KYBERNETIKA — VOLUME 55 (2019), NUMBER 1, PAGES 183–202

A NONMONOTONE LINE SEARCH FOR THE LBFGS METHOD IN PARABOLIC OPTIMAL CONTROL PROBLEMS

Omid Solaymani Fard, Farhad Sarani, Akbar Hashemi Borzabadi and Hadi Nosratipour

In this paper a nonmonotone limited memory BFGS (NLBFGS) method is applied for approximately solving optimal control problems (OCPs) governed by one-dimensional parabolic partial differential equations. A discretized optimal control problem is obtained by using piecewise linear finite element and well-known backward Euler methods. Afterwards, regarding the implicit function theorem, the optimal control problem is transformed into an unconstrained nonlinear optimization problem (UNOP). Finally the obtained UNOP is solved by utilizing the NLBFGS method. In comparison to other existing methods, the NLBFGS method shows a significant improvement especially for nonlinear and ill-posed control problems.

Keywords: optimal control, parabolic partial differential equations, backward Euler method, nonmonotone LBFGS method

Classification: 65K10, 90C30, 90C53

1. INTRODUCTION

In recent years, optimal control problems governed by partial differential equations (PDEs) are widely applied in finance [1, 9], biology [3, 7] and industrial [14, 16, 22]. Thus for such optimization problems, the development of efficient optimization techniques is required. In the past years, some monographs have been edited by researchers to various aspects of the optimal control of PDEs [19, 33].

The purpose of this paper is to apply an optimization method for approximately solving one-dimensional parabolic optimal control problems. Consider the infinite dimensional optimization problem given by

$$\min_{y,u} J(y,u), \tag{1}$$

subject to

$$E(y,u) = 0, \tag{2}$$

DOI: 10.14736/kyb-2019-1-0183

where $J(y,u)$ is a nonnegative quadratic cost functional and $E(y,u) = 0$ is a one-dimensional parabolic PDE that represents the equality constraint. Also, the variables u and y are respectively called the control and the state variables and (2) is called the state equation.

Numerical methods for solving optimal control problems fall into two main classes: *direct methods* and *indirect methods* . In direct methods, the control problem is first discretized and then a nonlinear programming (NLP) technique is applied to the resulted optimization problem. In indirect methods, one can try to approximate the solution to the necessary optimality conditions. In this work, a direct method is used for parabolic control problems. In order to discretize such problems in space, the piecewise linear finite element method is used to construct ordinary differential equations (ODEs) system. Then, the backward Euler method is applied for discretizing the problem in time to attain the fully discretized problem. The obtained finite dimensional problem is given as follows

$$\min_{\hat{y},\hat{u}} J(\hat{y},\hat{u}), \tag{3}$$

subject to

$$E(\hat{y},\hat{u}) = 0, \tag{4}$$

where

$$J(\hat{y},\hat{u}) : \mathbb{R}^{(n_y+n_u)} \longrightarrow \mathbb{R}, \qquad E(\hat{y},\hat{u}) : \mathbb{R}^{(n_y+n_u)} \longrightarrow \mathbb{R}^{n_y}.$$

Here, n_y and n_u denote the number of state and control variables, respectively. Regarding the equality-constrained problem of the form (3) – (4), the implicit function theorem allows us to consider the state variable $\hat{y}$ as a function depending on $\hat{u}$, that is, $\hat{y} = \hat{y}(\hat{u})$ in a neighborhood of the solution such that $E(\hat{y},\hat{u}) = 0$ [5, 14]. That means it can be formulated as the reduced problem

$$\min \ \hat{J}(\hat{u}), \quad \text{subject to } \hat{u} \in \mathbb{R}^{n_u}, \tag{5}$$

where $\hat{J}(\hat{u}) := J(\hat{y}(\hat{u}),\hat{u})$ is the so-called reduced objective.

There are many iterative methods for solving the problem (5). We refer to the general literatures on numerical optimization such as [4, 25] for more details. In this paper, the NLBFGS method developed by Amini et al. [2] is used to find the approximate solutions of the reduced control problem (5). In nonmonotone line search methods, some growth in the function value is permitted. As pointed out by many researchers [2, 17, 35, 37], nonmonotone schemes can improve the likelihood of finding a global optimum, as well as the convergence rate in cases where a monotone scheme is forced to creep along the bottom of a narrow curved valley. Encouraging numerical results have been reported for highly nonlinear and ill-conditioned problems by using nonmonotone schemes. Also, a nonmonotone scheme can be very effective when combined with a Newton-type direction.

The standard limited memory BFGS (LBFGS) method is applied to the optimal control of the viscous Burgers equation by Heinkenschloss [13], and it is shown that the LBFGS method does not work well for this type problems due to the failure of the Armijo line search rule used for computing an appropriate step size. Many PDE

constrained optimization problems have nonlinear and ill-posed structures. Because of these properties, using the nonmonotone schemes is more reliable for these problems.

The above discussion motivates us to apply a nonmonotone version of the LBFGS method (NLBFGS) in the line search framework to find the approximate solutions of the parabolic optimal control problems.

The paper is organized as follows. The existence of an optimal solution and optimality conditions are presented in Section 2. Section 3 is devoted to the discretization techniques for the parabolic control problems. In Section 4, the LBFGS method to generate the search direction and an adaptive nonmonotone line search to compute the step size are described, and then Algorithm 1 is presented. Numerical results are given in Section 5. The conclusions are discussed in the final section.

2. EXISTENCE OF AN OPTIMAL SOLUTION AND OPTIMALITY CONDITIONS

Consider the problem (3) – (4) and make the following assumption.

Assumption 2.1.

1. There exists an open set $D \subseteq \mathbb{R}^{(n_y+n_u)}$ with $\{(\hat{y},\hat{u}) : (\hat{y},\hat{u}) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_u},\ E(\hat{y},\hat{u}) = 0\} \subseteq D$ that J and E are twice continuously differentiable on D.

2. The constraint $E(\hat{y},\hat{u}) = 0$ has for any given $\hat{u} \in \mathbb{R}^{n_u}$ a unique solution $\hat{y} \in \mathbb{R}^{n_y}$.

3. $E_{\hat{y}}(\hat{y},\hat{u})$, the partial Jacobian of the function E with respect to $\hat{y}$, is bijective for all $(\hat{y},\hat{u}) \in \{(\hat{y},\hat{u}) : (\hat{y},\hat{u}) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_u},\ E(\hat{y},\hat{u}) = 0\}$.

Definition 2.2. A state-control pair $(y^*,u^*) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_u}$ is called optimal for (3) – (4), if $E(y^*,u^*) = 0$ and

$$J(y^*,u^*) \leq J(\hat{y},\hat{u}) \quad \forall (\hat{y},\hat{u}) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_u}, \quad E(\hat{y},\hat{u}) = 0.$$

Since $E_{\hat{y}}(\hat{y},\hat{u})$ is bijective, the implicit function theorem [11] yields the existence of the implicit function $\hat{y} = \hat{y}(\hat{u})$ such that $E(\hat{y},\hat{u}) = 0$. This implicit function is well-defined in a neighborhood of the solution (y^*,u^*). Now, we introduce a theorem for existence of an optimal control u^*.

Theorem 2.3. Let Assumption 2.1 hold. If $\hat{J} : \mathbb{R}^{n_u} \longrightarrow \mathbb{R}$ is bounded from below and the level set $L_\theta := \{\hat{u} \in \mathbb{R}^{n_u} | \hat{J}(\hat{u}) \leq \theta\}$ is a nonempty and bounded set for some $\theta < \infty$, then problem (5) has an optimal solution u^*.

