Computational Optimization and Applications, 1, (1992), 113–135

Numerical Solution of a Nonlinear Parabolic Control Problem by a Reduced SQP Method

F.-S. KUPFER AND E.W. SACHS
Universität Trier, FB IV-Mathematik, Postfach 3825, W-5500 Trier, Germany

Received January 28, 1992, Revised May 26, 1992.

Abstract. We consider a control problem for a nonlinear diffusion equation with boundary input that occurs when heating ceramic products in a kiln. We interpret this control problem as a constrained optimization problem, and we develop a reduced SQP method that presents for this problem a new and efficient approach of its numerical solution. As opposed to Newton's method for the unconstrained problem, where at each iteration the state must be computed from a set of nonlinear equations, in the proposed algorithm only the linearized state equations need to be solved. Furthermore, by use of a secant update formula, the calculation of exact second derivatives is avoided. In this way the algorithm achieves a substantial decrease in the total cost compared to the implementation of Newton's method in [2]. Our method is practicable with regard to storage requirements, and by choosing an appropriate representation for the null space of the Jacobian of the constraints we are able to exploit the sparsity pattern of the Jacobian in the course of the iteration. We conclude with a presentation of numerical examples that demonstrate the fast two-step superlinear convergence behavior of the method.

Keywords: optimal boundary control, nonlinear heat equation, reduced successive quadratic programming (SQP), constrained optimization, BFGS-update, null space parametrization, two-step superlinear convergence

1. Introduction

In this paper we consider a nonlinear diffusion problem with boundary input that occurs in the heating of kilns in the ceramic industry. In a recent publication ([2]) by Burger and Pogu, the following problem is formulated.

Let $y(x, t)$ denote the temperature inside the probe to be heated in a kiln where the time t ranges from 0 to T and the spatial domain Ω is [0,1]. The point $x = 1$ is located in the inside of the probe and $x = 0$ is at the boundary of the probe. Then the boundary value problem is given by

$$\begin{aligned} C(y(x,t))\frac{\partial y}{\partial t}(x,t) - \nabla(\lambda(y(x,t))\nabla y(x,t)) &= f(x,t) \text{ on } \Omega \times [0,T] \\ \lambda(y(x,t))\nabla y(x,t) &= p(x,t) \text{ on } \partial\Omega \times [0,T] \quad (1) \\ y(x,0) &= y_0(x) \text{ on } \Omega \end{aligned}$$

Here the functions $C, \lambda : \mathbb{R} \to \mathbb{R}$ denote the specific heat capacity and the heat conduction, respectively, which both depend on the temperature y. y_0 is

the initial temperature distribution, f is the source term, and p is the boundary input. Since $x = 1$ is located inside the probe, there is no heat flux at 1, whereas for $x = 0$ we impose a control law that enters linearly through the temperature inside the kiln:

$$\begin{aligned} p(0, t) &= g[y(0, t) - u(t)] \\ p(1, t) &= 0 \end{aligned} \tag{2}$$

g is a real number that we usually normalize to be 1.

The goal is to control the heating process in such a way that the temperature inside the probe follows a certain desired firing curve $\hat{y}(t)$

$$\text{Minimize} \int_0^T [(y(1, t) - \hat{y}(t))^2 + \alpha u^2(t)]dt \tag{3}$$

for a given constant $\alpha \geq 0$.

This optimal control problem can be formulated as an infinite dimensional optimization problem choosing the control and state space as follows:

$$u \in L^2(0, T),\ y \in L^2(0, T; H^1(\Omega))$$

Under proper assumptions on λ and C it is shown in [2] that there exists a solution to (1)–(3) for $\alpha > 0$.

In [2], the control u is considered to be the independent variable, and the state is substituted through (1) and (2) as a variable depending on u. For a numerical solution, the authors in [2] introduce a discretization of the differential equation and the objective function. Hence, the resulting optimization problem can be written as

$$\text{Minimize } F(y(u), u), \quad u \in \mathbb{R}^M$$

with $F: \mathbb{R}^{K+M} \to \mathbb{R}$, where the variable y depends on u through the solution of a discrete scheme for (1) and (2). In [2], the method of steepest descent, Newton's method, and the conjugate gradient method are used as optimization routines. Newton's method is reported to be very fast, but the calculation of the Hessian is rather expensive, and at each iteration a system of nonlinear equations must be solved. Similarly, the solution of the state equation is a drawback for the use of differential dynamic programming for discrete control systems [12]. This led us to consider a different approach to the numerical solution of the problem.

We consider y and u as independent variables and interpret the boundary value problem as a nonlinear equality constraint in a constrained optimization problem:

$$\text{Minimize } F(y, u) \text{ subject to } h(y, u) = 0, \quad y \in \mathbb{R}^K, \quad u \in \mathbb{R}^M \tag{4}$$

with $h : \mathbb{R}^{K+M} \to \mathbb{R}^K$. The discussion of the previous paragraph motivates to look at methods that produce infeasible iterates such as successive quadratic programming (SQP) methods that are among the most successful algorithms for solving constrained optimization problems. There are several issues that should be considered when solving a finite dimensional optimization problem (4), that comes from a discretization of a parabolic control problem. Among those are the following.

- In problems of a multidimensional space variable, the state variable y is much larger than the control variable u.
- The nonlinear equality constraint h and its linearization exhibit a large amount of structure.

If one uses a secant SQP method, a matrix, usually the Hessian of the Lagrangian or augmented Lagrangian function, needs to be updated. This matrix could be of rather large dimension as indicated by the first item. Reduced secant SQP methods offer a different viewpoint. Here, the matrix to be updated is only of the dimension of the null space of the Jacobian of the constraints, which is usually relatively small. The second item indicates that the structure in h and the resulting sparsity in h' should be used in the course of the iteration, and this can be achieved if reduced SQP is applied in an adequate way.

In this paper we solve (1)–(3) by a reduced secant SQP method, which presents a new approach to an efficient solution of this nonlinear boundary control problem. In Section 2 we motivate reduced methods for general constrained optimization problems and we state a convergence result. Furthermore, for problems of the type (4), we exploit the splitting of the variables into $y \in \mathbb{R}^K$ and $u \in \mathbb{R}^M$ in order to develop a reduced BFGS algorithm that is appropriate for optimal control problems. In Section 3, we formulate this algorithm for the minimization problem that results from a discretization of the parabolic control problem. A central feature of the algorithm is that the computation of second-order information is avoided. Furthermore, only the linearized discrete state equation must be solved, as opposed to the solution of a system of nonlinear equations per iteration of Newton's method. In addition, the sparsity of the Jacobian is used efficiently in the course of the algorithm, and we are able to retain a fast convergence rate and practicable storage requirements. In Section 4, we discuss implementation details and present numerical results.

2. Reduced SQP methods

In this section we motivate reduced SQP methods, and we develop a particular reduced BFGS algorithm for minimization problems of the type (4). First, we consider the following general constrained optimization problem:

$$\text{Minimize } F(z) \text{ subject to } h(z) = 0, \quad z \in \mathbb{R}^s \tag{5}$$

with $F: \mathbb{R}^s \to \mathbb{R}$ and $h: \mathbb{R}^s \to \mathbb{R}^q$.

We assume that $h'(z)$ is surjective for all z in some neighborhood of the solution and let $T(z) \in \mathbb{R}^{s \times (s-q)}$ be any basis for the null space of the Jacobian of the constraints:

$$\mathcal{N}[h'(z)] := \{p \in \mathbb{R}^s : h'(z)p = 0\} = \{T(z)w : w \in \mathbb{R}^{s-q}\} =: \mathcal{R}[T(z)]$$

At each iteration of an SQP method, a quadratic approximation of the Lagrangian is minimized under linearized constraints:

$$\text{Minimize } \nabla F(z)^T d + \frac{1}{2} d^T H d \text{ subject to } h'(z)d + h(z) = 0 \tag{6}$$

The new iterate is $z_+ = z + d$. Here $H \in \mathbb{R}^{s \times s}$ is an approximation to the Hessian $L_{zz}(z, l)$ of the Lagrangian function $L(z, l) = F(z) - l^T h(z)$.

