniform value on segment $\left[t_a, t_b\right]$ (up to 7 orders) and, as a corollary, we obtain a non-uniform convergence to u_* by method CG-PR.

The attempt of solving the problem by finite-dimensional optimization algorithms has not given a positive outcome. Next we tried to do expansion of function $u(t) = \sum_{i=1}^{n-1} u_i B_i(t)$ through B-splines of zero order (piece-wise functions) with carriers equal to time-step $\Delta t = \left(t_b - t_a\right)\big/n$ as it was made in [5]. Given this, the finite-dimensional control $u \in R^n$, $n = 100$, was found by quasi-Newton method BFGS [6,7]. The solution coincided with the previous curve 1 on **Figure 1**.

All minimizing was finished under relative change of J and $\|u\|$ less than 1%. It is necessary to notice, that

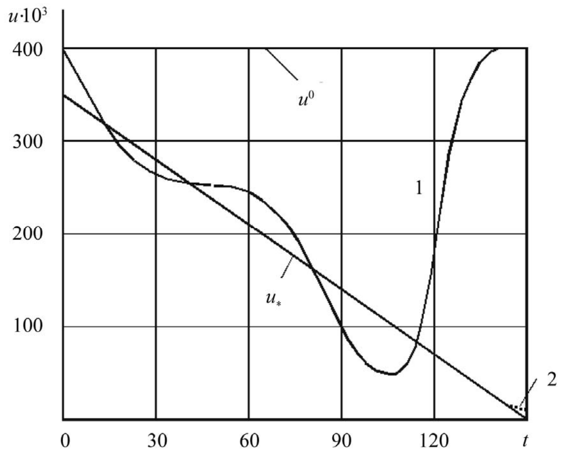

Figure 1. Solution of optimal control problem. 1: methods CG-PR and BFGS, 2: method (3), u^0: initial approximation, u_*: exact solution

the further iterating for method BFGS has allowed it to minimize J better than CG-PR. However, the curve 1 has varied not in essence. The outcomes speak that the optimization even with linear systems, which governed by PDE, is not always possible by traditional infinite-dimensional and finite-dimensional methods.

The curve 2 on **Figure 1** is a solution of problem (6), (7) by new infinite-dimensional method (3) under p chosen by method CG-PR with template function

$$\varphi = 0.2u^0 \tag{8}$$

The given function satisfies the NCO (4). We tried the second template function as:

$$\varphi(t) = 0.036u^0\left(1 + 8\left(t - t_a\right)\big/\left(t_b - t_a\right)\right) \tag{9}$$

It satisfies the strong NCO (5). Here solution has coincided with u_* precisely on **Figure 1**.

To select a function $\varphi(t)$ we analyze a behavior of function $\eta(t)$. A value of this function for all methods on the first experimental step $\left\|u^1 - u^0\right\| = 0.2\left\|u^0\right\|$ is shown in **Figure 2**. We see, that the classical methods CG-PR, BFGS (see the curve 1) realize the new NCO badly, to be exact, they do not implement its. Method (3) with φ in (8) (see the curve 2) not bad implements NCO (4), but does not implement strong NCO (5). Method (3) with φ in (9) (see the curve 3) implements strong NCO (5) and provides convergence to exact solution u_* better all especially on the first iterations.

It is necessary to tell, that the template functions φ in (8) and (9) give noticeably different minimization outcomes only on the first iteration. With growth of iterations they give approximately equal good outcomes. It is explained to that the parameters $\alpha^k(t)$, regulating a descent in method (3), are computed with the account of NCO only on the first step. For discussed method $\alpha^k(t) = \alpha^0(t)$.