P r o o f . Since $\hat{J}$ is bounded from below, the infimum of $\hat{J}(\hat{u})$, $\hat{u} \in \mathbb{R}^{n_u}$ exists. Let

$$J^* := \inf_{\hat{u} \in \mathbb{R}^{n_u}} \hat{J}(\hat{u}),$$

hence a minimizing sequence $\{\hat{u}_k\} \subseteq \mathbb{R}^{n_u}$ can be found with

$$\lim_{k \to \infty} \hat{J}(\hat{u}_k) = J^*.$$

For all k greater than some number $K < \infty$, all $\hat{u}_k$ have to lie in L_θ with $\theta \geq J^*$. L_θ is closed because of the continuity of $\hat{J}$ (which is implied by Assumption 2.1). Since L_θ is closed and bounded, then it is compact. Thus, there exists a convergent subsequence $\{\hat{u}_{k_p}\} \subseteq \mathbb{R}^{n_u}$ with $\hat{u}_{k_p} \longrightarrow u^*$ as $p \longrightarrow \infty$. Since $\hat{J}$ is continuous, u^* is a minimum, because

$$J^* = \lim_{p \longrightarrow \infty} \quad \hat{J}(\hat{u}_{k_p}) = \hat{J}(u^*).$$

□

2.1. Lagrangian approach and optimality conditions

Consider the problem (3)-(4). The Lagrangian function is given by

$$L(\hat{y}, \hat{u}, p) = J(\hat{y}, \hat{u}) + p^T E(\hat{y}, \hat{u}), \tag{6}$$

where $p \in \mathbb{R}^{n_y}$ is called a Lagrange multiplier for the constraint $E(\hat{y}, \hat{u}) = 0$.

Theorem 2.4. (**Fermat**) Let $u^* \in \mathbb{R}^{n_u}$ be a local minimizer of $\hat{J} : \mathbb{R}^{n_u} \longrightarrow \mathbb{R}$ and $y(u^*)$ its associated state. If Assumption 2.1 hold, then there exists a Lagrange multiplier $p \in \mathbb{R}^{n_y}$ such that the following system of equations is satisfied

$$\nabla_y L(y(u^*), u^*, p) = 0 \implies \nabla_y J(y(u^*), u^*) + E_y^T(y(u^*), u^*)p = 0, \tag{7a}$$

$$\nabla_u L(y(u^*), u^*, p) = 0 \implies \nabla \hat{J}(u^*) = \nabla_u J(y(u^*), u^*) + E_u^T(y(u^*), u^*)p = 0, \tag{7b}$$

$$\nabla_p L(y(u^*), u^*, p) = 0 \implies E(y(u^*), u^*) = 0. \tag{7c}$$

P r o o f . See [10]. □

System (7) is called the optimality system for u^*.

3. PARABOLIC OPTIMAL CONTROL PROBLEMS AND THEIR DISCRETIZATION PROCEDURE

Consider the following one-dimensional Burgers equation

$$y_t(x,t) - \nu y_{xx}(x,t) + y y_x(x,t) = u(x,t) \qquad (x,t) \in (0,1) \times (0,t_f), \tag{8}$$

with the initial condition

$$y(x,0) = f(x) \qquad x \in (0,1), \tag{9}$$

and the boundary conditions

$$y(0,t) = y(1,t) = 0 \qquad t \in (0,t_f), \tag{10}$$

and furthermore the cost functional given by

$$J(y,u) = \frac{1}{2} \int_0^{t_f} \int_0^1 ((y(x,t) - z(x,t))^2 + \beta u^2(x,t)) \, \mathrm{d}x\mathrm{d}t. \tag{11}$$

Here $z(x,t), f(x)$ are given functions and $\beta, \nu > 0$ are given parameters. Also, t_f denotes the final time. The control problem (8) – (11) is a nonlinear parabolic optimal control problem and is widely applied in flow control [23]. Recently, many efforts have been devoted to the development of the optimal control techniques for the Burgers equation [18, 26, 31, 36]. For a given initial condition $f \in L^2(0,1)$ and for the control $u \in L^2((0,1) \times (0,t_f))$, the verification that the control problem (8) – (11) satisfies the Assumption 2.1, especially the first and second one, is discussed in [34].
To discretize the control problem (8) – (11) in space, the finite element method is used. The finite element method and its extensions are among the most powerful computational tools for solving complex ordinary and partial differential equations. The references [6], [12] and [32] provide an excellent introduction and examples where the finite element method has been applied to a variety of areas in science and engineering. For the control problem (8) – (11), the finite element discretization in space is discussed by [13]. Consider a partition of $[0,1]$ into n subinterval $[x_{i-1}, x_i], i = 1, \ldots, n$, with $x_i = ih$ and $h = \frac{1}{n}$. Using the finite element discretization in space with n uniform subdivisions with the step size $h = \frac{1}{n}$ for state and control variables, the control problem (8) – (11) is transcribed into the following ordinary optimal control problem

$$\min_{\vec{u}} \int_0^{t_f} (\frac{1}{2}(\vec{y}(t)^T - \vec{z}(t)^T)A_h(\vec{y}(t) - \vec{z}(t)) + \frac{\beta}{2}\vec{u}(t)^T B_h \vec{u}(t))\,\mathrm{d}t, \tag{12}$$

where $\vec{y}(t) = (y_1(t), \ldots, y_{n-1}(t))^T$ is the solution of

$$A_h \frac{\mathrm{d}}{\mathrm{d}t}\vec{y}(t) + C_h \vec{y}(t) + N_h(\vec{y}(t)) + D_h \vec{u}(t) = 0, \quad t \in (0, t_f), \tag{13}$$

$$\vec{y}(0) = \vec{y}_0 = \vec{f},$$

where $\vec{u}(t) = (u_0(t), \ldots, u_n(t))^T$, $\vec{y}_0 = (y_0(x_1), \ldots, y_0(x_{n-1}))$ and $\vec{f} = (f(x_1), \ldots, f(x_{n-1}))$. Also $B_h \in \mathbb{R}^{(n+1)\times(n+1)}$, $A_h, C_h \in \mathbb{R}^{(n-1)\times(n-1)}$, $D_h \in \mathbb{R}^{(n-1)\times(n+1)}$ are matrices and $N_h(\vec{y}(t)), \vec{z}(t) \in \mathbb{R}^{(n-1)}$ are vectors.

The backward Euler method is applied for time discretization. The backward Euler method is an implicit method which contrary to explicit methods ensures both unconditional stability and unconditional positivity. Because of these properties, the backward Euler method is used for discretizing the problem (12) – (13) in time. Let

$$0 = t_0 < t_1 < \cdots < t_{N+1} = t_f$$

and define

$$\Delta t_i = t_{i+1} - t_i, \qquad i = 0, \ldots, N.$$

The fully discretized problem is given as follows

$$\min_{\vec{u}_0,\ldots,\vec{u}_{N+1}} J(\vec{y}, \vec{u}) = \sum_{i=0}^{N+1} \frac{\Delta t_{i-1} + \Delta t_i}{2} \left(\frac{1}{2}(\vec{y}_i^T - \vec{z}_i^T)A_h(\vec{y}_i - \vec{z}_i) + \frac{\beta}{2}\vec{u}_i^T B_h \vec{u}_i \right), \tag{14}$$

where $\Delta t_{-1} := \Delta t_{N+1} := 0$ and $\vec{y}_1, \ldots, \vec{y}_{N+1}$ is the solution of

$$(A_h + \Delta t_i C_h)\vec{y}_{i+1} + \Delta t_i N_h(\vec{y}_{i+1}) + \Delta t_i D_h \vec{u}_{i+1} - A_h \vec{y}_i = 0, \quad i = 0, \ldots, N, \tag{15}$$

where $\vec{y}_0 = \vec{f} = (f(x_1), \ldots, f(x_{n-1}))$. Also, we set $\vec{u} = (\vec{u_0}^T, \ldots, \vec{u}_{N+1}^T)^T$.