With any right-inverse $R(z) \in \mathbb{R}^{s \times q}$ of $h'(z)$, i.e., $h'(z)R(z) = I \in \mathbb{R}^{q \times q}$, the feasible points in (6) can be represented as

$$d = T(z)w - R(z)h(z), \quad w \in \mathbb{R}^{s-q}$$

and we obtain the following closed-form expression for the SQP step (see e.g., [8]):

$$d_{SQP} = -T(z)[T(z)^T H T(z)]^{-1} T(z)^T [\nabla F(z) - HR(z)h(z)] - R(z)h(z)$$

In the case of linear constraints, all iterates z_k generated from an SQP method are feasible for $k > 1$ so that the expression $HR(z)h(z)$ vanishes in d_{SQP}. Hence, it is reasonable to replace $T(z)^T H T(z)$ by a matrix that approximates the reduced Hessian $T(z)^T L_{zz}(z, l) T(z)$ directly. If the idea to approximate only the reduced Hessian while discarding the $HR(z)h(z)$ term is applied to the case of general constraints, one is led to the reduced SQP step:

$$p = -T(z)B^{-1}T(z)^T \nabla F(z) - R(z)h(z) \tag{7}$$

Here $B \in \mathbb{R}^{(s-q)\times(s-q)}$ is interpreted as an approximation to the reduced Hessian and usually a secant update formula is used to compute the new matrix B_+. We formulate an iteration of a reduced secant method with a BFGS update for general constrained optimization problems of the type (5).

2.1. *Algorithm 1 (reduced BFGS method)*

Given $z \in \mathbb{R}^s$ and $B \in \mathbb{R}^{(s-q)\times(s-q)}$, B nonsingular.

1. Solve $Bw = -T(z)^T \nabla F(z)$.
2. Set $z_+ = z + T(z)w - R(z)h(z)$.
3. Compute $v = T(z + T(z)w)^T \nabla F(z + T(z)w) - T(z)^T \nabla F(z)$.

4. Set

$$B_+ = B + \frac{vv^T}{v^T w} - \frac{(Bw)(Bw)^T}{w^T Bw}$$

if it is well defined, else set $B_+ = B$.

This method has the advantage that only a matrix of the dimension $s-q$ needs to be stored and updated as opposed to a matrix in $\mathbb{R}^{s\times s}$ in an SQP method. Furthermore, since in general only the reduced Hessian is positive definite at the solution, algorithm 1 is in line with the second-order sufficiency condition, and, locally, the reduced BFGS update in steps 3 and 4 maintains the positive definiteness of the secant approximation B.

For a general null space basis and right-inverse, the step (7) is introduced in [8], and modifications of the standard step are discussed in this general setting in [8], [9], and [10]. Frequently, only a particular choice for R and T is considered, namely $R(z) = h'(z)^T[h'(z)h'(z)^T]^{-1}$ is the Moore-Penrose pseudo-inverse of $h'(z)$, and $T(z)$ is an orthonormal basis of $\mathcal{N}[h'(Z)]$. In this case, the restoration step $-R(z)h(z)$ can be regarded as a minimum norm Newton step on the equation $h(z) = 0$. Reduced SQP methods within this setting, which is sometimes called the orthogonal framework and is generally implemented using a QR factorization, are studied in [4], [5], [6], [11], [17], and [19]. In our application, however, the control space is the proper choice of parameter space; the orthogonal framework is not particularly suitable in this case, because it does not automatically preserve the sparsity structure. In addition, the control space parametrization allows a simple interpretation in infinite dimensions, which is important for fine discretizations, in contrast to the QR factorization. At the end of this section we will demonstrate how to take advantage of the splitting into control and state variables in order to define very naturally the matrices R and T. This leads to a fast and efficient algorithm for the solution of the parabolic boundary control problem presented in the introduction.

Next we state a convergence result for general reduced SQP methods.

ASSUMPTION 1. *F and h are twice differentiable on a ball D, which contains the solution z^* of (5). Furthermore, $F''(\cdot)$ and $h'(\cdot)$ are Lipschitz-continuous on D.*

If the choice of null space basis or right-inverse changes too much from one iterate to the next, fast convergence for reduced methods can be impeded. Therefore, some smoothness for R and T must be required. We summarize the definition and the properties of the null space representation and the right-inverse in the following.

ASSUMPTION 2. *For each $z \in D$, let $T(z) \in \mathbb{R}^{s\times(s-q)}$ be a matrix of full rank with*

$$\mathcal{N}[h'(z)] = \mathcal{R}[T(z)]$$

Furthermore there exists a matrix $R(z) \in \mathbb{R}^{s \times q}$ *with* $h'(z)R(z) = I_q$.

In addition, $T(\cdot)$, $R(\cdot)$ *are differentiable and* $T'(\cdot)$, $R'(\cdot)$ *are Lipschitz-continuous on D.*

We note that Assumption 2 implies that $h'(z)$ is surjective for $z \in D$. To prove a fast rate of convergence, a second-order sufficiency condition is required. This can be formulated as follows: Let $L(z, l)$ denote the Lagrangian, and let $l^* \in \mathbb{R}^q$ be the Lagrange multiplier corresponding to z^*.

ASSUMPTION 3. *There exists some* $m > 0$ *such that*

$$\zeta^T L_{zz}(z^*, l^*)\zeta \geq m\|\zeta\|^2 \quad \textit{for all} \quad \zeta \in \mathcal{N}[h'(z^*)]$$

Since only reduced second-order information is approximated, one cannot in general expect a reduced method (7) to be q-superlinearly convergent (see the examples in [3] and [18]). However, under the stated assumptions, a two-step q-superlinear convergence rate can be achieved:

THEOREM 1. *Let* $z^* \in \mathbb{R}^s$ *be a solution of (5), and let Assumptions 1–3 be satisfied. Suppose that* $\{z_k\}_{k=0}^{\infty}$ *is generated by algorithm 1 with* B_0 *symmetric and positive definite. Then there exist positive constants* ϵ *and* δ *such that if*

$$\|z_0 - z^*\| < \epsilon \qquad \textit{and} \qquad \|B_0 - T(z^*)^T L_{zz}(z^*, l^*)T(z^*)\| < \delta$$

the sequence $\{z_k\}$ *is well defined and converges to* z^* *at a two-step superlinear rate*

$$\lim_{k \to \infty} \frac{\|z_{k+1} - z^*\|}{\|z_{k-1} - z^*\|} = 0$$

The previous theorem has been proven in [6] in the orthogonal framework. The second gradient evaluation in step 3 of algorithm 1 was avoided by [17] for the orthogonal framework while retaining the convergence result under a more stringent update rule. In this paper, we use the reduced SQP method in the form of algorithm 1 because it allows an extension of the convergence result to general choices of R and T and to an infinite dimensional setting [14], in which the optimal control problem is formulated originally.

