A guide to the package `dsoa`

Brian C. Fabien
Department of Mechanical Engineering
University of Washington
Seattle, WA 98195
e-mail: fabien@u.washington.edu

1 Introduction

This document describes how to use the ANSI C/C++ computer package `dsoa`. The computer codes associated with this package implement a direct method for the solution of dynamic system optimization problems described by index-1 differential-algebraic equations. In particular we consider the problem;

$$\text{DSOA}: \min_{y(t_i)\in\mathcal{R}^{n_y}, u(t)\in\mathcal{R}^{n_u},\ p\in\mathcal{R}^{n_p}} J = \phi(y(t_f),p) + \int_{t_i}^{t_f} L(y(t),u(t),p)\,dt \tag{1}$$

subject to

$$M\dot{y}(t) = F(y(t),u(t),p), \quad t\in[t_i,t_f], \tag{2}$$
$$0 = \Gamma_j(y(t_i),u(t_i),p),\ j=1,2,\cdots,n_\Gamma, \tag{3}$$
$$0 \geq d_j(y(t),u(t),p),\ j=1,2,\cdots,n_d, \quad t\in[t_i,t_f], \tag{4}$$
$$0 = \Psi_j(y(t_f),u(t_f),p),\ j=1,2,\cdots,n_\Psi. \tag{5}$$

Here, $y(t)\in\mathcal{R}^{n_y}$ denotes the state vector, $u(t)\in\mathcal{R}^{n_u}$ is the control input, $p\in\mathcal{R}^{n_p}$ are system parameters.

The optimization problem DSOA seeks to find the state $y(t_i)$ at the initial time, t_i; the control, $u(t)$; and the parameters p that minimizes the scalar cost functional J. The cost functional is made up of the terminal penalty function, $\phi(t_f,p)$, and the integrand $L(y(t),u(t),p)$, which are both scalars. The initial time t_i, and the final time t_f are both fixed. The system state vector, $y(t)\in\mathcal{R}^{n_y}$, must satisfy the differential equations (2), where $F(y(t),u(t),p)\in\mathcal{R}^{n_y}$. The matrix $M\in\mathcal{R}^{n_y\times n_y}$ is constant in these systems.

If M is singular then (2) represents a system of differential-algebraic equations, otherwise if M is nonsingular then (2) represents a system of ordinary differential equations (ODEs). We assume that the differential-algebraic equations have differentiation index-1, i.e., $\text{rank}[M\ F_y(y,u,p)] = n_y$ for all $u(t)$, p and $y(t)$, where $F_y = \partial F/\partial y$.

Equation (3) indicates that the state, control and parameters must satisfy the n_Γ constraints $\Gamma_j = 0$ at the initial time. These initial constraints can be used to ensure that the initial conditions are consistent, i.e., for a given $u(t_i)$ and p there is a $\dot{y}(t_i)$ such that $M\dot{y}(t_i) - F(y(t_i), u(t_i), p) = 0$. Equation (4) indicates that the inequality constraints $d_j(y(t), u(t), p)$ $j = 1, 2, \cdots, n_d$ must be satisfied in the entire interval $t_i \leq t \leq t_f$. Finally, (5) indicates that the state and parameters must satisfy the constraints $\Psi_j(y(t_f), p) = 0$, $j = 1, 2, \cdots, n_\Psi$, at the final time.

It is assumed that the functions ϕ, L, F, d, Γ, and Ψ are continuously differentiable with respect to their arguments.

Note that the ODEs/DAEs (2) are autonomous; however, nonautonomous systems can be easily accommodated by adding a new state variable to represent the time as shown in Example 2 below. It should also be noted that minimum time problems can also be written in the form (1)–(5) (see [1] pp. 106).

The numerical methods implemented in the `dsoa` package are described in the papers [30, 31, 32, 33]. In section 3 we will present a simplified approach to using the package `dsoa` for solving nonlinear optimal control problems. Section 5 contains a description of 83 test problems that have appeared in the literature. This section also evaluates the performance of the `dsao` package when it is used to solve these test problems.

2 Installation

You can install the `dsoa` package using the following steps.

