Derivative Computations for a Class of Optimal Control Problems *

Luís N. Vicente †

Abstract

This paper addresses the computation of first and second order derivatives for a class of optimal control problems by the sensitivity and adjoint equation methods. The issues considered are the relationships between the derivative structure of the full and the reduced formulations and the properties of the null-space basis operator associated with the linearized state equation.

Keywords. optimal control, nonlinear optimization, adjoints, sensitivities

AMS subject classifications. 49M37, 90C06, 90C30

1 Introduction

In this paper we analyze the derivative structure of problems of the form

$$\begin{array}{ll} \text{minimize} & f(y,u) \\ \text{subject to} & c(y,u) = 0, \end{array} \tag{1}$$

arising in optimal control. Here u represents the control, y represents the state, and $c(y,u) = 0$ represents the state equation. Often, y and u belong to a function space such as the Sobolev space H^1 or the space L^2, and the state equation is a differential equation in y. Examples of optimal control problems of the form (1) are given, e.g., in [2], [4], [5], [8], [9], [10], [11], [12], possibly with the inclusion of bound constraints of the form:

$$\underline{y} \le y \le \overline{y}, \quad \underline{u} \le u \le \overline{u}. \tag{2}$$

We assume that the nonlinear optimization problem (1) is finite dimensional and we use a Hilbert space structure for its description. The first assumption is natural in the context of numerical solutions of optimal control problems. The second assumption is made because a Hilbert space structure is at least implicitly used in many optimization algorithms using iterative solvers. For this purpose let $\mathcal{Y}$, $\mathcal{U}$, and Λ be finite dimensional Hilbert spaces of dimension n_y, n_u, and n_y, respectively. These Hilbert spaces can be identified with $\mathbb{R}^{n_y}$, $\mathbb{R}^{n_u}$, and $\mathbb{R}^{n_y}$, respectively, but are equipped with scalar products $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$, $\langle \cdot, \cdot \rangle_{\mathcal{U}}$, and $\langle \cdot, \cdot \rangle_{\Lambda}$. In the context of this paper, the functions

$$\begin{array}{rcl} f : \mathcal{Y} \times \mathcal{U} & \to & \mathbb{R}, \\ c : \mathcal{Y} \times \mathcal{U} & \to & \Lambda, \end{array}$$

*Written as part of the teaching material used to support the lecture *Optimization Methods for Control and Design Problems* given by the author at the *School of Finite Elements and Applications*, CIM, Coimbra 1998. This paper is motivated from the approaches given in the papers [3] and [7].

†Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra, Portugal. E-Mail: lvicente@mat.uc.pt.
Support for this work has been provided by Centro de Matemática da Universidade de Coimbra, FCT, and Praxis XXI 2/2.1/MAT/346/94.

are assumed to be once or twice differentiable, depending on the context.

In many cases, the optimal control problem is not posed in the form (1), but the state equation $c(y,u) = 0$ is used to define y as a function of u with the aid of the implicit function theorem. This procedure eliminates the state variables y and the state constraint $c(y,u) = 0$. The resulting problem is given by

$$\text{minimize} \quad \widehat{f}(u) = f(y(u), u), \tag{3}$$

with the addition of the constraints $\underline{y} \le y(u) \le \overline{y}$ and $\underline{u} \le u \le \overline{u}$ if the bounds (2) are included in (1). Obviously, the two problems (1) and (3) are related, but they are not necessarily equivalent. If for given u the equation $c(y,u) = 0$ has more than one solution, the implicit function theorem will select one solution branch $y(u)$, provided the assumptions of the implicit function theorem are satisfied.

The discretized problem inherits structure from the infinite dimensional problem that is not easily captured by techniques included in nonlinear optimization codes. Besides the obvious splitting of the optimization variables into y and u, there is also a scaling associated with these variables that is derived from the infinite dimensional problem and its discretization [2], [8]. The scalar products $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$, $\langle \cdot, \cdot \rangle_{\mathcal{U}}$, and $\langle \cdot, \cdot \rangle_{\Lambda}$ introduced above are one way of taking into account the scaling of the problem [7], [8]: they can be used for instance to compute the discretization of a H^1 scalar product by means of a stiffness matrix. Another sensitive issue is that much of the problem information is not available explicitly, but can only be accessed indirectly. For example, the derivative or partial derivatives of c may not be available in matrix form but only the result of a derivative-times-vector operation may be accessible. This is, e.g, the case if $c(y,u) = 0$ corresponds to a partial differential equation discretized with a finite element method. In this case it is often not necessary to assemble the finite element matrices, but only to store the contributions from individual elements in the FEM mesh. This alternative allows one to compute matrix-vector multiplications without forming the matrix explicitly.