Next we discuss the choice for T and R, that is appropriate in applications in optimal control. In the formulation of the optimization problem in (4), which arises from the parabolic control problem, one can see that the variables z in (5) are split into $(y^T, u^T)^T \in \mathbb{R}^{K+M}$, the state and control variable. We can use this splitting to parametrize the null space of $h'(y, u)$ by the control. Obviously, an element $(\eta^T, \nu^T)^T$ belongs to $\mathcal{N}[h'(y, u)]$ if and only if

$$h'_y(y, u)\eta = -h'_u(y, u)\nu$$

where $h'_y \in \mathbb{R}^{K \times K}$ and $h'_u \in \mathbb{R}^{K \times M}$ denote the Jacobians of h with respect to y and u. Provided that h'_y is nonsingular, the elements in the null space can be represented as follows:

$$\begin{pmatrix}\eta\\ \nu\end{pmatrix} \in \mathcal{N}[h'(y,u)] \quad \Leftrightarrow \quad \begin{pmatrix}\eta\\ \nu\end{pmatrix} = \begin{pmatrix}-h'_y(y,u)^{-1}h'_u(y,u)\nu\\ \nu\end{pmatrix}$$

This suggests the following natural choice, which defines what could be called the separability framework:

$$T(y,u) = \begin{pmatrix}-h'_y(y,u)^{-1}h'_u(y,u)\\ I_M\end{pmatrix} \text{ and } R(y,u) = \begin{pmatrix}h'_y(y,u)^{-1}\\ \mathcal{O}\end{pmatrix}$$

where the second component in $R(y,u)$ denotes the null matrix in $\mathbb{R}^{M\times K}$. With these definitions for R and T, the reduced SQP step (7) can be written in the following way:

$$\begin{aligned}\begin{pmatrix}\Delta y\\ \Delta u\end{pmatrix} &= T(y,u)\Delta u - R(y,u)h(y,u)\\ &= \begin{pmatrix}-h'_y(y,u)^{-1}[h'_u(y,u)\Delta u + h(y,u)]\\ \Delta u\end{pmatrix}\end{aligned}$$

where Δu is the solution of

$$B\Delta u = -T(y,u)^T\nabla F(y,u)$$

Recall that for the BFGS update in algorithm 1 we have to evaluate the reduced gradient at the intermediate point

$$\begin{pmatrix}y\\ u\end{pmatrix} + T(y,u)\Delta u = \begin{pmatrix}y - h'_y(y,u)^{-1}h'_u(y,u)\Delta u\\ u + \Delta u\end{pmatrix}$$

If we insert these relations in algorithm 1, we are led to the following method, which is applicable to problems of the type (4).

2.2. Algorithm 2 (reduced BFGS method, separability approach)

Given $u \in \mathbb{R}^M$, $y \in \mathbb{R}^K$, and $B \in \mathbb{R}^{M\times M}$, B nonsingular.

1. Solve $B\Delta u = -T(y,u)^T\nabla F(y,u)$.
2. Solve $h'_y(y,u)\Delta y = -h'_u(y,u)\Delta u - h(y,u)$. Set $u_+ = u+\Delta u$ and $y_+ = y+\Delta y$.
3. Solve $h'_y(y,u)\eta = -h'_u(y,u)\Delta u$.
   Compute $v = T(y+\eta, u_+)^T\nabla F(y+\eta, u_+) - T(y,u)^T\nabla F(y,u)$.
4. Set

$$B_+ = B + \frac{vv^T}{v^T\Delta u} - \frac{(B\Delta u)(B\Delta u)^T}{\Delta u^T B\Delta u}$$

if it is well defined, else set $B_+ = B$.

The reduced BFGS method in the separability approach is adopted in [13] where reduced SQP is applied to parameter identification problems, and in [16] for the solution of semilinear parabolic control problems. In the next section we will discuss the realization of algorithm 2 for a discretized version of the parabolic control problem presented in the introduction.

3. Reduced BFGS method for discretized control problem

We follow the discretization that was proposed in [2] for the optimal control problem (1)–(3). The variational form of the boundary value problem (1), (2) can be written as

$$\left(C(y(\cdot,t))\frac{\partial y}{\partial t}(\cdot,t), v\right) - (\nabla(\lambda(y(\cdot,t))\nabla y(\cdot,t)), v) = (f(\cdot,t), v) \quad \forall v \in H^1(\Omega) \tag{8}$$

where $(\cdot,\cdot)$ denotes the inner product in $L^2(\Omega)$. We suppose the following for the specific heat capacity and the heat conduction:

ASSUMPTION 4. *$C, \lambda \in C^1(\mathbb{R})$, and there are constants $C_i, \lambda_i \in \mathbb{R}$, $i = 1,2$, such that*

$$0 < C_1 \le C(t) \le C_2, \quad 0 < \lambda_1 \le \lambda(t) \le \lambda_2 \qquad \forall t \in \mathbb{R}$$

As a consequence of Assumption 4, the following functions

$$\Gamma(s) = \int_0^s C(\tau)\,d\tau, \quad \beta(s) = \int_0^s \lambda(\tau)d\tau$$

are strictly monotone increasing. Integration by parts of (8) taking the boundary conditions into account and some calculations (see [2]) lead to

$$\begin{aligned} (\frac{\partial}{\partial t}\Gamma(y(\cdot,t)), v) + (\nabla\beta(y(\cdot,t)), \nabla v) + \langle y(\cdot,t), v\rangle &= (f(\cdot,t), v) \\ &\quad + \langle u(t), v\rangle \\ y(\cdot,t)(0) &= y_0(\cdot) \end{aligned} \tag{9}$$

where we denote $\langle v, w\rangle = gv(0)w(0)$ for $v, w \in H^1(\Omega)$. If we set

$$\phi(x,t) = \Gamma(y(x,t)), \ \phi_0(x) = \Gamma(y_0(x)), \quad \gamma(s) = \beta(\Gamma^{-1}(s))$$

then we can rewrite (9) as

$$\begin{aligned}(\frac{\partial}{\partial t}\phi(\cdot, t), v) + (\nabla\gamma(\phi(\cdot,t)), \nabla v) + \langle \Gamma^{-1}(\phi(\cdot, t)), v\rangle &= (f(\cdot, t), v) \\ &\quad + \langle u(t), v\rangle \qquad (10)\\ \phi(\cdot, t)(0) &= \phi_0(\cdot)\end{aligned}$$

which is linear with respect to the time differentiation.

Set $V = H^1(\Omega)$ and let $f \in L^2(0, T; L^2(\Omega)), y_0 \in L^2(\Omega)$ and $\hat{y} \in L^2(0, T)$ be given. To discretize the control problem the space interval is divided into N subintervals of equidistant length $h = 1/N$ with grid points $x_i = (i-1)h$, $i = 1, \ldots, N+1$. The time discretization is performed by partitioning the interval $[0, T]$ into M equidistant subintervals of length $\tau = T/M$ with grid points $t^j = (j-1)\tau$, $j = 1, \ldots, M+1$. We introduce a finite dimensional subspace V^N of V

$$V^N = \text{span}\{b_1, \ldots, b_{N+1}\}$$

where $\{b_1, \ldots, b_{N+1}\}$ is the basis of linear spline functions satisfying

$$b_i(x_j) = \delta_{ij}, \quad i, j = 1, \ldots, N+1$$

The state space $Y = L^2(0, T; V)$ is approximated by the subspace Y^{MN} of functions from $(0, T]$ into V^N that are constant on each interval $(t^j, t^{j+1}]$:

$$Y^{MN} = \{w(x, t) = \sum_{j=1}^{M} w^{j+1}(x)\mathcal{X}_j(t); w^j \in V^N, j = 2, \ldots, M+1\}$$

where $\mathcal{X}_j$ denotes the characteristic function on $(t^j, t^{j+1}]$. The control space $U = L^2(0, T)$ is approximated by the subset U^M of piecewise constant functions on $(0, T]$.