Remark 3.1. Discretization procedure to the one-dimensional linear parabolic control problems is similar to the previous case. For instance, consider the following diffusion equation

$$y_t(x,t) = y_{xx}(x,t) + u(x,t), \tag{16}$$

with the initial condition

$$y(x,0) = g(x), \quad 0 \le x \le x_f, \tag{17}$$

and the boundary conditions

$$y_x(0,t) = y_x(x_f,t) = 0, \qquad 0 \le t \le t_f, \tag{18}$$

and furthermore the cost functional given by

$$J(y,u) = \frac{1}{2}\int_0^{t_f}\int_0^{x_f} ((y(x,t) - z(x,t))^2 + \beta u^2(x,t))\,\mathrm{d}x\mathrm{d}t. \tag{19}$$

Here $z(x,t), g(x)$ are given functions and $\beta > 0$ is a regularization parameter. As before, t_f denotes the final time. For the control problem (16) – (19), the fully discretized problem is given as follows

$$\min_{\vec{u_0},\ldots,\vec{u}_{N+1}} J(\vec{y},\vec{u}) = \sum_{i=0}^{N+1} \frac{\Delta t_{i-1} + \Delta t_i}{2}\Big(\frac{1}{2}(\vec{y}_i^T - \vec{z}_i^T)B_h(\vec{y}_i - \vec{z}_i) + \frac{\beta}{2}\vec{u}_i^T B_h \vec{u}_i\Big), \tag{20}$$

where $\Delta t_{-1} := \Delta t_{N+1} := 0$ and $\vec{y}_1, \ldots, \vec{y}_{N+1}$ is the solution of

$$(B_h + \Delta t_i F_h)\vec{y}_{i+1} + \Delta t_i G_h \vec{u}_{i+1} - B_h \vec{y}_i = 0, \quad i = 0,\ldots,N, \tag{21}$$

where $\vec{y}_0 = \vec{g} = (g(x_0),\ldots,g(x_n))$. Also $\vec{u} = (\vec{u_0}^T,\ldots,\vec{u}_{N+1}^T)^T$, $B_h \in \mathbb{R}^{(n+1)\times(n+1)}$ is defined as before, $F_h, G_h \in \mathbb{R}^{(n+1)\times(n+1)}$ are matrices and $\vec{z}(t) \in \mathbb{R}^{(n+1)}$ is a vector.

4. A NONMONOTONE LBFGS METHOD FOR THE OBTAINED DISCRETIZED CONTROL PROBLEMS

For solving the fully discretized problems (14) – (15) and (20) – (21) that can be converted to the unconstrained optimization problem (5), each iteration of a line search method computes a search direction d_k and then decides how far to move along that direction. Most line search algorithms require d_k to be a descent direction because this property guarantees that the function $\hat{J}$ can be reduced along this direction. Moreover, the search direction often has a Newton-type form

$$d_k = -H_k \nabla \hat{J}(\hat{u}_k), \tag{22}$$

where H_k is a quasi-Newton approximation of the inverse matrix $\nabla^2 \hat{J}(\hat{u}_k)$. The most popular quasi-Newton algorithm is the BFGS method, named for its discoverers Broyden, Fletcher, Goldfarb and Shanno [25]. The BFGS method computes H_k by

$$H_{k+1} = (I - \rho_k s_k \omega_k^T) H_k (I - \rho_k \omega_k s_k^T) + \rho_k s_k s_k^T, \tag{23}$$

where $s_k = \hat{u}_{k+1} - \hat{u}_k$, $\omega_k = \nabla \hat{J}(\hat{u}_{k+1}) - \nabla \hat{J}(\hat{u}_k)$ and $\rho_k = 1/\omega_k^T s_k$.

The BFGS method is not useful for solving large problems because Hessian matrices can not be computed at a reasonable cost or are not sparse. For this reason, the limited memory BFGS (LBFGS) method can be an appropriate choice. This method (see [20, 24]) is an adaptation of the BFGS method for large-scale problems. The implementation is almost identical to that of the standard BFGS method, the only difference is that the inverse Hessian approximation is not formed explicitly, but defined by a small number of BFGS updates. It often provides a fast rate of linear convergence, and requires minimal storage. Hence, it is in principal suitable for large-scale optimization problems. The main idea of the LBFGS method which is based on the BFGS updating formula is to use the curvature information from only the most recent iterations to construct the Hessian approximation. The curvature information from earlier iterations, which is less likely to be relevant to the actual behavior of the Hessian at the current iteration, is discarded in the interest of saving storage.

Let H_0 be a symmetric and positive definite starting matrix and $m = \min\{k, 5\}$. Then the limited memory version of H_k is defined by

$$\begin{aligned} H_{k+1} = {} & (V_k^T \cdots V_{k-m}^T) H_0 (V_{k-m} \cdots V_k) \\ & + \rho_{k-m} (V_k^T \cdots V_{k-m+1}^T) s_{k-m} s_{k-m}^T (V_{k-m+1} \cdots V_k) \\ & + \rho_{k-m+1} (V_k^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \cdots V_k) \\ & \vdots \\ & + \rho_k s_k s_k^T, \end{aligned} \tag{24}$$

where $V_k = I - \rho_k y_k s_k^T$. The L-BFGS code is available from the web page:
`http://users.eecs.northwestern.edu/~nocedal/software.html#lbfgs`.
We re-write this code in MATLAB and exploit it to generate the search direction d_k.

After computing the direction d_k, a nonmonotone Armijo line search procedure is used to compute the step size α_k. The nonmonotone globalization technique is useful in difficult nonlinear problems, because of the fact that it may help escaping from steep sided valleys and may improve both the possibility of finding the global optimum and the rate of convergence [2, 27, 28]. Amini et al. [2] introduced the nonmonotone term as follows

$$N_k = \eta_k \hat{J}_{l(k)} + (1 - \eta_k) \hat{J}_k, \tag{25}$$

where $\eta_k \in [0, 1)$ and

$$\hat{J}_{l(k)} = \max_{0 \le j \le r(k)} \{\hat{J}_{k-j}\}, \quad k = 0, 1, 2, \cdots, \tag{26}$$

where $r(0) = 0$ and $0 \le r(k) \le min\{r(k-1) + 1, M\}$ with $M \ge 0$ and in the following, a new nonmonotone Armijo-type line search proposed as follows

$$\hat{J}(\hat{u}_k + \alpha_k d_k) \le N_k + \delta \alpha_k \nabla \hat{J}^T(\hat{u}_k) d_k. \tag{27}$$

It is obvious that when η_k is close to 1, one can obtain a strong nonmonotone strategy and get a weaker nonmonotone strategy when η_k is close to 0. Hence, by choosing η_k

adaptively, one can increase the affect of $\hat{J}_{l(k)}$ far from the solution and decrease it near to the solution. Considering the morphology of the function, the best criterion to measure the closeness of the current point $\hat{u}_k$ to the solution $\hat{u}^*$ is to assess the first-order optimality condition, so $\|\nabla \hat{J}(\hat{u}_k)\|$ can be used as a criteria for the closeness to the solution. Therefore, the parameter η_k can be updated adaptively by

$$\eta_k = 1 - e^{-||\nabla \hat{J}(\hat{u}_k)||}. \tag{28}$$

It is clear that when $||\nabla \hat{J}(\hat{u}_k)||$ is major, then η_k will also be major, therefore the nonmonotone strategy will be stronger. On the other hand, when $||\nabla \hat{J}(\hat{u}_k)||$ tends toward zero, then η_k tends to zero, so, nonmonotone strategy will be weaker and tend to be a monotone stategy.

Now, we can outline Algorithm 1 as the LBFGS method equipped with the nonmonotone Armijo line search rule for the discretized parabolic optimal control problems.

Algorithm 1: The LBFGS method with nonmonotone Armijo line search (NLBFGS)

Input: Given $\hat{u}_0 \in \mathbb{R}^{n_u}$, $\sigma, \delta \in (0,1), m, M, \epsilon_1 > 0$ and a symmetric positive definite matrix H_0.