Typically, sensitivity and adjoint equation methods are used to compute the gradient and second-order derivative information for $\widehat{f}$. However, the same issues also arise for certain first and second order derivative computations related to the problem (1). The main purpose of this paper is to describe the sensitivity and adjoint equation methods for (1) and (3) and to establish a common framework that can be used in many optimization algorithms for these problems and takes into account the issues raised in the previous paragraph. (For more discussions on sensitivity and adjoint equation approaches see, e.g., [1].) The properties of the null-space basis operator associated with the linearized state equation (that plays an important role in optimization algorithms) are studied in detail.

2 Derivative computations: Adjoints and sensitivities

We consider the Lagrangian associated with problem (1)

$$\ell(y, u, \lambda) = f(y, u) + \langle \lambda, c(y, u) \rangle_{\Lambda}. \tag{4}$$

In Sections 2.1 and 2.2 we use the sensitivity and adjoint equation methods to compute the gradient and second-order derivative information for $\widehat{f}$ and ℓ. We use the scalar products framework and focus on computational aspects related with these derivative calculations.

The fact that $\widehat{f}$ and f are objective functions is not important. It is only important that $\widehat{f} : \mathcal{U} \to I\!R$ depends on the implicit function $y(u)$. In general the sensitivity and adjoint equation methods are needed when derivative information of a function, say, $\widehat{h} : \mathcal{U} \to I\!R$ is computed that is a composition of a function h and $y(\cdot)$. Thus many of the derivations below also apply in this context.

2.1 First-order derivatives

Under the assumptions of the implicit function theorem (which will be assumed throughout this paper) the derivative of the implicitly defined function $y(\cdot)$ is given as the solution of

$$c_y(y(u), u)y'(u) = -c_u(y(u), u). \tag{5}$$

This equation is called the *sensitivity equation* and its solution is called the sensitivity of y. We can now compute the gradient of $\widehat{f}$:

$$\begin{aligned}
\langle \nabla \widehat{f}(u), v\rangle_{\mathcal{U}} &= \langle \nabla_y f(y(u), u), y'(u)v\rangle_{\mathcal{Y}} + \langle \nabla_u f(y(u), u), v\rangle_{\mathcal{U}} \\
&= \langle \nabla_y f(y(u), u), -c_y(y(u), u)^{-1} c_u(y(u), u)v\rangle_{\mathcal{Y}} + \langle \nabla_u f(y(u), u), v\rangle_{\mathcal{U}} \\
&= \langle -(c_y(y(u), u)^{-1} c_u(y(u), u))^* \nabla_y f(y(u), u) + \nabla_u f(y(u), u), v\rangle_{\mathcal{U}}.
\end{aligned}$$

Hence,

$$\nabla \widehat{f}(u) = -\Big(c_y(y(u), u)^{-1} c_u(y(u), u)\Big)^* \nabla_y f(y(u), u) + \nabla_u f(y(u), u). \tag{6}$$

The formula (6) is used in *the sensitivity equation method* to compute the gradient. First, the sensitivity matrix

$$S(y(u), u) = c_y(y(u), u)^{-1} c_u(y(u), u)$$

is computed and then the gradient is formed using (6):

$$\widehat{f}(u) = -S(y(u), u)^* \nabla_y f(y(u), u) + \nabla_u f(y(u), u).$$

To introduce the adjoint equation approach, we rewrite the formula (6) for the gradient as follows:

$$\nabla \widehat{f}(u) = -c_u(y(u), u)^* (c_y(y(u), u)^*)^{-1} \nabla_y f(y(u), u) + \nabla_u f(y(u), u).$$

Thus one can compute the adjoint variables $\lambda(u)$ by solving the adjoint equation

$$c_y(y(u), u)^* \lambda(u) = -\nabla_y f(y(u), u)$$

and then compute the gradient using

$$\nabla \widehat{f}(u) = c_u(y(u), u)^* \lambda(u) + \nabla_u f(y(u), u).$$

This calculation is *the adjoint equation method* to compute the gradient.