To derive a discretized form for the boundary value problem we assume for a moment that a control $u = \Sigma_{j=1}^{M} u^{j+1}\mathcal{X}_j \in U^M$ is given. The inhomogeneous term f and the initial temperature y_0 are replaced by the elements $f^j \in V^N$ and $y_{0,N} \in V^N$, respectively, which satisfy

$$\begin{aligned}(f^{j+1}, v) &= \tfrac{1}{\tau}\int_{t^j}^{t^{j+1}} (f(\cdot, t), v)dt \quad \forall v \in V^N, \quad j = 1, \ldots, M\\ (y_0, N, v) &= (y_0, v) \qquad\qquad \forall v \in V^N\end{aligned}$$

Using these approximations a discretization for (10) is given by

$$\begin{aligned}(1/\tau)(\phi^{j+1} - \phi^j, v) + (\nabla\gamma(\phi^{j+1}), \nabla v) + \langle \Gamma^{-1}(\phi^{j+1}), v\rangle &= (f^{j+1}, v) \\ &\quad + \langle u^{j+1}, v\rangle (11)\\ \phi^1 &= r\Gamma(y_0, N)\end{aligned}$$

to hold for all $v \in V^N$ and some $\phi^j \in V^N$, $j = 1, \ldots, M+1$, where r denotes the restriction mapping

$$r\upsilon = \sum_{i=1}^{N+1} \upsilon(x_i) b_i, \qquad \upsilon \in C[0, 1]$$

Since we use y and not ϕ in the objective function of the control problem, we transform (11) back into an approximating scheme for (9). Let $y^j \in V^N$ be the elements that satisfy $\phi^j = r\Gamma(y^j)$, $j = 1, \ldots, M+1$. Then the relations

$$\begin{aligned} \langle \Gamma^{-1}(\phi^j), \upsilon \rangle &= \langle y^j, \upsilon \rangle \quad \text{and} \\ (\nabla\gamma(\phi^j), \nabla\upsilon) &= (\nabla\beta(y^j), \nabla\upsilon) = (\nabla r\beta(y^j), \nabla\upsilon) \end{aligned}$$

are true for all $\upsilon \in V^N$, $j = 1, \ldots, M+1$, and we can rewrite (11) as

$$\begin{aligned} (1/\tau)(r\Gamma(y^{j+1}) - r\Gamma(y^j), b_i) + (\nabla r\beta(y^{j+1}), \nabla b_i) + \langle y^{j+1}, b_i \rangle \\ = (f^{j+1}, b_i) + \langle u^{j+1}, b_i \rangle \end{aligned} \tag{12}$$

with $y^1 = y_{0,N}$ to hold for $i = 1, \ldots, N+1$ and some $y^j \in V^N$, $j = 1, \ldots, M+1$.

We arrive at a nonlinear system of equations by defining the tridiagonal matrices A and D in $\mathbb{R}^{(N+1)\times(N+1)}$ through

$$A = (1/\tau)((b_i, b_j))_{i,j=1,\ldots,N+1} \quad \text{and} \quad D = ((\nabla b_i, \nabla b_j))_{i,j=1,\ldots,N+1}$$

From (15) in [2] we have for all $w \in V^N$

$$\begin{aligned} &(1/\tau)(r\Gamma(w), b_i) = (A\Gamma(\overline{w}))_i \quad \text{and} \\ &(\nabla r\beta(w), \nabla b_i) = (D\beta(\overline{w}))_i \quad i = 1, \ldots, N+1 \end{aligned}$$

where $\overline{w} \in \mathbb{R}^{N+1}$ denotes the coefficient vector of w and

$$\Gamma(\overline{w}) = (\Gamma(w_j); j = 1, \ldots, N+1)^T, \quad \beta(\overline{w}) = (\beta(w_j); j = 1, \ldots, N+1)^T$$

Then (12) is equivalent to solve successively for $j = 1, \ldots, M$ the following nonlinear equations

$$\begin{aligned} A\Gamma(\overline{y}^{j+1}) + D\beta(\overline{y}^{j+1}) + (gy_1^{j+1}, 0, \ldots, 0)^T \\ = a^j + (gu^{j+1}, 0, \ldots, 0)^T + A\Gamma(\overline{y}^j) \end{aligned} \tag{13}$$

where $\overline{y}^1$ is obtained from the solution of

$$(\tau A)\overline{y}^1 = ((y_0, b_i); i = 1, \ldots, N+1)^T \tag{14}$$

and $a^j \in \mathbb{R}^{N+1}$ denotes the vector with the components

$$a_i^j = \frac{1}{\tau} \int_{t^j}^{t^{j+1}} (f(\cdot, t), b_i) dt, \quad i = 1, \ldots, N+1, \; j = 1, \ldots, M \tag{15}$$

Since we consider only the discretized problem from here on, we omit the bar and denote the coefficient vectors simply by

$$y = ((y^2)^T, \ldots, (y^{M+1})^T)^T \in \mathbb{R}^{M(N+1)} \text{ and } u = (u^2, \ldots, u^{M+1})^T \in \mathbb{R}^M$$

We consider y and u as independent variables and write (13) as nonlinear equality constraints in the notation of problem (4). Let

$$h : \mathbb{R}^{M(N+1)} \times \mathbb{R}^M \to \mathbb{R}^{M(N+1)}$$

where the components $h^j : \mathbb{R}^{M(N+1)} \times \mathbb{R}^M \to \mathbb{R}^{N+1}$ are defined as

$$h^1(y, u) = A\Gamma(y^2) + D\beta(y^2) + g(y_1^2 - u^2)e^1 - (a^1 + A\Gamma(y^1)) \tag{16}$$

and for $j = 3, \ldots, M+1$

$$h^{j-1}(y, u) = A(\Gamma(y^j) - \Gamma(y^{j-1})) + D\beta(y^j) + g(y_1^j - u^j)e^1 - a^{j-1} \tag{17}$$

Here y^1 denotes the solution of (14), a^j is defined by (15), and $e^1 = (1, 0, \ldots, 0)^T$ in $\mathbb{R}^{N+1}$.

For the approximation of the objective function in (3) we replace $y(1, \cdot)$, u and $\hat{y}$ by their discrete expressions

$$y_{MN}(1, \cdot) = \sum_{j=1}^{M} y_{N+1}^{j+1}\chi_j, \quad u_M = \sum_{j=1}^{M} u^{j+1}\chi_j, \quad \text{and} \quad \hat{y}_M = \sum_{j=1}^{M} \hat{y}^{j+1}\chi_j$$

where the data $\hat{y}^j$ are computed by integrating $\hat{y}$ on the subinterval $(t^j, t^{j+1}]$:

$$\hat{y}^{j+1} = \frac{1}{\tau}\int_{t^j}^{t^{j+1}} \hat{y}(t)dt, \quad j = 1, \ldots, M \tag{18}$$

Then we have the following objective in the discrete control problem

$$\text{Minimize } F(y, u) = \tau \sum_{j=2}^{M+1} [(y_{N+1}^j - \hat{y}^j)^2 + \alpha(u^j)^2] \tag{19}$$

The definitions (16), (17), and (19) completely describe the discrete optimal control problem in the notation of the optimization problem (4). To apply algorithm 2 we now focus on the Jacobian of h and on the computation of the reduced gradient. For $j = 2, \ldots, M+1$ we let $Q_j \in \mathbb{R}^{(N+1)\times M}$ denote the matrix with the elements

$$(Q_j)_{ik} = \begin{cases} -g & , \text{ if } i = 1 \text{ and } k = j - 1 \\ 0 & \text{else} \end{cases} \tag{20}$$

and for $w \in \mathbb{R}^{N+1}$ we define the following tridiagonal matrices in $\mathbb{R}^{(N+1)\times(N+1)}$

$$\begin{aligned} G(w) &= AC_d(w) + D\lambda_d(w) + \operatorname{diag}(g, 0, \ldots, 0) \\ E(w) &= -AC_d(w) \end{aligned} \tag{21}$$

with the diagonal matrices

$$C_d(w) = \operatorname{diag}(C(w_1), \ldots, C(w_{N+1})) \text{ and } \lambda_d(w) = \operatorname{diag}(\lambda(w_1), \ldots, \lambda(w_{N+1}))$$