1. First retrieve the archive `dsoa.zip` from the web site `http://abs-5.me.washington.edu/dsoa/dsoa.zip`

2. On Unix based systems uncompress the archive in a command shell using:

   ```
   unzip dsoa.zip
   ```

   This command will create the following directories;

   ```
   dsoa/doc
   dsoa/include
   dsoa/lib
   dsoa/lib_ad
   dsoa/src
   dsoa/test_fd
   dsoa/test_ad
   ```

   The directory `dsoa/doc` contains documentation related to the package. The directory `dsoa/include` contains the C/C++ header files, as well as the template files used by the solver. The directory `dsoa/lib` stores the C object code for the functions associated with the package. This library is used with the finite difference computations. The directory `dsoa/lib_ad` stores the C++ object code for the functions associated with the package. This library is used with the automatic differentiation computations. The directory `dsoa/src` contains the source code for the package. The directory `dsoa/test_fd` contains the test problems used with the finite difference computations. The directory `dsoa/test_ad` contains the test problems used with the automatic differentiation computations.

3. The object code can be created using the commands:

   ```
   cd lib
   cc -c -O -I ../include ../src/*.c
   ar -rcs libdsoafd.a *.o
   cd ../lib_ad
   c++ -c -O ../include ../src/*.c
   c++ -c -O ../include ../src/*.C
   ar -rcs libdsoaad.a *.o
   ```

   If you are using the GNU C/C++ compilers then use `-O3` instead of `-O` for improved performance. If you are using the Sun C++ compiler then use `CC` instead of `c++`.

4. To execute the test problem `ex1` using finite difference gradient evaluations then issue the following commands:

   ```
   cd test_fd
   dsoa ex1
   ```

   (Note you may need to `chmod +x dsoa` for this to work.) Here, `dsoa` is a script that combines the file `ex1` with the template file `../include/_dsoa_function_` to construct a C program that uses the `dsoa` solver. The script compiles the program, links it with the object code in `lib/`, and then executes the program.

   To solve the test problems using automatic differentiation we must execute the `dsoa` command in the `test_ad` directory. In this case the `dsoa` script links the input file with C++ code in the library `lib_ad`.

3 The user interface to `dsoa`

Using the package `dsoa` to solve nonlinear optimal control problems is in fact quite simple. This is demonstrated in section 5 where we consider the solution of numerous optimal control problems that have appeared in the literature. The ease by which the package can be used is primarily due to a simplified user interface to the numerical solution methods. The user interface is template driven, and takes advantage of the fact that the problem DSOA is highly structured.

To solve an optimal control problem using the `dsoa` package one creates an ASCII file that describes the problem in the form of DSOA. The input file is in fact ANSI C code that describes the problem DSOA. In order to simplify the input certain variables are predefined, and the problem equations are entered in specified blocks.

The input file is divided into three main sections, namely; (i) Declaration of user defined variables, if any; (ii) Assignment of the problem dimensions and fixed parameters; and (iii) Statements that define the equations in the problem.

The user defined variables can be any valid C type; however, the variable names can not be one of the reserved words listed in Table 1, Table 2 and Table 3. Table 1 lists the predefined variables for the problem dimensions and fixed parameters. This table also shows the C type for the variables, and their default values. Table 2 shows the predefined names for the system variables and their C type. Finally, Table 3 shows the names for the various code blocks that are used to evaluate the expressions that define the problem DSOA.

`n_states`	$= n_y$, the number of state variables (`int`). Default `n_states=0`.
`n_controls`	$= n_u$, the number of control variables (`int`). Default `n_controls=0`.
`n_parameters`	$= n_p$, the number of parameter variables (`int`). Default `n_parameters=0`.
`n_initial`	$= n_\Gamma$, the number of initial condition constraints (`int`). Default `n_initial=0`.
`n_terminal`	$= n_\Psi$, the number of terminal condition constraints (`int`). Default `n_terminal=0`.
`n_inequality`	$= n_d$, the number of inequality constraints (`int`). Default `n_inequality=0`.
`n_nodes`	$= N$, the number of time nodes (`int`). Default `N=0`.
`initial_time`	$= t_i$, the initial time (`double`). Default `initial_time=0.0`.
`final_time`	$= t_f$, the initial time (`double`). Default `final_time=1.0`.
`tolerance`	$= \epsilon$, the convergence tolerance (`double`). Default `tolerance=1.0e-4`.
`input_file`	= the name of the file containing the input data (`char *`).
`output_file`	= the name of the file containing the output data (`char *`). Default `dsoa_output.data`.