Begin

$k \leftarrow 0$;

Compute $\nabla \hat{J}(\hat{u}_k)$;

While ($\|\nabla \hat{J}(\hat{u}_k)\| \geq \epsilon_1$) **{Start of outer loop}**

{Determination of search direction}

Generate a descent direction d_k using (22) and (24);

Set $\alpha_k = 1$ and evaluate $\hat{J}(\hat{u}_k + \alpha_k d_k)$;

While $\hat{J}(\hat{u}_k + \alpha_k d_k) > N_k + \delta \alpha_k d_k^T \nabla \hat{J}(\hat{u}_k)$ **{Start of backtraking loop}**

$\alpha_k \leftarrow \sigma \alpha_k$ and evaluate $\hat{J}(\hat{u}_k + \alpha_k d_k)$.

End While {End of inner loop}

$\hat{u}_{k+1} \leftarrow \hat{u}_k + \alpha_k d_k$;

$k \leftarrow k + 1$;

Compute $\nabla \hat{J}(\hat{u}_k)$;

Compute $\hat{J}_{l(k)}$, η_k and N_k by (26), (28) and (25), respectively;

End While {End of outer loop}

End

Now, by applying Theorem 2.3 for the discretized problem (14) – (15), we obtain an algorithm for computing $\nabla \hat{J}(\hat{u})$ in Algorithm 1.

Algorithm 2: Computation of $\nabla \hat{J}(u)$ for the discretized problem (14) – (15)

1- Given $\vec{u}_0, \ldots, \vec{u}_{N+1}$ and $\vec{y}_0$, compute $\vec{y}_1, \ldots, \vec{y}_{N+1}$ by solving

$$(A_h + \Delta t_i C_h)\vec{y}_{i+1} + \Delta t_i N_h(\vec{y}_{i+1}) = -\Delta t_i D_h \vec{u}_{i+1} + A_h \vec{y}_i, \quad i = 0, \ldots, N.$$

2- Compute $\vec{p}_1, \ldots, \vec{p}_{N+1}$ by solving

$$(A_h + \Delta t_N C_h + \Delta t_N N_h'(\vec{y}_{N+1}))^T \vec{p}_{N+1} = -\frac{\Delta t_N}{2} A_h(\vec{y}_{N+1} - \vec{z}_{N+1}),$$

$$(A_h + \Delta t_{i-1} C_h + \Delta t_{i-1} N_h'(\vec{y}_i))^T \vec{p}_i = A_h^T \vec{p}_{i+1} - \frac{\Delta t_{i-1} + \Delta t_i}{2} A_h(\vec{y}_i - \vec{z}_i), \quad i = N, \ldots, 1,$$

where $N_h'(\vec{y}_i)$ denotes the Jacobian of $N_h(\vec{y}_i)$.
3- Compute $\nabla \hat{J}(u)$ from below

$$\nabla \hat{J}(u) = \begin{bmatrix} \beta \frac{\Delta t_0}{2} B_h \vec{u}_0 \\ \beta \frac{\Delta t_0 + \Delta t_1}{2} B_h \vec{u}_1 + {D_h}^T(\Delta t_0 \vec{p}_1) \\ . \\ . \\ . \\ \beta \frac{\Delta t_{i-1} + \Delta t_i}{2} B_h \vec{u}_i + {D_h}^T(\Delta t_{i-1} \vec{p}_i) \\ . \\ . \\ . \\ \beta \frac{\Delta t_{N-1} + \Delta t_N}{2} B_h \vec{u}_N + {D_h}^T(\Delta t_{N-1} \vec{p}_N) \\ \beta \frac{\Delta t_N}{2} B_h \vec{u}_{N+1} + {D_h}^T(\Delta t_N \vec{p}_{N+1}) \end{bmatrix}.$$

Remark 4.1. In step 2 of Algorithm 2, the determinant of the matrix of coefficients is nonzero and thus this step satisfies the third part of Assumption 2.1. Also, the computation of $\nabla \hat{J}(\hat{u})$ for the discretized problem (20) – (21) is similar to Algorithm 2 and so is omitted.

5. COMPUTATIONAL RESULTS

This section reports numerical results obtained by testing the proposed algorithm, NLBFGS, compared with different approximate methods. Numerical experiments are performed in double precision arithmetic format in MATLAB 8.2 programming environment. Since the proposed algorithm equipped an nonmonotone Armijo line search does not guarantee that $y_k^T s_k > 0$. Hence, to generate this condition, we will skip the new update if $y_k^T s_k > 0$ is not satisfied. We set $M = 10$ to calculate the nonmonotone term (26). If (4) is nonlinear in $\hat{y}$, then (4) must be solved using an iterative method such that $\|E(\hat{y}, \hat{u})\|_2 < \xi$. Because of this reason, the system (15)is solved by applying an inexact Newton's method with a stopping tolerance of $\xi = 10^{-2} \min\{{h_i}^2, {\Delta t_j}^2\}$ where $i = 1, \ldots, n$, and $j = 0, \ldots, N$. We also use primary functions in MATLAB programming environment for solving the linear systems that occur during the comparison of the performances of LBFGS and NLBFGS methods. The parameters used in the algorithm NLBFGS are specified as follows:

$$\hat{u}_0 = 0, \quad H_0 = I, \quad \sigma = \tfrac{1}{2}, \quad \delta = 10^{-4}.$$

Example 1. Consider the control problem (16) – (19) with

$$x_f = 4, \quad t_f = 1, \quad \beta = 1, \quad g(x) = 1 + x, \quad z(x,t) = 0.$$

In Tables 1 and 2, the numerical results obtained by NLBFGS method are compared with LBFGS method. These results are obtained with $m = 5, 25$ and stopping criterion $\epsilon_1 = 10^{-6}$. In Table 2, the bold **(LF)** means that the line search fails because no sufficient decrease could be detected for computing an appropriate step size and stopping criterion $\epsilon_1 = 10^{-6}$ is not satisfied. From Table 2, it is also evident that both methods become weaker, especially for smaller values of $0 < \beta < 1$. The results show that as the value of the β parameter decreases, then the effectiveness of both methods is reduced. Also, for more clarity, $\|\nabla \hat{J}(\hat{u}_k)\|$ for all iterations with $\beta = 10^{-3}$ and $m = 25$ is given in Figure 1. The residuals values of optimal state $y^*(2,t)$ and optimal control $u^*(2,t)$, for $\beta = 1$, obtained by the present method with $n = 32$ and $N = 1024$, compared with different approximate methods by surnames radial basis functions (RBF method) [29], shift Chebyshev polynomial (SCP method) [15], shift Legendre polynomial (SLP method) [8] and Taylor series (TS method) [30] are shown in Tables 3 and 4. The resulting graphs of optimal state $y^*(x,t)$ and optimal control $u^*(x,t)$ for $\beta = 1$ and $\beta = 0.001$ are shown in Figures 2 and 3, respectively. Furthermore, Table 5 shows the values of the residual error for the discretized form of (16), $\|R(x,t)\|_2$, for various meshes.

Mesh	Item	Method with $m = 5$		Method with $m = 25$	
		NLBFGS	LBFGS	NLBFGS	LBFGS
8×64	Iteration	5	5	5	5
	$\hat{J}$	1.499437e+01	1.499437e+01	1.499437e+01	1.499437e+01
	$\|\nabla \hat{J}\|_2$	2.248813e-07	2.248813e-07	2.248813e-07	2.248813e-07
	Time	0.488308	0.490453	0.845513	0.849548
16×256	Iteration	5	5	5	5
	$\hat{J}$	1.499860e+01	1.499860e+01	1.499860e+01	1.499860e+01
	$\|\nabla \hat{J}\|_2$	8.199877e-08	8.199877e-08	8.199877e-08	8.199877e-08
	Time	0.868159	0.888479	1.237915	1.300276
32×1024	Iteration	4	4	4	4
	$\hat{J}$	1.499987e+01	1.499987e+01	1.499987e+01	1.499987e+01
	$\|\nabla \hat{J}\|_2$	5.342866e-07	5.342866e-07	5.342866e-07	5.342866e-07
	Time	2.179626	2.197954	2.484486	2.714798
64×4096	Iteration	4	4	4	4
	$\hat{J}$	1.500020e+01	1.500020e+01	1.500020e+01	1.500020e+01
	$\|\nabla \hat{J}\|_2$	1.898753e-07	1.898753e-07	1.898753e-07	1.898753e-07
	Time	8.096211	8.099363	8.320182	8.541796

Tab. 1. Numerical results for Example 1 ($\beta = 1$).