Traditionally, the sensitivity equation method and the adjoint equation method have been used in the context of the reduced problem (3). However, the same techniques are also needed to compute derivative information for the solution of (1). Consider for this purpose the partial gradients of the Lagrangian (4):

$$\nabla_y \ell(y, u, \lambda) = \nabla_y f(y, u) + c_y(y, u)^* \lambda, \qquad \nabla_u \ell(y, u, \lambda) = \nabla_u f(y, u) + c_u(y, u)^* \lambda.$$

We see that $\nabla_y \ell(y, u, \lambda) = 0$ corresponds to the adjoint equation

$$c_y(y, u)^* \lambda = -\nabla_y f(y, u). \tag{7}$$

If we define $\lambda(y,u)$ as the solution of (7), then

$$\nabla_u \ell(y,u,\lambda)|_{\lambda=\lambda(y,u)} = \nabla_u f(y,u) - c_u(y,u)^*(c_y(y,u)^*)^{-1}\nabla_y f(y,u).$$

In particular,

$$\nabla \widehat{f}(u) = \nabla_u \ell(y,u,\lambda)|_{y=y(u),\lambda=\lambda(u)}. \tag{8}$$

With

$$W(y,u) = \begin{pmatrix} -c_y(y,u)^{-1}c_u(y,u) \\ I_{n_u} \end{pmatrix} \tag{9}$$

we can write

$$\nabla_u \ell(y,u,\lambda)|_{\lambda=\lambda(y,u)} = W(y,u)^* \begin{pmatrix} \nabla_y f(y,u) \\ \nabla_u f(y,u) \end{pmatrix}$$

and

$$\nabla \widehat{f}(u) = W(y,u)^* \begin{pmatrix} \nabla_y f(y,u) \\ \nabla_u f(y,u) \end{pmatrix}\Big|_{y=y(u)}.$$

An optimization algorithm applied to the solution of (1) may require the evaluation of the Lagrangian $f(y,u) + \langle \lambda(y,u), c(y,u)\rangle_\Lambda$, where $\lambda(y,u)$ is the solution of (7). If the adjoint equation method is used for the derivatives, the adjoint variables $\lambda(y,u)$ can be calculated. If only the sensitivities $S(y,u) = c_y(y,u)^{-1}c_u(y,u)$ and their adjoints are provided, adjoint variables cannot be computed from (7). In such a situation we can evaluate the corresponding value of the Lagrangian by solving the *linearized state equation*

$$c_y(y,u)s \;=\; -c(y,u) \tag{10}$$

and by using the relation

$$\begin{aligned} \langle \lambda(y,u), c(y,u)\rangle_\Lambda &= -\langle (c_y(y,u)^*)^{-1}\nabla_y f(y,u), c(y,u)\rangle_\Lambda \\ &= -\langle \nabla_y f(y,u), c_y(y,u)^{-1}c(y,u)\rangle_{\mathcal{Y}}\,. \end{aligned}$$

2.2 Second-order derivatives

The issue of sensitivities and adjoints not only arise in gradient calculations, but also in Hessian computations. The Hessian of the Lagrangian

$$\nabla^2_{xx}\ell(y,u,\lambda) = \begin{pmatrix} \nabla^2_{yy}\ell(y,u,\lambda) & \nabla^2_{yu}\ell(y,u,\lambda) \\ \nabla^2_{uy}\ell(y,u,\lambda) & \nabla^2_{uu}\ell(y,u,\lambda) \end{pmatrix} \tag{11}$$

and the reduced Hessian

$$\widehat{H}(y,u) = W(y,u)^* \begin{pmatrix} \nabla^2_{yy}\ell(y,u,\lambda) & \nabla^2_{yu}\ell(y,u,\lambda) \\ \nabla^2_{uy}\ell(y,u,\lambda) & \nabla^2_{uu}\ell(y,u,\lambda) \end{pmatrix} W(y,u)\Big|_{\lambda=\lambda(y,u)} \tag{12}$$

play an important role in the development of optimization methods. Moreover, one can show that the Hessian of the reduced functional in (3) is given by

$$\nabla^2 \widehat{f}(u) = \widehat{H}(y(u),u). \tag{13}$$

The proof is the following [6]: Since, as we have seen in (8), $\nabla \widehat{f}(u) = \nabla_u \ell(y(u),u,\lambda(u))$, we have

$$\nabla^2 \widehat{f}(u) = \nabla^2_{uy}\ell(y(u),u,\lambda(u))y'(u) + \nabla^2_{uu}\ell(y(u),u,\lambda(u)) + \nabla^2_{u\lambda}\ell(y(u),u,\lambda(u))\lambda'(u).$$