Then we can write the Jacobian of h as

$$h'(y, u) = (h'_y(y, u), h'_u(y, u))$$

where

$$h'_y(y, u) = \begin{pmatrix} G(y^2) & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ E(y^2) & G(y^3) & & & & & \cdot \\ 0 & \cdot & \cdot & & & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ \cdot & & & & \cdot & \cdot & \cdot \\ \cdot & & & & E(y^{M-1}) & G(y^M) & 0 \\ 0 & \cdot & \cdot & \cdot & 0 & E(y^M) & G(y^{M+1}) \end{pmatrix} \in \mathbb{R}^{M(N+1)\times M(N+1)}$$

and

$$h'_u(y, u) = (Q_2^T, Q_3^T, \ldots, Q_{M+1}^T)^T \in \mathbb{R}^{M(N+1)\times M} \tag{22}$$

Note that $h'_y(y, u) = h'_y(y)$ only depends on the state variables, and $h'_u(y, u) = h'_u$ is a constant matrix. In the previous section we assumed that h'_y is nonsingular and we used the partial Jacobians to derive an appropriate null space basis and right-inverse. In connection with the discussion of Newton's method for solving the discrete scheme (13) it is shown in [2] that the matrix $G(w)$ is nonsingular for all $w \in \mathbb{R}^{N+1}$ if the step lengths h and τ are chosen properly. From this result we can conclude that h'_y is invertible.

LEMMA 1. *Let Assumption 4 hold and assume that h and τ are such that*

$$\frac{\tau}{h^2} < \frac{1}{6}\left(\frac{\lambda_2}{C_1} - \frac{\lambda_1}{C_2}\right)^{-1} \tag{23}$$

Then $h'_y(y)$ is nonsingular for all $y \in \mathbb{R}^{M(N+1)}$.

For the sake of completeness we state our choice for R and T in the following lemma.

LEMMA 2. *Let Assumption 4 be satisfied and suppose that (23) holds. Then*

$$T(y) := \begin{pmatrix} -h'_y(y)^{-1}h'_u \\ I_M \end{pmatrix} \in \mathbb{R}^{M(N+2)\times M} \tag{24}$$

is a basis for the null space of $h'(y, u)$, *and*

$$R(y) := \begin{pmatrix} h'_y(y)^{-1} \\ \mathcal{O} \end{pmatrix} \in \mathbb{R}^{M(N+2)\times M(N+1)}$$

is a right-inverse for $h'(y, u)$ *for all* $y \in \mathbb{R}^{M(N+1)}$ *and* $u \in \mathbb{R}^M$. *Here* $\mathcal{O}$ *denotes the null matrix in* $\mathbb{R}^{M\times M(N+1)}$.

Proof. Since $h'_y(y)$ is nonsingular, the operators T and R are well defined on $\mathbb{R}^{M(N+1)}$. Furthermore, $T(y)$ has full rank, and for all $\nu \in \mathbb{R}^M$

$$h'(y, u)T(y)\nu = (h'_y(y), h'_u)\begin{pmatrix} -h'_y(y)^{-1}h'_u\nu \\ \nu \end{pmatrix} = 0$$

If on the other hand for $\eta \in \mathbb{R}^{M(N+1)}$, $\nu \in \mathbb{R}^M$ the relation

$$0 = h'(y, u)\begin{pmatrix} \eta \\ \nu \end{pmatrix} = h'_y(y)\eta + h'_u\nu$$

holds, then $\eta = -h'_y(y)^{-1}h'_u\nu$, and, consequently, $(\eta^T, \nu^T)^T \in \mathcal{R}[T(y)]$.

Obviously, $h'(y, u)R(y)\eta = \eta$ for all $\eta \in \mathbb{R}^{M(N+1)}$. □

Note that the representation (24) of the null space basis is not only needed for the definition of the reduced SQP step, but it can be used further for the calculation of the corresponding reduced gradient. We show more generally

LEMMA 3. *Assume that Assumption 4 and (23) hold, and let* $\eta \in \mathbb{R}^{M(N+1)}$ *and* $\nu \in \mathbb{R}^M$ *be given. Then for all* $y \in \mathbb{R}^{M(N+1)}$

$$T(y)^T\begin{pmatrix} \eta \\ \nu \end{pmatrix} = g(\pi_1^2, \pi_1^3, \ldots, \pi_1^{M+1})^T + \nu \tag{25}$$

where $\pi = ((\pi^2)^T, (\pi^3)^T, \ldots, (\pi^{M+1})^T)^T \in \mathbb{R}^{M(N+1)}$ *is the solution of the scheme*

$$\begin{aligned} G(y^{M+1})^T\pi^{M+1} &= \eta^{M+1} \\ G(y^j)^T\pi^j &= \eta^j - E(y^j)^T\pi^{j+1}, \quad j = M, \ldots, 2 \end{aligned}$$

Proof. Obviously, π is the solution of $h'_y(y)^T\pi = \eta$. Then (20), (22), and (24) imply that

$$\begin{aligned} T(y)^T\begin{pmatrix} \eta \\ \nu \end{pmatrix} &= (-h'^T_u h'_y(y)^{-T}, I_M)\begin{pmatrix} \eta \\ \nu \end{pmatrix} \\ &= -h'^T_u\pi + \nu = g(\pi_1^2, \ldots, \pi_1^{M+1})^T + \nu \end{aligned}$$

which proves the assertion. □

Lemma 3 can be applied to compute the reduced gradient $T(y)^T\nabla F(y, u)$, where the partial derivatives of the objective function in (19) are given by

$$\frac{\partial F}{\partial y_i^j}(y, u) = \left\{ \begin{array}{rl} 2\tau(y_{N+1}^j - \hat{y}^j) & \text{for } i = N+1 \\ 0 & \text{for } i = 1, \ldots, N \end{array} \right\} \text{for } j = 2, \ldots, M+1$$

and $\nabla_u F(y, u) = 2\alpha\tau u$. Hence, all the steps in algorithm 2 can be formulated for the optimal control problem. We set $e^{N+1} = (0, \ldots, 0, 1)^T \in \mathbb{R}^{N+1}$, and we note that the right-hand side in the linear equation for the state step can be simplified to

$$h_u' \Delta u + h(y, u) = h(y, u + \Delta u)$$

In detail, the reduced BFGS algorithm for the discrete optimal control problem (19) subject to (16) and (17) requires the following steps:

3.1. Algorithm 3 (reduced BFGS method for DOCP, separability approach)

Given $u \in \mathbb{R}^M$, $y \in \mathbb{R}^{M(N+1)}$, and a positive definite matrix $B \in \mathbb{R}^{M \times M}$.

Step 1 (computation of the reduced gradient $T(y)^T \nabla F(y, u)$):
Compute the adjoint state $\pi = ((\pi^2)^T, \ldots, (\pi^{M+1})^T)^T$ from

$$G(y^{M+1})^T \pi^{M+1} = 2\tau(y_{N+1}^{M+1} - \hat{y}^{M+1}) e^{N+1}$$

and successively for $j = M, \ldots, 2$ from

$$G(y^j)^T \pi^j = 2\tau(y_{N+1}^j - \hat{y}^j) e^{N+1} - E(y^j)^T \pi^{j+1}$$

Set $c(y, u) = g(\pi_1^2, \ldots, \pi_1^{M+1})^T + 2\alpha\tau u$.