β	Item	Method with $m = 5$		Method with $m = 25$	
		NLBFGS	LBFGS	NLBFGS	LBFGS
10^{-2}	Iteration	13	51	11	16
	$\hat{J}$	2.049153e+00	1.485024e+01	2.049153e+00	2.049153e+00
	$\|\nabla\hat{J}\|_2$	7.115463e-07	6.008818e-03**(LF)**	5.831008e-07	1.401287e-07
	Time	29.137373	114.382628	22.757801	30.709843
10^{-3}	Iteration	38	51	19	51
	$\hat{J}$	6.516870e-01	1.489867e+01	6.516872e-01	8.888296e+00
	$\|\nabla\hat{J}\|_2$	9.991684e-07	6.351331e-03**(LF)**	6.873622e-07	2.122447e-03**(LF)**
	Time	109.809225	292.347146	65.452776	219.645003
10^{-4}	Iteration	51	51	47	51
	$\hat{J}$	2.090319e-01	1.804914e+01	2.065172e-01	1.722022e+01
	$\|\nabla\hat{J}\|_2$	2.727256e-05**(LF)**	1.117405e-02**(LF)**	9.330302e-07	3.948579e-03**(LF)**
	Time	195.196179	439.364967	149.270008	311.607472
10^{-5}	Iteration	51	51	51	51
	$\hat{J}$	9.430789e-02	1.820208e+01	8.287246e-02	1.785269e+01
	$\|\nabla\hat{J}\|_2$	6.560806e-05**(LF)**	8.024630e-03**(LF)**	9.423505e-05**(LF)**	1.465229e-03**(LF)**
	Time	243.865859	582.593476	215.915091	462.616446
10^{-6}	Iteration	51	51	51	51
	$\hat{J}$	8.858351e-02	5.934376e+00	6.828165e-02	1.689529e+01
	$\|\nabla\hat{J}\|_2$	1.016241e-04**(LF)**	3.468926e-03**(LF)**	1.184144e-04**(LF)**	3.831390e-03**(LF)**
	Time	284.845243	746.318423	253.121125	636.513783

Tab. 2. Numerical results for Example 1 with the mesh 64×4096.

$\|y^*_{NLBFGS} - y^*_{RBF}\|_2$	$\|y^*_{NLBFGS} - y^*_{SCP}\|_2$	$\|y^*_{NLBFGS} - y^*_{SLP}\|_2$	$\|y^*_{NLBFGS} - y^*_{TS}\|_2$
1.61e-02	2.8895e-04	2.7531e-04	3.3410e-04

Tab. 3. Residuals values of $y^*(2,t)$ with $n = 32$ and $N = 1024$.

$\|u^*_{NLBFGS} - u^*_{RBF}\|_2$	$\|u^*_{NLBFGS} - u^*_{SCP}\|_2$	$\|u^*_{NLBFGS} - u^*_{SLP}\|_2$	$\|u^*_{NLBFGS} - u^*_{TS}\|_2$
2.3e-03	1.3e-03	1.2e-03	3.6e-03

Tab. 4. Residuals values of $u^*(2,t)$ with $n = 32$ and $N = 1024$.

β	*Mesh*	8×64	16×256	32×1024	64×4096	128×16384
1	$\|R(x,t)\|_2$	3.2392e-16	2.4827e-16	1.7829e-16	1.2884e-16	9.2308e-17
0.001	$\|R(x,t)\|_2$	3.7200e-17	3.8083e-17	2.8253e-17	2.1376e-17	1.7129e-17

Tab. 5. $\|R(x,t)\|_2$ for Example 1 ($m = 25$).

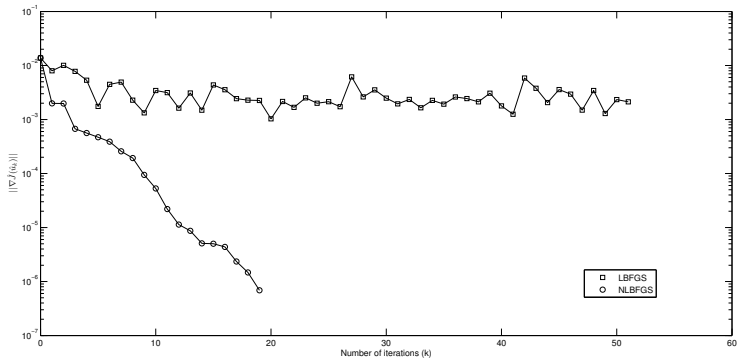

Fig. 1. $\|\nabla \hat{J}(\hat{u}_k)\|$ for Example 1 with the mesh 64×4096.

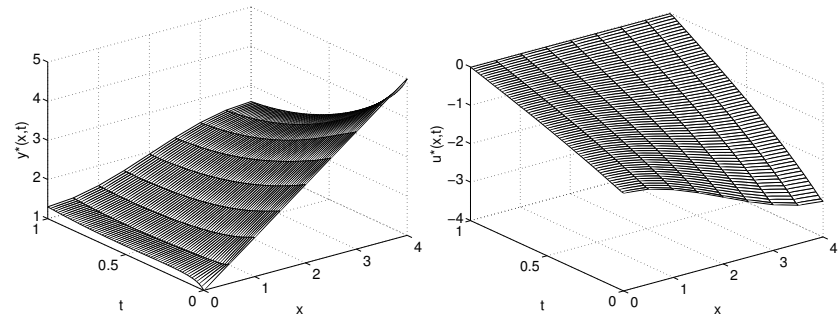

Fig. 2. Numerical solutions for the optimal state (left) and optimal control (right) via the NLBFGS method for Example 1 with $n = 8$ and $N = 64$ ($\beta = 1$).

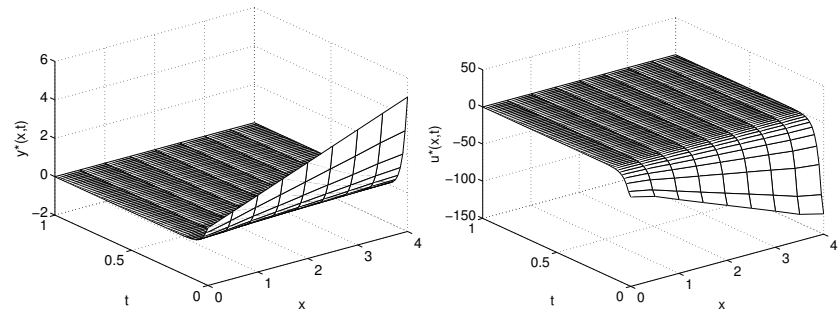

Fig. 3. Numerical solutions for the optimal state (left) and optimal control (right) via the NLBFGS method for Example 1 with $n = 8$ and $N = 64$ ($\beta = 0.001$).

Example 2. Consider the control problem (8) – (11) with

$$t_f = 1, \quad \beta = 0.05, \quad \nu = 0.01 \text{ and}$$

$$z(x,t) = f(x) = \begin{cases} 1 & x \in (0, \frac{1}{2}], \\ 0 & \text{else.} \end{cases}$$

In Table 6, the numerical results obtained by NLBFGS method are compared with LBFGS method. These results are obtained with stopping criterion $\epsilon_1 = 10^{-5}$. The resulting graphs of optimal state $y^*(x,t)$ and optimal control $u^*(x,t)$ are shown in Figure 4.