The derivative $\lambda'(u)$ can be obtained by differentiating

$$\nabla_y \ell(y(u), u, \lambda(u)) = 0,$$

which gives

$$\nabla^2_{yy}\ell(y(u), u, \lambda(u))y'(u) + \nabla^2_{yu}\ell(y(u), u, \lambda(u)) + \nabla^2_{y\lambda}\ell(y(u), u, \lambda(u))\lambda'(u) = 0.$$

Thus, since $\nabla^2_{y\lambda}\ell(y(u), u, \lambda(u)) = c_y(y(u), u)^*$, we get

$$\lambda'(u) = (c_y(y(u), u)^*)^{-1}\left(\nabla^2_{yy}\ell(y(u), u, \lambda(u))c_y(y(u), u)^{-1}c_u(y(u), u) - \nabla^2_{yu}\ell(y(u), u, \lambda(u))\right).$$

This expression for $\lambda'(u)$ and the fact that $\nabla^2_{u\lambda}\ell(y(u), u, \lambda(u)) = c_u(y(u), u)^*$ complete the proof of (13).

We note that the computation of (11) and (12) requires knowledge of the adjoint variables λ. In many algorithms, these are computed via the adjoint equations (7). If only the sensitivities $S(y, u) = c_y(y, u)^{-1}c_u(y, u)$ and their adjoints are provided, adjoint variables cannot be computed from (7). If no estimate for λ is available, then the operators in (11) and (12) cannot be computed. In cases in which $\nabla_y f(y, u) \approx 0$ for (y, u) near the solution, one may set $\lambda = \lambda(y, u) \approx 0$, cf. (7). This leads to the approximations

$$\nabla^2_{xx}\ell(y, u, \lambda) \approx \begin{pmatrix} \nabla^2_{yy}f(y, u) & \nabla^2_{yu}f(y, u) \\ \nabla^2_{uy}f(y, u) & \nabla^2_{uu}f(y, u) \end{pmatrix}$$

and

$$\widehat{H}(y, u) \approx W(y, u)^* \begin{pmatrix} \nabla^2_{yy}f(y, u) & \nabla^2_{yu}f(y, u) \\ \nabla^2_{uy}f(y, u) & \nabla^2_{uu}f(y, u) \end{pmatrix} W(y, u). \tag{14}$$

The situation $\nabla_y f(y, u) \approx 0$ often arises in least squares functionals $f(y, u) = \frac{1}{2}\|y - y_d\|^2_{\mathcal{Y}} + \frac{\gamma}{2}\|u\|^2_{\mathcal{U}}$, where y_d is some desired state. In this case $\nabla_y f(y, u) = y - y_d$ and if the given data y_d can be fitted well, then $\nabla_y f(y, u) \approx 0$. If $y = y(u)$, the approximation (14) is the Gauss-Newton approximation to the Hessian $\nabla^2\widehat{f}(u)$ (see the derivation in the next paragraph).

The Hessian $\nabla^2\widehat{f}(u)$ of the reduced objective can also be computed by using second-order sensitivities. In this approach one applies the chain rule to $\nabla\widehat{f}(u) = y'(u)^*\nabla_y f(y(u), u) + \nabla_u f(y(u), u)$ and get

$$\begin{aligned} \nabla^2\widehat{f}(u) &= y''(u)^*\nabla_y f(y(u), u) + y'(u)^*\left(\nabla^2_{yy}f(y(u), u)y'(u) + \nabla^2_{yu}f(y(u), u)\right) \\ &\quad + \nabla^2_{uy}f(y(u), u)y'(u) + \nabla^2_{uu}f(y(u), u), \end{aligned}$$

where

$$y''(u)^*\nabla_y f(y(u), u) = \sum_{i=1}^{n_y}(\nabla_y f(y(u), u))_i\nabla^2 y_i(u).$$

Thus,

$$\nabla^2\widehat{f}(u) = \sum_{i=1}^{n_y}(\nabla_y f(y(u), u))_i\nabla^2 y_i(u) + W(y(u), u)^* \begin{pmatrix} \nabla^2_{yy}f(y(u), u) & \nabla^2_{yu}f(y(u), u) \\ \nabla^2_{uy}f(y(u), u) & \nabla^2_{uu}f(y(u), u) \end{pmatrix} W(y(u), u),$$

where the second-order derivatives of $y(u)$ can be obtained by applying the implicit function theorem to (5). Unlike (11) and (12), this approach avoids the explicit use of Lagrange multipliers.