Step 2 (computation of the new control):
Solve $B\Delta u = -c(y, u)$.
Set $u_+ = u + \Delta u$.

Step 3 (computation of the new state and of the intermediate point):
Determine $\xi = ((\xi^2)^T, \ldots, (\xi^{M+1})^T)^T$ and $\eta = ((\eta^2)^T, \ldots, (\eta^{M+1})^T)^T$ from the solution of

$$G(y^2)\xi^2 = -h^1(y, u_+), \quad G(y^2)\eta^2 = g(\Delta u)^2 e^1$$

and successively for $j = 3, \ldots, M+1$ from

$$\begin{aligned} G(y^j)\xi^j &= -h^{j-1}(y, u_+) - E(y^{j-1})\xi^{j-1} \\ G(y^j)\eta^j &= g(\Delta u)^j e^1 - E(y^{j-1})\eta^{j-1} \end{aligned}$$

Set $y_+ = y + \xi$ and $\tilde{y} = y + \eta$.

Step 4 (computation of the reduced gradient $T(\tilde{y})^T \nabla F(\tilde{y}, u_+)$):
Compute $\pi = ((\pi^2)^T, \ldots, (\pi^{M+1})^T)^T$ from

$$G(\tilde{y}^{M+1})^T \pi^{M+1} = 2\tau(\tilde{y}_{N+1}^{M+1} - \hat{y}^{M+1})e^{N+1}$$

and for $j = M, \ldots, 2$ from

$$G(\tilde{y}^j)^T \pi^j = 2\tau(\tilde{y}_{N+1}^j - \hat{y}^j)e^{N+1} - E(\tilde{y}^j)^T \pi^{j+1}$$

Set $c(\tilde{y}, u_+) = g(\pi_1^2, \ldots, \pi_1^{M+1})^T + 2\alpha\tau u_+$.

Step 5 (computation of B_+):

Set $v = c(\tilde{y}, u_+) - c(y, u)$.

Set

$$B_+ = B + \frac{vv^T}{v^T \Delta u} - \frac{(B\Delta u)(B\Delta u)^T}{\Delta u^T B \Delta u}$$

if it is well defined, else set $B_+ = B$.

Remark 1. The computation of the step in y and of the intermediate step η (step 3 in algorithm 3) involve the solution of the same systems of linear equations with different inhomogeneous terms. The adjoint equations in step 1 and step 4 are not the same because the data in the matrices and in the right-hand sides differ. The computation of y_+, $\tilde{y}$, and of the reduced gradients is rather cheap, since only tridiagonal systems need to be solved. We note further that due to the appropriate choice of the null space basis and of the right-inverse, the sparsity of the Jacobian h' could be used efficiently in the course of algorithm 3, which is reflected in steps 1, 3, and 4.

Remark 2. One advantage of algorithm 3 over a general SQP method is the dimension of the secant approximation. We only have to store and update the matrix B which is of order M in contrast to a matrix in $\mathbb{R}^{M(N+2)\times M(N+2)}$ in an SQP secant method. Since for the solution of the tridiagonal systems only vectors in $\mathbb{R}^{N+1}$ need to be stored, algorithm 3 is practicable also with regard to storage requirements.

Remark 3. The complexity of algorithm 3 is dominated by the cost for the factorization of B and the computation of Δu in step 2. More precisely, the total amount of work in one iteration of the algorithm is the solution of $4M$ tridiagonal systems each of dimension $N + 1$, plus some negligible matrix-vector multiplications (steps 1, 3, 4), and the solution of a positive definite system in step 2 after performing the update in step 5. We implemented the BFGS update in its factored form (see [7]), i.e., updating the Cholesky factorization of B to obtain the Cholesky decomposition of B_+. For this implementation, the total cost of updating the reduced Hessian approximation and obtaining its Cholesky factorization and solving the linear system for Δu is $O(M^2)$. Consequently, if $M > N$ which is the situation in our numerical experiments, the complexity of one iteration of algorithm 3 is $O(M^2)$. In comparison, the implementation of Newton's method that is used in [2] requires $O(M^3)$ elementary operations per iteration. Recall that in [2] the optimal control problem is interpreted as

an unconstrained optimization problem so that the Hessian and the gradient of the corresponding objective function must be calculated in order to compute the Newton step. It is reported in [2] that for the calculation of the Hessian approximately $2M^2$ tridiagonal systems in $\mathbb{R}^{N+1}$ must be solved. The Newton step is computed by means of Gaussian limination so that the cost both for the calculation of the Hessian and for its factorization is $O(M^3)$. The gradient is computed from the solution of the adjoint equation, which is the scheme given in step 1 of algorithm 3. However, since the state is considered as a variable depending on the control, the data y in the adjoint equation must be computed from the solution of the set of nonlinear equations (13). In [2] Newton's method is used for the solution of (13). Note that in algorithm 3 only a linearization of these equations is solved to compute the step in the state space.

The computational results in the next section show that a fast two-step superlinear rate of convergence for algorithm 3 can be observed numerically.

4. Numerical Experiments

We implemented algorithm 3 for the discrete optimal control problem (19) subject to (16) and (17). The Kirchnoff and the enthalpic transformations β and Γ were calculated analytically and the integrals in (14), (15), and (18) were approximated by the Simpson rule. The BFGS update was implemented in its factored form as described in the previous section. LINPACK was used for linear algebra manipulations. The computations were done in double-precision FORTRAN on a SUN Sparcstation 1.

To evaluate the performance of the algorithm we will present numerical results for three examples. The desired state $\hat{y}$ comes from interpolation of an attainable state for the infinite-dimensional unregularized control problem. With exception of the last problem, the parabolic boundary value problems used in this process were taken from [1] with some modification in the parameter values.

In the tables we use the following notation

$$\rho_k = (\|u_{M,k} - u_{M,k-1}\|_U^2 + \|y_{MN,k} - y_{MN,k-1}\|_Y^2)^{1/2}$$

$$F_k = F(y_k, u_k), \quad \text{and} \quad \phi_k = \|h(y_k, u_k\|_Y$$

Here the subscript k denotes the iteration number and $y_{MN,k}$, $u_{M,k}$ denote the functions in Y^{MN} and U^M with the coefficient vectors h and F respectively. Recall that y_k and u_k were defined in (16), (17), and (19).

We want to document primarily the efficiency of the proposed method, and we want to demonstrate that the two-step superlinear convergence rate can be observed numerically. Under standard assumptions for linear convergence results it can be shown, see e.g., [15], that the iterates generated by a reduced secant method converge l-step superlinearly, $l = 1, 2$, if and only if the corresponding

sequences for the steps ρ_k/ρ_{k-1} respectively ρ_k/ρ_{k-2} tend to zero. We monitor these ratios for a discussion of the rate of convergence properties of the algorithm.

To fully concentrate on the local convergence behavior of the method we did not use any globalization strategy in the first two examples. In most of the test cases the algorithm was successful, even when the starting values, in particular those for the state variables, were not chosen too close to the solution. We started the minimization with $B_0 = \eta I$, where the constant $\eta > 0$ was selected to reflect the scaling of the control variables.

4.1. Example 1

The heat capacity and the heat conduction are given by linear functions

$$\begin{aligned} C(y) &= q_1 + q_2 y, \quad y \in I\!R \\ \lambda(y) &= r_1 + r_2 y, \quad y \in I\!R \end{aligned}$$

where r_1, q_1, r_2, and q_2 are constants such that both functions take positive values in the range of variation of the temperature y.