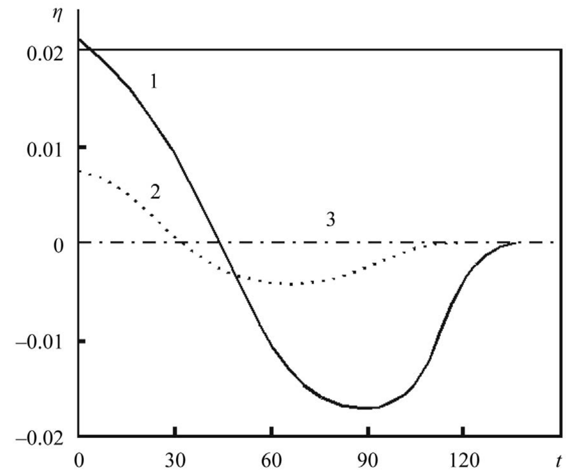

Figure 2. NCO-function η(t) for first experimental step. 1: method CG-PR; 2: method (3) with NCO (4); 3: method (3) with strong NCO (5).

Table 1. Initial and final values of the objective functional, the proximity to an exact solution, and strict NCO (on a first experimental step).

Method	Iteration k	Functional J^k	$\left\|u^k - u_*\right\|$	$\left\|\eta\right\|$
All	0	1.86×10^4	2.97×10^6	
CG-PR	14	2.62	1.73×10^6	1.32×10^{-1}
BFGS	10	2.72	1.74×10^6	1.32×10^{-1}
(3), (8)	54	2.12×10^{-4}	4.31×10^4	4.71×10^{-2}
(3), (9)	52	1.06×10^{-4}	7.85×10^3	1.82×10^{-4}

Everywhere for searching a step-size b^k, the method Wolfe with quadratic interpolation was used (Wright, Nocedal, 1999). Here step-size was computed from conditions

$$\begin{aligned} &J\left(u^k + b^k p^k\right) \le J\left(u^k\right) + c_1 b^k \left\langle \nabla J^k, p^k \right\rangle, \\ &\left|\left\langle \nabla J\left(u^k + b^k p^k\right), p^k \right\rangle\right| \le c_2 \left|\left\langle \nabla J^k, p^k \right\rangle\right| \end{aligned} \tag{10}$$

The parameters of a method were given $c_1 = 10^{-4}$ and $c_2 = 0.1$.

In the **Table 1** are shown the obtained values of the objective functional, the proximity to exact solution, and NCO (5) (on a first step) for all methods. From outcomes of computations it is seen, that the new method on the basis of algorithm (3) with NCO (5) minimizes the functional J on 4 orders better than the traditional methods. The method has allowed us to approach to optimal solution u_* on 3 orders closer.

3. Conclusion

Thus, the new NCO has appeared effective for constructing the algorithms of direct optimization for processes which governed by PDE. The algorithm (3) with NCO (4) or (5) can be recommended to solving the infinite-dimensional optimization problems.

REFERENCES

[1] N. M. Alexandrov, "Optimization of Engineering Systems Governed by Differential Equations," *SIAG/OPT Views & News*, Vol. 11, No. 2, 2000, pp. 1-4.

[2] R. M. Lewis, "The Adjoint Approach in a Nutshell," *SIAG/OPT Views & News*, Vol. 11, No. 2, 2000, pp. 9-12.

[3] V. K. Tolstykh, "Direct Extreme Approach for Optimization of Distributed Parameter Systems," Yugo-Vostok, Donetsk, 1997.

[4] V. K. Tolstykh, "New First-Order Algorithms for Optimal Control under Large and Infinite-Dimensional Objective Functions New First-Order Algorithm for Optimal Control with High Dimensional Quadratic Objective," *Proceedings of the* 16*th IMACS World Congress on Scientific Computation*, *Applied Mathematics and Simulation*, Lausanne, 21-25 August 2000. http://metronu.ulb.ac.be/imacs/lausanne/CP/215-3.pdf

[5] R. H. W. Hoppe and S. I. Petrova, "Applications of the Newton Interior-Point Method for Maxwell's Equations," *Proceedings of the* 16*th IMACS World Congress on Scientific Computation*, *Applied Mathematics and Simulation*, Lausanne, 21-25 August 2000. http://metronu.ulb.ac.be/imacs/lausanne/SP/107-7.pdf

[6] C. T. Kelly, "Iterative Methods for Optimization," SIAM, Philadelphia, 1999. doi:10.1137/1.9781611970920

[7] J. Nocedal, S. J. Wright, "Numerical Optimization," Springer, New York, 1999. doi:10.1007/b98874