2.3 The operator $W(y,u)$

The introduction of $W(y,u)$ which plays an important role in optimization methods for (7) allowed an elegant and compact notation for the first-order derivatives and the second-order derivatives. It also localizes the use of the sensitivity equation method and the adjoint equation method in the derivative calculations. In all derivative computations, the sensitivity equation method or the adjoint equation method is only needed to evaluate the application of $W(y,u)$ and $W(y,u)^*$ onto vectors. For example, the computation of the y-component z_y of $z = W(y,u)d_u$ is done in two steps:

$$\begin{array}{lrcl} \text{Compute} & v_y & = & -c_u(y,u)d_u. \\ \text{Solve} & c_y(y,u)z_y & = & v_y. \end{array}$$

If the sensitivities $S(y,u) = c_y(y,u)^{-1}c_u(y,u)$ are known, then $z_y = -S(y,u)d_u$. The equation $c_y(y,u)z_y = v_y$ is a generalized linearized state equation, cf. (10). Similarly, for given d the matrix-vector product $z = W(y,u)^*d$, $d = (d_y, d_u)$, is computed successively as follows:

$$\begin{array}{lrcl} \text{Solve} & c_y(y,u)^*v_y & = & -d_y. \\ \text{Compute} & v_u & = & c_u(y,u)^*v_y. \\ \text{Compute} & z & = & v_u + d_u. \end{array}$$

Again, if the adjoint of the sensitivities $S(y,u) = c_y(y,u)^{-1}c_u(y,u)$ are known, then $z = -S(y,u)^*d_y + d_u$. The equation $c_y(y,u)^*v_y = -d_y$ is a generalized adjoint equation, cf. (7).

A number of optimization methods for (1) or (3) require the computation of some of the quantities

$$H(y,u,\lambda)s, \quad \langle s, H(y,u,\lambda)s\rangle_{\mathcal{X}}, \quad W(y,u)^*H(y,u,\lambda)s,$$
$$W(y,u)^*H(y,u,\lambda)W(y,u)s_u, \quad \langle s_u, W(y,u)^*H(y,u,\lambda)W(y,u)s_u\rangle_{\mathcal{U}}$$

for given $s = (s_y, s_u)$ and s_u, where $H(y,u,\lambda)$ is the Hessian $\nabla^2_{xx}\ell(y,u,\lambda)$ or an approximation thereof. Often, one does not approximate the Hessian $\nabla^2_{xx}\ell(y,u,\lambda)$, but the reduced Hessian. If $\widehat{H}(y,u) \approx W(y,u)^*\nabla^2_{xx}\ell(y,u,\lambda)W(y,u)$, then this approximation fits into the previous framework in which the full Hessian is approximated by setting

$$H(y,u,\lambda) = \begin{pmatrix} 0 & 0 \\ 0 & \widehat{H}(y,u) \end{pmatrix}. \tag{15}$$

If $H(y,u,\lambda)$ is given by (15), then the definition of $W(y,u)$ implies the equalities

$$H(y,u,\lambda)s = \begin{pmatrix} 0 \\ \widehat{H}(y,u)s_u \end{pmatrix},$$
$$\langle s, H(y,u,\lambda)s\rangle_{\mathcal{X}} = \langle s_u, \widehat{H}(y,u)s_u\rangle_{\mathcal{U}}, \quad W(y,u)^*H(y,u,\lambda)s = \widehat{H}(y,u)s_u,$$
$$W(y,u)^*H(y,u,\lambda)W(y,u)s_u = \widehat{H}(y,u)s_u,$$
$$\langle s_u, W(y,u)^*H(y,u,\lambda)W(y,u)s_u\rangle_{\mathcal{U}} = \langle s_u, \widehat{H}(y,u)s_u\rangle_{\mathcal{U}}.$$