If we choose the following data

$$\begin{aligned} \hat{y}(t) &= 2 - e^{\rho t} \\ f(x, t) &= [\rho(q_1 + 2q_2) + \pi^2(r_1 + 2r_2)]e^{\rho t}\cos\pi x \\ &\quad -r_2\pi^2 e^{2\rho t} + (2r_2\pi^2 + \rho q_2)e^{2\rho t}\cos^2\pi x \\ y_0(x) &= 2 + \cos\pi x \end{aligned}$$

with $\rho < 0$, then an optimal solution of the control problem (1)–(3) for $\alpha = 0$ is given by

$$\begin{aligned} y_*(x, t) &= 2 + e^{\rho t}\cos\pi x \\ u_*(t) &= 2 + e^{\rho t} \end{aligned}$$

In this example we set

$$T = 0.5, \quad g = 1.0, \quad \alpha = 0.0$$

and the parameter values are chosen as follows

$$r_1 = q_1 = 4.0, \quad r_2 = -1.0, \quad q_2 = 1.0, \quad \rho = -1.0$$

In Table 1 we select $B_0 = I$, and we consider a close approximation for $y_*(1, \cdot)$ and for u_* by choosing

$$(y_0)^j_{N+1} = \hat{y}^j, \quad (u_0)^j = u^j_* + 0.05, \quad j = 2, \ldots, M + 1$$

The remaining state variables $(y_0)^j_i$, $i = 1, \ldots, N$, $j = 2, \ldots, M + 1$ are set to zero. The fifth column indicates a two-step superlinear convergence rate, while from column four a one-step superlinear rate is not observable.

Table 1. Two-step superlinear convergence for example 1 with $N = 18$, $M = 100$.

k	ϕ_k	ρ_k	ρ_k/ρ_{k-1}	ρ_k/ρ_{k-2}	F_k
0	0.498E+03	–	–	–	0.000E+00
1	0.120E+03	0.282E+01	–	–	0.873E+00
2	0.190E+03	0.342E+01	1.21556	–	0.112E+00
3	0.156E+02	0.144E+01	0.42107	0.51184	0.880E−03
4	0.307E+00	0.508E+00	0.35272	0.14852	0.543E−05
5	0.319E−01	0.193E+00	0.37941	0.13383	0.283E−05
6	0.179E+00	0.411E+00	2.13226	0.80901	0.114E−06
7	0.847E−02	0.759E−01	0.18453	0.39346	0.131E−08
8	0.494E−04	0.632E−02	0.08331	0.01537	0.177E−09
9	0.597E−06	0.804E−03	0.12716	0.01059	0.159E−09
10	0.396E−10	0.652E−05	0.00811	0.00103	0.159E−09
11	0.396E−10	0.726E−07	0.01113	0.00009	0.159E−09

For the same example we change the data in the objective function in order to reduce the effect of the discretization error. We use Newton's method to solve the discrete state equation (13) for a given discretization level with u_* as input. The values of the computed solution y then serve as data in the discrete control problem, i.e., we choose

$$\hat{y}^j = y^j_{N+1}, \quad j = 2, \ldots, M + 1$$

We initialize the iteration process with

$$y_0 \equiv 0 \text{ and } u_0 \equiv 2.6$$

and B_0 is a particular positive scaling matrix that causes the first step in u to be in a scaled steepest descent direction. The results are documented in Table 2, where the fifth column depicts the value of $||u_{M,k} - u_*||_U$. This provides information on the performance of the algorithm with respect to the unknown control u_*. It should be noted that for this example the algorithm does not only exhibit a two-step but also a one-step superlinear convergence rate.

Table 2. Two-step superlinear convergence for example 1 with $N = 18$, $M = 100$.

k	ϕ_k	ρ_k	ρ_k/ρ_{k-2}	$\|\|u_{M,k} - u_*\|\|_U$	F_k
0	0.316E+02	–	–	0.155E+00	0.742E+00
1	0.152E+02	0.204E+01	–	0.771E−01	0.295E+00
2	0.137E+01	0.694E+00	–	0.229E+00	0.527E−03
3	0.383E−01	0.677E−01	0.033224	0.216E+00	0.129E−05
4	0.173E−01	0.155E+00	0.222775	0.661E−01	0.111E−06
5	0.313E−02	0.608E−01	0.897424	0.760E−02	0.152E−08
6	0.522E−04	0.762E−02	0.049249	0.735E−03	0.978E−11
7	0.323E−06	0.626E−03	0.010290	0.777E−03	0.240E−14
8	0.599E−10	0.858E−05	0.001126	0.780E−03	0.577E−15
9	0.606E−12	0.461E−07	0.000074	0.780E−03	0.577E−15
10	0.595E−12	0.225E−10	0.000003	0.780E−03	0.577E−15

4.2. Example 2

The functions C and λ are chosen as in example 1. For

$$\begin{aligned} \hat{y}(t) &= \frac{1}{2}(1 - e^{-t}) \\ f(x, t) &= q(x)e^{-t}[q_1 + q_2 q(x)w(t)] - w(t)[r_1 + r_2 w(t)(\frac{3}{2}x^2 - 3x + 2)] \\ y_0(x) &= 0 \end{aligned}$$

where we denote $q(x) = \frac{x^2}{2} - x + 1$ and $w(t) = 1 - e^{-t}$, we obtain the solution

$$\begin{aligned} y_*(x, t) &= (\frac{x^2}{2} - x + 1)(1 - e^{-t}) \\ u_*(t) &= (1 - e^{-t})[1 + \frac{1}{g}(r_1 + r_2(1 - e^{-t}))] \end{aligned}$$

We choose $g = 1.0$ and compute the solution until $T = 0.1$ with a discretization level of $N = 10$ and $M = 100$.

In Table 3 we consider the unregularized problem ($\alpha = 0$) and we compute the solution for the parameter set

$$q_1 = q_2 = 0.5, \quad r_1 = 1.0, \quad r_2 = -0.5$$

where the algorithm is started with

$$B_0 = 0.25I, \quad y_0 \equiv -0.74, \quad \text{and} \quad u_0 \equiv 0.185$$

The results indicate the superlinear convergence behavior as predicted by the theory.

Table 3. Two-step superlinear convergence for example 2 with $\alpha = 0$.

k	ϕ_k	ρ_k	ρ_k/ρ_{k-1}	ρ_k/ρ_{k-2}	F_k
0	0.196E+01	–	–	–	0.584E−01
1	0.617E+01	0.531E+00	–	–	0.920E−01
2	0.119E+01	0.231E+00	0.4355	–	0.503E−02
3	0.111E+00	0.613E−01	0.2653	0.11554	0.386E−04
4	0.153E−02	0.568E−02	0.0926	0.02456	0.658E−06
5	0.242E−02	0.453E−01	7.9760	0.73850	0.428E−07
6	0.107E−03	0.938E−02	0.2071	1.65181	0.337E−08
7	0.140E−06	0.336E−03	0.0358	0.00742	0.331E−08
8	0.366E−08	0.563E−04	0.1678	0.00601	0.331E−08
9	0.148E−13	0.113E−06	0.0020	0.00034	0.331E−08
10	0.149E−14	0.262E−08	0.0233	0.00005	0.331E−08

In Table 4 we use the data

$$q_1 = q_2 = 0.2, \quad r_1 = 1.0, \quad r_2 = -0.5, \quad \alpha = 10^{-3}$$

and we start with $B_0 = (2 \cdot 10^{-6})I$, $u_0 \equiv 0.1$, and

$$(y_0)_i^j = (y_*)_i^j, \quad i = 1, \ldots, N+1, \quad j = 2, \ldots, M+1$$

Note that the parameter α is fairly large, so that the initial values y_0 cannot be regarded as an exact approximation to the state variables.