Table 1: Problem dimensions and fixed parameters

`y`	$= y$, the states (`Matrix`).
`u`	$= u$, the controls (`Matrix`).
`p`	$= p$, the parameters (`Matrix`).
`phi`	$= \phi$, the terminal penalty term (`double`).
`L`	$= L$, the integrand of the cost functional (`double`).
`M`	$= M$, the coefficient matrix in the ODEs/DAEs (`Matrix`).
`F`	$= F$, the equations that define the ODEs/DAEs (`Matrix`).
`d`	$= d$, the equations that define the inequality constraints (`Matrix`).
`Gamma`	$= \Gamma$, the equations that define the initial condition constraints (`Matrix`).
`Psi`	$= \Psi$, the equations that define the terminal condition constraints (`Matrix`).
`y0`	an estimate of the state vector at the initial time, $y(t_i)$ (`Matrix`).
`u0`	an estimate of the optimal control history (`Matrix`).
`p0`	an estimate of the optimal parameters (`Matrix`).

Table 2: Problem variables

`set_M`	Initialize M.
`initial_constraint`	Evaluate Γ.
`dynamic_equation_and_cost_function`	Evaluate F and L.
`inequality_constraint`	Evaluate d.
`terminal_condition`	Evaluate ϕ and Ψ.
`solution_estimate`	Assign estimates for $y(t_i)$, $\bar{u}$ and p.

Table 3: Problem code blocks

3.1 Examples

The construction of input files for the `dsoa` package is best illustrated via examples. To that end we describe in detail the input files used to solve three nonlinear optimal control problems.

Example 1 Consider the problem;

$$\min_{u(t)} J = -y_1(1)$$

subject to

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= u \\ 0 &= y_1(0) \\ 0 &= y_2(0) \\ 0 &= y_2(1) \\ 1 &\geq |u(t)| \end{aligned}$$

Here we wish to find the bounded control $u(t)$ that maximizes the distance traveled, i.e., $y_1(t)$. (Hence, the variable $y_2(t)$ represents the velocity, and the control $u(t)$ is a force.) In terms of the problem DSOA we see that $t_i = 0$, $t_f = 1$, $\phi = -y_1(1)$, $L = 0$, $M = \text{diag}([1,\ 1])$, $f = [y_2,\ u]^T$, $d = [u - 1,\ -1 - u]^T$, $\Gamma = [y_1(0),\ y_2(0)]^T$, $\Psi = [y_2(1)]$.

A `dsoa` input file that describes this problem is shown in listing below.

```
n_states = 2;
n_controls = 1;
n_initial = 2;
n_inequality = 2;
n_terminal = 1;
n_nodes = 51;
initial_time = 0.0;
final_time = 1.0;
tolerance = 1.0e-4;
output_file = "ex4.data";

```

```
12 initial_constraint {
13         Gamma(1) = y(1);
14         Gamma(2) = y(2);
15 }
16
17 dynamic_equation_and_cost_function {
18         f(1) = y(2);
19         f(2) = u(1);
20 }
21
22 inequality_constraint {
23         d(1) = u(1)-1.0;
24         d(2) = -1.0-u(1);
25 }
26
27 terminal_condition {
28         phi = -y(1);
29         Psi(1) = y(2);
30 }
```

This ASCII file is stored with the name `ex4`.

Hopefully, the reader will see that this `dsoa` input file listing is a direct representation of the optimal control problem. Lines 1 through 10 establishes the values of the problem dimensions and fixed variables. The code block on lines 12 through 15 is used to evaluate the initial condition constraints, Γ. The code block on lines 17 through 20 is used to evaluate the right hand side of the ODEs, F. The code block on lines 22 through 25 is used to evaluate the inequality constraints, d. Finally, the code block on lines 27 through 30 is used to evaluate the terminal conditions, ϕ and Ψ.