Finally, a study of the projector associated with the operator $W(y,u)$ allows a better understanding of the reduced gradient $W(y,u)^*\nabla f(y,u)$. In fact, the operator $W(y,u)$ defines an oblique projector onto $\mathcal{N}(J(y,u))$:

$$P_{obl}(y,u) = W(y,u)W(y,u)^*,$$

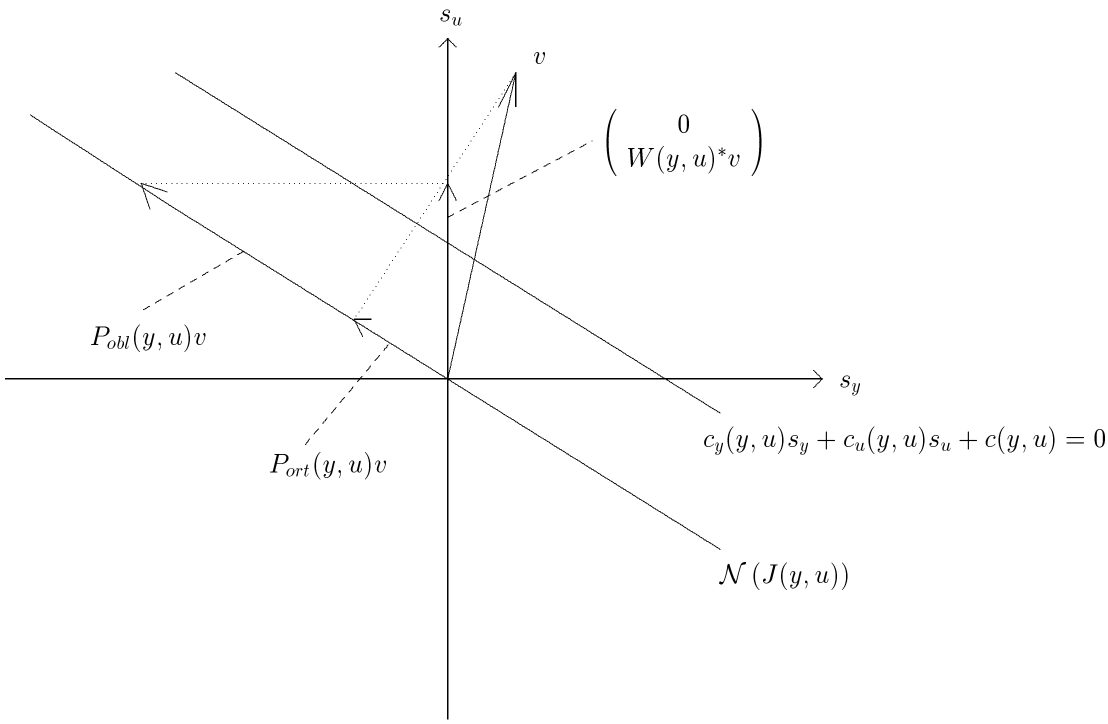

Figure 1: The action of the orthogonal and oblique projectors.

where $\mathcal{N}(J(y,u))$ is the null space of $J(y,u) = \begin{pmatrix} c_y(y,u) & c_u(y,u) \end{pmatrix}$. Also, it can be easily proved that

$$P_{ort}(y,u) = W(y,u)\left(W(y,u)^* W(y,u)\right)^{-1} W(y,u)^* \tag{16}$$

provides an orthogonal projector onto $\mathcal{N}(J(y,u))$. In Figure 1 we depict the action of the projectors $P_{obl}(y,u)$ and $P_{ort}(y,u)$ on a given vector v. The following proposition [13] provides an explanation for the geometric form of $P_{obl}(y,u)v$.

Proposition 2.1 *Given a vector v in $\mathcal{Y} \times \mathcal{U}$,*

$$P_{ort}(y,u)v = P_{ort}(y,u) \begin{pmatrix} 0 \\ W(y,u)^* v \end{pmatrix}. \tag{17}$$

In addition, $\begin{pmatrix} 0 \\ W(y,u)^* v \end{pmatrix}$ is the unique vector in the vector space $\{(y,u) \in \mathcal{Y} \times \mathcal{U} : \; y = 0\}$ for which (17) holds.