Table 4. Two-step superlinear convergence for example 2 with $\alpha = 10^{-3}$.

k	ϕ_k	ρ_k	ρ_k/ρ_{k-1}	ρ_k/ρ_{k-2}	F_k
0	0.539E−01	–	–	–	0.100E−05
1	0.225E−02	0.341E−01	–	–	0.362E−05
2	0.911E−03	0.241E−01	0.70579	–	0.549E−06
3	0.527E−03	0.218E−01	0.90390	0.63797	0.821E−06
4	0.191E−03	0.134E−01	0.61422	0.55519	0.380E−06
5	0.148E−05	0.109E−02	0.08171	0.05019	0.380E−06
6	0.146E−06	0.357E−03	0.32729	0.02674	0.380E−06
7	0.486E−10	0.692E−05	0.01937	0.00634	0.380E−06
8	0.106E−11	0.101E−05	0.14636	0.00283	0.380E−06
9	0.641E−15	0.161E−08	0.00159	0.00023	0.380E−06
10	0.638E−15	0.930E−10	0.05771	0.00009	0.380E−06

The last example was designed to demonstrate that the globalization procedure that we use currently in our code works well in practice. We implemented the globalizing technique that is proposed in [8] for general reduced SQP methods. In [8] a second-order correction step is added to the reduced SQP step (7) under certain conditions. The step-length is then determined along an arc-shaped search path in order to decrease an exact penalty function with the help of an Armijo-like criterion. It is shown in [8] that with this strategy the "Maratos effect" is avoided, i.e., the step-size is equal to one after a finite number of iterations.

4.3. Example 3

We choose C and λ again as linear functions

$$\begin{aligned} C(y) &= q_1 + q_2 y, \quad y \in I\!R \\ \lambda(y) &= r_1 + r_2 y, \quad y \in I\!R \end{aligned}$$

With the data

$$\hat{y}(t) = -\cos \pi t$$

$$
\begin{aligned}
f(x, t) &= -\pi[q_1 + q_2 x(x-2)\cos \pi t]x(x-2)\sin \pi t \\
&\quad -2[r_1 + r_2(3x^2 - 6x + 2)\cos \pi t]\cos \pi t \\
y_0(x) &= x(x-2)
\end{aligned}
$$

we obtain the following solution for the unregularized control problem

$$
\begin{aligned}
y_*(x, t) &= x(x-2)\cos \pi t \\
u_*(t) &= \frac{2r_1}{g}\cos \pi t
\end{aligned}
$$

We solve the problem for $N = 10$, $M = 100$, and the following set of parameters

$$r_1 = q_1 = 4.0, \quad r_2 = -1.0, \quad q_2 = 1.0, \quad T = 0.5, \quad g = 0.5, \quad \alpha = 5 \cdot 10^{-5}$$

The iteration is stopped, if

$$(\|T(y_k)^T \nabla F(y_k, u_k)\|_U^2 + \|h(y_k, u_k)\|_Y^2)^{1/2} < 10^{-8}$$

With the start values

$$B_0 = (5 \cdot 10^{-7})I, \quad y_0 \equiv 0, \quad \text{and} \quad u_0 \equiv 0$$

the global method yields the solution after 9 iterations, while without globalization the algorithm terminates after 19 iterations.

Table 5. Global convergence for example 3.

k	ϕ_k	ρ_k	ρ_k/ρ_{k-1}	ρ_k/ρ_{k-2}	F_k
0	0.103E+02	–	–	–	0.250E+00
1	0.996E+01	0.728E+01	–	–	0.172E+00
2	0.605E+01	0.243E+02	3.33944	–	0.181E+00
3	0.331E+01	0.211E+02	0.86691	2.89499	0.170E−01
4	0.116E+00	0.799E+01	0.37949	0.32898	0.250E−01
5	0.129E+00	0.588E+01	0.73610	0.27934	0.372E−02
6	0.201E−01	0.352E+01	0.59760	0.43990	0.293E−02
7	0.822E−03	0.132E+01	0.37676	0.22515	0.282E−02
8	0.506E−06	0.120E+00	0.09078	0.03420	0.282E−02
9	0.256E−09	0.760E−02	0.06316	0.00573	0.282E−02

References

1. J. Burger and C. Machbub, "*Comparison of numerical solutions of a one-dimensional nonlinear heat equation*," Commun. Appl. Numer. Methods, 7, pp. 1–14, 1991.
2. J. Burger and M. Pogu, "*Functional and numerical solution of a control problem originating from heat transfer*," *J. Optim. Theory Appl.*, 68, pp. 49–73, 1991.
3. R.H. Byrd, "*An example of irregular convergence in some constrained optimization methods that use the projected Hessian*," *Math. Programming*, 32, pp. 232–237, 1985.
4. R.H. Byrd, "*On the convergence of constrained optimization methods with accurate Hessian information on a subspace*," *SIAM J. Numer. Anal.*, 27, pp. 141–153, 1990.
5. R.H. Byrd and J. Nocedal, "*An analysis of reduced Hessian methods for constrained optimization*," *Math. Programming*, 49, pp. 285–323, 1991.
6. T.F. Coleman and A. R. Conn, "*On the local convergence of a quasi–Newton method for the nonlinear programming problem*," *SIAM J. Numer. Anal.*, 21, pp. 755–769, 1984.
7. J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall: Englewood Cliffs, NJ, 1983.
8. D. Gabay, "*Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*," *Math. Programming Study*, 16, pp. 18–44, 1982.
9. J.C. Gilbert, "*Une méthode de quasi-Newton réduite en optimisation sous contraintes avec priorité à la restauration*," in *Proc. of the Seventh Int. Conf. on Analysis and Optimization of Systems*, A. Bensoussan and J.L. Lions, eds., Springer: Heidelberg, Germany, pp. 40–53, 1986.
10. J.C. Gilbert, "*On the local and global convergence of a reduced quasi-Newton method*," *Optimization*, 20, pp. 421–450, 1989.
11. C. B. Gurwitz, *Local convergence of a two-piece update of a projected Hessian matrix*, Department of Computer and Information Science, Brooklyn College, Brooklyn, NY, tech. report, 1991.
12. D.H. Jacobson and D.Q. Mayne, *Differential Dynamic Programming*, Elsevier, New York, 1970.
13. K.Kunisch and E. W. Sachs, "*Reduced SQP methods for parameter identification problems*," *SIAM J. Numer. Anal.*, to appear.
14. F.-S. Kupfer, *An infinite-dimensional convergence theory for reduced SQP methods in Hilbert space*, Universität Trier, Fachbereich IV – Mathematick, tech. rep., 1990.
15. F.-S. Kupfer, *Reduced successive quadratic programming in Hilbert space with applications to optimal control*, doctoral thesis, Universität Trier, 1992.
16. F.-S. Kupfer and E. W. Sachs, "*A prospective look at SQP methods for semilinear parabolic control problems*," in *Optimal Control of Partial Differential Equations*, Irsee 1990, K.-H. Hoffmann and W. Krabs, eds., vol. 149, Springer Lect. Notes in Control and Information Sciences, 1991, pp. 143–157.
17. J. Nocedal and M.L. Overton, "*Projected Hessian updating algorithms for nonlinearly constrained optimization*," *SIAM J. Numer. Anal.*, 22, pp. 821–850, 1985.
18. Y. Yuan, "*An only 2-step q-superlinear convergence example for some algorithms that use reduced Hessian approximations*," *Math. Programming*, 32, pp. 224–231, 1985.
19. J. Zhang and D. Zhu, "*A trust region type dogleg method for nonlinear optimization*," *Optimization*, 21, pp. 543–557, 1990.