Mesh	Item	Method with $m = 5$		Method with $m = 25$	
		NLBFGS	LBFGS	NLBFGS	LBFGS
8×64	Iteration	9	51	8	12
	$\hat{J}$	3.648315e-02	3.651749e-02	3.648311e-02	3.648323e-02
	$\|\nabla \hat{J}\|_2$	6.194044e-06	2.393841e-04**(LF)**	4.884531e-06	9.112672e-06
	Time	0.932989	3.008503	0.864858	0.891160
16×256	Iteration	9	38	8	12
	$\hat{J}$	4.797972e-02	4.797965e-02	4.798137e-02	4.797974e-02
	$\|\nabla \hat{J}\|_2$	2.667570e-06	2.635461e-06	9.950448e-06	2.959558e-06
	Time	3.944498	8.837802	3.389260	3.955541
32×1024	Iteration	7	51	8	12
	$\hat{J}$	5.598883e-02	9.289697e-02	5.598737e-02	5.598938e-02
	$\|\nabla \hat{J}\|_2$	7.756053e-06	9.671602e-04**(LF)**	4.788751e-06	6.262238e-06
	Time	11.558364	54.334143	12.156612	13.622016
64×4096	Iteration	6	43	6	12
	$\hat{J}$	6.087640e-02	6.086642e-02	6.087640e-02	6.085215e-02
	$\|\nabla \hat{J}\|_2$	6.842519e-06	4.796432e-06	6.842519e-06	1.473778e-06
	Time	41.397493	174.344337	39.994682	55.644107

Tab. 6. Numerical results for Example 2.

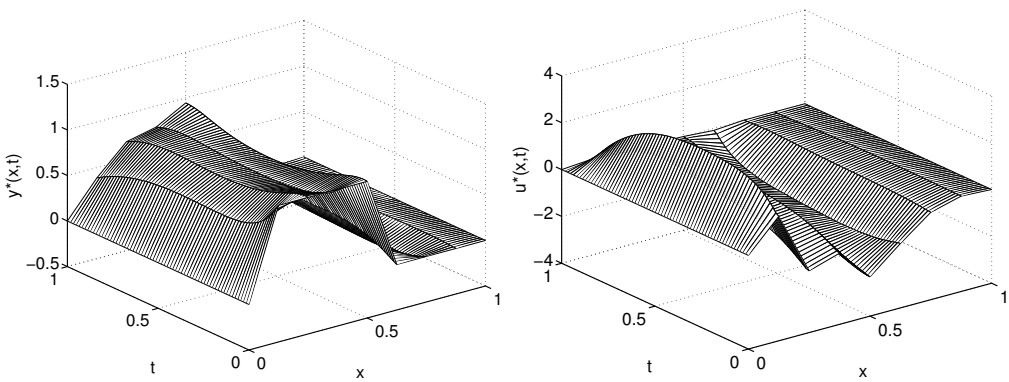

Fig. 4. Numerical solutions for the optimal state (left) and optimal control (right) via the NLBFGS method for Example 2 with $n = 8$ and $N = 64$.

Example 3. Consider the control problem (8) – (11) with

$$t_f = 1, \quad \beta = 0.001, \quad \nu = 0.05, \quad f(x) = sin(4\pi x), \quad z(x,t) = 0.$$

In Table 7, the numerical results obtained by NLBFGS method are compared with LBFGS method. These results are obtained with $m = 5, 25$ and stopping criterion $\epsilon_1 = 10^{-5}$. Figure 5 shows the graphs of optimal state $y^*(x,t)$ and optimal control $u^*(x,t)$.

Mesh	Item	Method with $m = 5$		Method with $m = 25$	
		NLBFGS	LBFGS	NLBFGS	LBFGS
8×64	Iteration	5	51	5	51
	$\hat{J}$	4.133468e-03	7.829341e-03	4.133468e-03	6.984157e-03
	$\|\nabla \hat{J}\|_2$	6.328187e-06	1.004423e-03**(LF)**	6.328187e-06	3.010519e-04**(LF)**
	Time	0.511251	11.523740	0.510006	6.370753
16×256	Iteration	5	51	5	51
	$\hat{J}$	5.542397e-03	1.043922e-02	5.542397e-03	1.122081e-02
	$\|\nabla \hat{J}\|_2$	9.253305e-06	4.995947e-04**(LF)**	9.253305e-06	2.092796e-04 **(LF)**
	Time	2.152638	46.344747	2.451300	28.532890
32×1024	Iteration	4	51	4	51
	$\hat{J}$	6.154599e-03	8.958653e-03	6.154599e-03	8.314431e-03
	$\|\nabla \hat{J}\|_2$	8.199020e-06	1.457551e-04**(LF)**	8.199020e-06	3.935704e-05**(LF)**
	Time	6.545823	184.164083	6.187243	96.019940
64×4096	Iteration	2	6	2	6
	$\hat{J}$	7.276688e-03	8.439227e-03	7.276688e-03	8.439227e-03
	$\|\nabla \hat{J}\|_2$	7.266556e-06	9.238552e-06	7.266556e-06	9.238552e-06
	Time	15.227677	44.394250	14.062550	41.377309

Tab. 7. Numerical results for Example 3.

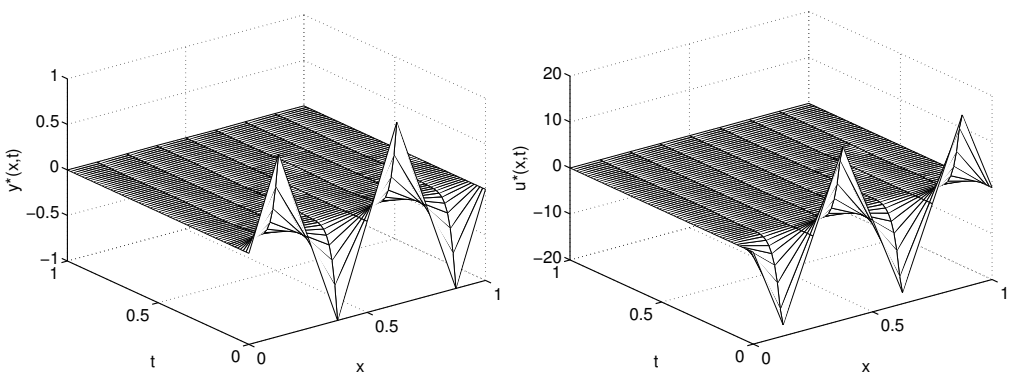

Fig. 5. Numerical solutions for the optimal state (left) and optimal control (right) via the NLBFGS method for Example 3 with $n = 8$ and $N = 64$.

Example 4. Consider the control problem (8) – (11) with

$$t_f = 1, \quad \beta = 0.01, \quad \nu = 0.01, \quad f(x) = sin(4\pi x), \quad z(x,t) = 0.$$

In Table 8, the numerical results obtained by NLBFGS method are compared with LBFGS method. These results are obtained with stopping criterion $\epsilon_1 = 10^{-5}$. Figure 6 shows the graphs of optimal state $y^*(x,t)$ and optimal control $u^*(x,t)$.

Mesh	Item	Method with $m = 5$		Method with $m = 25$	
		NLBFGS	LBFGS	NLBFGS	LBFGS
8×64	Iteration	5	51	5	7
	$\hat{J}$	1.385427e-02	4.106236e-02	1.385322e-02	1.385427e-02
	$\|\nabla\hat{J}\|_2$	9.216487e-06	2.613823e-03**(LF)**	9.216487e-06	4.567480e-06
	Time	0.431840	5.037703	0.556830	0.575118
16×256	Iteration	7	51	7	8
	$\hat{J}$	1.866906e-02	2.556663e-02	1.866742e-02	1.866876e-02
	$\|\nabla\hat{J}\|_2$	4.577507e-06	5.207466e-04**(LF)**	4.616202e-06	2.046263e-06
	Time	2.254163	23.797662	2.222336	2.595283
32×1024	Iteration	6	10	6	8
	$\hat{J}$	2.025195e-02	2.024898e-02	2.023379e-02	2.025195e-02
	$\|\nabla\hat{J}\|_2$	6.059584e-06	7.632950e-06	6.059584e-06	3.085386e-06
	Time	7.431759	10.538716	7.424505	8.395777
64×4096	Iteration	5	6	5	6
	$\hat{J}$	2.070344e-02	2.079572e-02	2.070344e-02	2.079572e-02
	$\|\nabla\hat{J}\|_2$	4.909728e-06	5.708431e-06	4.909728e-06	5.708431e-06
	Time	27.106260	30.321599	25.292322	28.031794

Tab. 8. Numerical results for Example 4.