Proof: The proof of the first part is the following:

$$\begin{aligned}
& P_{ort}(y,u) \begin{pmatrix} 0 \\ W(y,u)^* v \end{pmatrix} \\
= \;& W(y,u)\left(W(y,u)^* W(y,u)\right)^{-1} \begin{pmatrix} -c_y(y,u)^{-1} c_u(y,u) \\ I_{n_u} \end{pmatrix}^* \begin{pmatrix} 0 \\ W(y,u)^* v \end{pmatrix} \\
= \;& W(y,u)\left(W(y,u)^* W(y,u)\right)^{-1} W(y,u)^* v \\
= \;& P_{ort}(y,u)v,
\end{aligned}$$

where we used (9) and the form of $P_{ort}(y,u)$ given in (16). To prove the uniqueness suppose that $x_1 = (0\ u_1)$ and $x_2 = (0\ u_2)$ satisfy $P_{ort}(y,u)x_1 = P_{ort}(y,u)x_2$. From $P_{ort}(y,u)(x_1 - x_2) = 0$ we conclude that $x_1 - x_2$ is orthogonal to $\mathcal{N}(J(y,u))$, i.e., $W(y,u)^*(x_1 - x_2) = 0$. But this is just

$$\begin{pmatrix} -c_y(y,u)^{-1}c_u(y,u) \\ I_{n_u} \end{pmatrix}^* \begin{pmatrix} 0 \\ u_1 - u_2 \end{pmatrix} = 0$$

and $u_1 = u_2$. □

From this proposition we know how to depict $W(y,u)^*v$ along the u axis. Note that

$$P_{obl}(y,u)v = W(y,u)W(y,u)^*v = \begin{pmatrix} -c_y(y,u)^{-1}c_u(y,u)\,W(y,u)^*v \\ W(y,u)^*v \end{pmatrix}$$

lies in the null space $\mathcal{N}(J(y,u))$ and is also depicted in Figure 1.

References

[1] J. Borggaard, J. Burns, E. Cliff, and S. Schreck, editors. *Computational Methods for Optimal Design. Proceedings of the ALSOR Workshop on Optimal Design and Control, Arlington, VA, 30-September – 3-October 1997*, Progress in Systems and Control Theory, Basel, Boston, Berlin, 1998. Birkhäuser Verlag. `http://www.icam.vt.edu/workshop/proceedings.html`.

[2] E. M. Cliff, M. Heinkenschloss, and A. Shenoy. An optimal control problem for flows with discontinuities. *J. Optim. Theory Appl.*, 94:273–309, 1997.

[3] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente. Trust–region interior–point algorithms for a class of nonlinear programming problems. *SIAM J. Control Optim.*, 36:1750–1794, 1998.

[4] M. D. Gunzburger, L. S. Hou, and T. P. Svobotny. Optimal control and optimization of viscous, incompressible flows. In M. D. Gunzburger and R. A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics*, pages 109–150, Cambridge, New York, 1993. Cambridge University Press.

[5] N. Handagama and S. Lenhart. Optimal control of a PDE/ODE system modeling a gas-phase bioreactor. In M. A. Horn, G. Simonett, and G. Webb, editors, *Mathematical Models in Medical and Health Sciences*, Nashville, TN, 1998. Vanderbilt University Press.

[6] M. Heinkenschloss. Projected sequential quadratic programming methods. *SIAM J. Optim.*, 6:373–417, 1996.

[7] M. Heinkenschloss and L. N. Vicente. An interface between optimization and application for the numerical solution of optimal control problems. Technical Report 98-08, Departamento de Matemática, Universidade de Coimbra, 1998. Accepted for publication in ACM Trans. Math. Software.

[8] M. Heinkenschloss and L. N. Vicente. Numerical solution of semielliptic optimal control problems using SQP based optimization algorithms. Technical report, Department of Computational and Applied Mathematics, Rice University, 1998. In preparation.

[9] K. Ito and K. Kunisch. Augmented Lagrangian-SQP methods for nonlinear optimal control problems of tracking type. *SIAM J. Control and Optimization*, 34:874–891, 1996.

[10] F.-S. Kupfer and E. W. Sachs. Numerical solution of a nonlinear parabolic control problem by a reduced SQP method. *Comput. Optim. and Appl.*, 1:113–135, 1992.

[11] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations.* Springer Verlag, Berlin, Heidelberg, New York, 1971.

[12] P. Neittaanmäki and D. Tiba. *Optimal Control of Nonlinear Parabolic Systems. Theory, Algorithms, and Applications.* Marcel Dekker, New York, Basel, Hong Kong, 1994.

[13] L. N. Vicente. *Trust-Region Interior-Point Algorithms for a Class of Nonlinear Programming Problems.* PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251, USA, 1996.