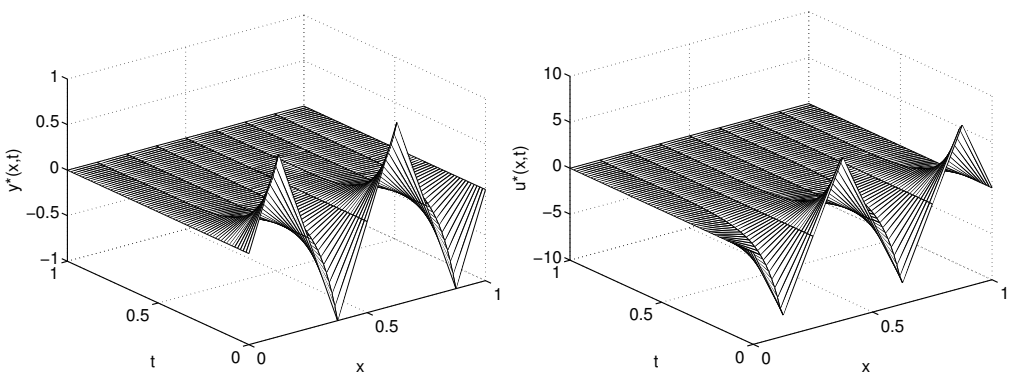

Fig. 6. Numerical solutions for the optimal state (left) and optimal control (right) via the NLBFGS method for Example 4 with $n = 8$ and $N = 64$.

Example 5. Consider the problem (8) – (11) with the following cost functional

$$J(y,u)=\frac{1}{2}\int_0^{t_f}\int_0^1((y(x,t)-z(x,t))^2+\beta u^2(x,t))\,\mathrm{d}x\mathrm{d}t+\frac{1}{2}\int_0^1(y(x,t_f)-f(x))^2\mathrm{d}x \quad (29)$$

and with

$$t_f=1,\quad \beta=0.01,\quad \nu=0.01,\quad z(x,t)=f(x)=sin(\pi\tfrac{tan(1.3(2x-1))}{tan(1.3)}).$$

Here, the function $f(x)$ is the same as in (9). This example is an optimal flow control problem and corresponds to a target optimal control problem. ([23], section 7.2). Two targets $z(x,t)$ and $f(x)$ are generally determined based on physical arguments such as laminar flow and solutions of minimum drag. In Table 9, the numerical results obtained by NLBFGS method are compared with LBFGS method. These results are obtained with $m=5,25$ and stopping criterion $\epsilon_1=10^{-5}$. Figure 7 shows the graphs of optimal state $y^*(x,t)$ and optimal control $u^*(x,t)$. Also, Table 10 shows the values of the residual error for the discretized form of (8), $\|R(x,t)\|_2$, for various meshes.

Mesh	Item	Method with $m=5$		Method with $m=25$	
		NLBFGS	LBFGS	NLBFGS	LBFGS
8×64	Iteration	20	51	14	51
	$\hat{J}$	4.792175e-03	1.014637e-01	4.792634e-03	7.789185e-02
	$\|\nabla\hat{J}\|_2$	5.763069e-06	8.246042e-03**(LF)**	6.644234e-06	7.055744e-03**(LF)**
	Time	2.047803	9.087033	1.438303	5.966637
16×256	Iteration	14	51	12	51
	$\hat{J}$	6.685872e-03	7.802178e-02	6.684669e-03	7.349081e-02
	$\|\nabla\hat{J}\|_2$	7.965832e-06	2.032267e-03**(LF)**	7.793169e-06	1.628931e-03**(LF)**
	Time	5.989127	35.234971	5.172790	25.602314
32×1024	Iteration	11	51	11	51
	$\hat{J}$	6.723092e-03	6.147657e-02	6.710989e-03	6.748334e-02
	$\|\nabla\hat{J}\|_2$	9.730515e-06	4.330754e-04**(LF)**	8.728665e-06	5.809067e-04**(LF)**
	Time	18.697483	127.543102	18.519027	94.483924
64×4096	Iteration	9	51	10	51
	$\hat{J}$	6.967697e-03	1.264703e-01	6.862438e-03	1.034227e-01
	$\|\nabla\hat{J}\|_2$	8.253798e-06	3.019590e-04**(LF)**	8.003223e-06	2.605691e-04**(LF)**
	Time	66.359313	507.085028	71.685829	378.163260

Tab. 9. Numerical results for Example 5.

m	*Mesh*	8×64	16×256	32×1024	64×4096
5	$\|R(x,t)\|_2$	5.1871e-08	4.6291e-09	9.3675e-11	1.4799e-12
25	$\|R(x,t)\|_2$	5.1954e-08	4.6118e-09	8.7941e-11	1.4199e-12

Tab. 10. $\|R(x,t)\|_2$ for Example 5

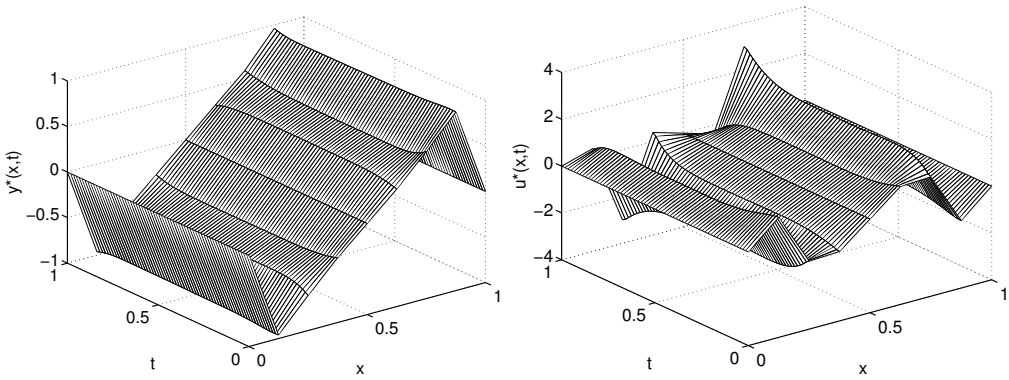

Fig. 7. Numerical solutions for the optimal state (left) and optimal control (right) via the NLBFGS method for Example 5 with $n = 8$ and $N = 64$ ($m = 25$).

As can be seen in Tables 1,2 and 6-9, the proposed algorithm, NLBFGS algorithm, with $m = 25$ is the most robust, i. e., it can solve the most problems. On the whole, the figures and tables show that the proposed algorithm has efficient performances and promising behavior for solving parabolic optimal control problems.

6. CONCLUSION

In this paper, a nonlinear optimization scheme has been used to find the optimal control to parabolic distributed parameter systems. It is well-known that the nonmonotone globalization technique is a useful tool in difficult nonlinear problems, because of the fact that it may help escaping from steep sided valleys and may improve both the possibility of finding the global optimum and the rate of convergence. So, due to the efficiency and low memory requirements of LBFGS method, in this paper, an efficient nonmonotone version of the LBFGS method, NLBFGS, has been applied to solve the unconstrained optimization problem arisen from discretization of the optimal control problems. To validate the algorithm NLBFGS from a computational point of view, we compared it with different approximate methods. This strategy is especially suited for problems which are ill-posed. Preliminary numerical results indicated that the proposed algorithm has efficient performances and promising behavior for solving parabolic optimal control problems.

(Received September 10, 2017)

REFERENCES

[1] H. Albrecher, W. J. Runggaldier, and W. Schachermayer: Advanced Financial Modelling. Radon series on computational and applied mathematics, Walter de Gruyter, 2009. DOI:10.1515/9783110213140

[2] K. Amini, M. Ahookhosh, and H. Nosratipour: An inexact line search approach using modified nonmonotone strategy for unconstrained optimization. Numer. Algor. *66* (2014), 49–78. DOI:10.1007/s11075-013-9723-x

[3] S. Aniţa, V. Arnăutu, and V. Capasso: An Introduction to Optimal Control Problems in Life Sciences and Economics: From Mathematical Models to Numerical Simulation with MATLAB. Birkhäuser, Boston 2011.

[4] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty: Nonlinear Programming: Theory and Algorithms. Wiley, New York 2006. DOI:10.1002/0471787779

[5] A. Borzi and V. Schulz: Computational Optimization of Systems Governed by Partial Differential Equations. SIAM, 2012. DOI:10.1137/1.9781611972054

[6] F. Brezzi and M. Fortin: Mixed and Hybrid Finite Element Methods. Springer, New York 2012. DOI:10.1007/978-1-4612-3172-1

[7] S. Cantrell, C. Cosner, and S. Ruan: Spatial Ecology. CRC Mathematical and Computational Biology, CRC Press 2009. DOI:10.1201/9781420059861

[8] R. Y. Chang and S. Y. Yang: Solution of two point boundary value problems by generalized orthogonal polynomials and application to optimal control of lumped and distributed parameter systems. International Journal of Control *43* (1986), 1785–1802. DOI:10.1080/00207178608933572

[9] P. Christofides, A. Armaou, Y. Lou, and A. Varshney: Control and Optimization of Multiscale Process Systems, Control Engineering. Birkhäuser, Boston 2008. DOI:10.1007/978-0-8176-4793-3

[10] E. De Klerk, C. Roos, and T. Terlaky: Nonlinear Optimization. University Of Waterloo, Waterloo 2005.

[11] I. Griva, S. G. Nash, and A. Sofer: Linear and Nonlinear Optimization. SIAM, Philadelphia 2009. DOI:10.1137/1.9780898717730

[12] J. Haslinger and P. Neittaanmäki: Finite Element Approximation for Optimal Shape, Material and Topology Design. Wiley, 1996.

[13] M. Heinkenschloss: Numerical Solution of Implicitly Constrained Optimization Problems. CAAM Technical Report TR08-05, Rice University (2008).

[14] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich: Optimization with PDE Constraints. Springer, Netherlands 2008.

[15] I. R. Horng and J. H. Chou: Application of shifted Chebyshev series to the optimal control of linear distributed-parameter systems. Int. J. Control *42* (1985), 233–241. DOI:10.1080/00207178508933359

[16] W. W. Hu: Approximation and Control of the Boussinesq Equations with Application to Control of Energy Efficient Building Systems. Ph.D. Thesis, Department of Mathematics, Virginia Tech. 2012.

[17] Y. Ji, Y. Li, K. Zhang, and X. Zhan: A new nonmonotone trust-region method of conic model for solving unconstrained optimization. J. Comput. Appl. Math. *233* (2010), 1746–1754. DOI:10.1016/j.cam.2009.09.011

[18] K. Kunisch and S. Volkwein: Control of the burgers equation by a reduced-order approach using proper orthogonal decomposition. J. Optim. Theory Appl. *102* (1999), 345–371. DOI:10.1023/a:1021732508059

[19] J. L. Lions: Optimal Control of Systems Governed by Partial Differential Equations. Springer–Verlag, 1971. DOI:10.1007/978-3-642-65024-6

[20] D. Liu and J. Nocedal: On the limited memory BFGS method for large-scale optimization. Math. Program. *45* (1989), 503–528. DOI:10.1007/bf01589116

[21] P. Merino: Finite element error estimates for an optimal control problem governed by the Burgers equation. Comput. Optim. Appl. *63* (2016), 793–824. DOI:10.1007/s10589-015-9790-0

[22] C. Meyer, P. Philip, and F. Tröltzsch: Optimal control of a semilinear PDE with nonlocal radiation interface conditions. SIAM J. Control Optim. *45* (2006), 699–721. DOI:10.1137/040617753

[23] B. R. Noack, M. Morzynski, and G. Tadmor: Reduced-Order Modelling for Flow Control. Springer, Vienna 2011. DOI:10.1007/978-3-7091-0758-4

[24] J. Nocedal: Updating quasi-Newton matrices with limited storage. Math. Comput. *35* (1980) 773–782. DOI:10.1090/s0025-5718-1980-0572855-7

[25] J. Nocedal and S. Wright: Numerical Optimization. Springer, New York 2006. DOI:10.1007/b98874

[26] H. Nosratipour, A. H. Borzabadi, and O. S. Fard: Optimal control of viscous Burgers equation via an adaptive nonmonotone Barzilai-Borwein gradient method. Int. J. Comput. Math. *95* (2018) 1858–1873. DOI:10.1080/00207160.2017.1343472

[27] H. Nosratipour, A. H. Borzabadi, and O. S. Fard: On the nonmonotonicity degree of nonmonotone line searches. Calcolo *54* (2017) 1217–1242. DOI:10.1007/s10092-017-0226-3

[28] H. Nosratipour, O. S. Fard, and A. H. Borzabadi: An adaptive nonmonotone global Barzilai–Borwein gradient method for unconstrained optimization. Optimization *66* (2017) 641–655. DOI:10.1080/02331934.2017.1287702

[29] J. A. Rad, S. Kazem, and K. Parand: Optimal control of a parabolic distributed parameter system via radial basis functions. Commun. Nonlinear Sci. Numer. Simul. *19* (2014), 2559–2567. DOI:10.1016/j.cnsns.2013.01.007

[30] M. Razzaghi and A. Arabshahi: Optimal control of linear distributed-parameter systems via polynomial series. Int. J. Systems Sci. *20* (1989), 1141–1148. DOI:10.1080/00207728908910200

[31] Z. Sabeh, M. Shamsi, and M. Dehghan: Distributed optimal control of the viscous Burgers equation via a Legendre pseudo-spectral approach. Math. Methods Appl. Sci. *39* (2016), 3350–3360. DOI:10.1002/mma.3779

[32] G. Strang and G. Fix: An Analysis of the Finite Element Method. Wellesley-Cambridge Press, 2008.

[33] F. Tröltzsch: Optimal Control of Partial Differential Equations: Theory, Methods, and Applications. Graduate studies in mathematics, American Mathematical Society, 2010. DOI:10.1090/gsm/112

[34] F. Tröltzsch and S. Volkwein: The SQP method for control constrained optimal control of the Burgers equation. ESAIM: COCV *6* (2001), 649–674. DOI:10.1051/cocv:2001127

[35] F. S. Wang and J. B. Jian: A new nonmonotone linesearch SQP algorithm for unconstrained minimax problem. Numer. Funct. Anal. Optim. *35* (2014), 487–508. DOI:10.1080/01630563.2013.873454

[36] F. Yılmaz and B. Karasözen: Solving distributed optimal control problems for the unsteady Burgers equation in COMSOL multiphysics. J. Comput. Appl. Math. *235* (2011), 4839–4850. DOI:10.1016/j.cam.2011.01.002

[37] H. Zhang and W. W. Hager: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. *14* (2004), 1043–1056. DOI:10.1137/s1052623403428208

Omid Solaymani Fard, Department of Applied Mathematics, School of Mathematics and Computer Science, Damghan University, Damghan. Iran.
e-mail: osfard@du.ac.ir

Farhad Sarani, Department of Applied Mathematics, School of Mathematics and Computer Science, Damghan University, Damghan. Iran.
e-mail: saranifarhad@gmail.com

Akbar Hashemi Borzabadi, Department of Applied Mathematics, School of Mathematics and Computer Science, Damghan University, Damghan. Iran.
e-mail: borzabadi@du.ac.ir

Hadi Nosratipour, Department of Applied Mathematics, School of Mathematics and Computer Science, Damghan University, Damghan. Iran.
e-mail: hadi.nosratipour@gmail.com