We note the following:

- It should be clear that `y(i)` is the i-th state variable $y_i(t)$, etc.
- The expressions in each code block must be enclosed in braces { }.
- By default the matrix M is set equal to the identity matrix. So, there is no need to have a code block `set_M` in this case.
- By default the estimate of the states at the initial time is zero. Hence, there is no need for a code block `initial_state`.

To obtain the numerical solution to the problem we issue the command

```
dsoa ex4
```

in a Unix shell. This command performs the following operations;

1. The file `ex4` is combined with additional code in the template file `include/_dsoa_function_` to form a C program. This additional code contains macros that convert the code block names into conditional statements, and makes calls to the appropriate functions in the `dsoa` package.

2. The C program is compiled and linked with the `dsoa` library of functions.

3. The program is then executed. On successful completion the solution is stored in the file `ex4.data`. (The content of the data file `ex4.data` is described below.)

Some of the output generated by the command `dsoa ex4` is as follows.

```
iter = 1, f = 0.000e+00, |hg| = 0.000e+00, |DL| = 1.403e+00
iter = 2, f = -1.666e-03, |hg| = 4.066e-19, |DL| = 4.082e-02
iter = 3, f = -9.996e-03, |hg| = 2.168e-19, |DL| = 4.082e-02
.
.
.
iter = 56, f = -2.497e-01, |hg| = 6.661e-16, |DL| = 2.981e-04
iter = 57, f = -2.498e-01, |hg| = 6.661e-16, |DL| = 2.981e-04
iter = 58, f = -2.500e-01, |hg| = 4.441e-16, |DL| = 2.117e-04

RSQP_solve: |d| <= SQRT_DBL_EPSILON*(1+|x|)

$ f     = -0.25
$ |hg|  = 4.44089e-16
$ |DL|  = 1.00218e-14
$ #f    = 58
$ #grad = 58
$ #soc  = 0
$ #xqp  = 0
$ n     = 52
$ m     = 3
$ p     = 102
$ exit  = 0
$ time  = 1 seconds
```

In this output `iter` is the iteration counter for the NLP solver; `f` is the NLP function value at the current iterate; `|hg|` is the infinity norm of the equality and inequality constraints of the NLP problem, `|DL|` is the norm of the gradient of the Lagrangian associated with the NLP problem.

On termination of the NLP algorithm, the solver also reports; `#f` the number of NLP function evaluations performed, `#grad` the number of NLP gradient evaluations performed, `#soc` the number of second-order corrections performed, `#xqp` the number of relaxed QP problems solved, `n` the number of unknowns in the NLP problem, `m` the number of equality constraints, `p` the number of inequality constraints, `exit` the exit code for the solver (see [30, 31]), and `time` the time required to solve the NLP problem.

Example 2 Consider the problem [44],

$$\min_{u(t)\in\mathcal{R}} \int_0^1 y_1^2 + y_2^2 + 0.005u^2 \, dt$$

subject to

$$\begin{aligned}
\dot{y}_1 &= y_2, \\
\dot{y}_2 &= -y_2 + u, \\
\dot{y}_3 &= 1, \\
y_1(0) &= 0, \\
y_2(0) &= -1, \\
y_3(0) &= 0, \\
0 &\geq u - 20, \\
0 &\geq -20 - u, \\
0 &\geq -(8(y_3 - 0.5)^2 - 0.5 - y_2).
\end{aligned}$$

The `dsoa` script used to describe this problem is shown in the listing below.

```
n_states     = 3;
n_controls   = 1;
n_initial    = 3;
n_inequality = 3;
n_nodes      = 101;
initial_time = 0.0;
final_time   = 1.0;
output_file  = "ex6.data";

initial_constraint {
        Gamma(1) = y(1);
        Gamma(2) = y(2)+1.0;
        Gamma(3) = y(3);
}

dynamic_equation_and_cost_function {
        F(1) = y(2);
        F(2) = -y(2) + u(1);
        F(3) = 1.0;
        L = y(1)*y(1) + y(2)*y(2) + 0.005*u(1)*u(1);
}

inequality_constraint {
        d(1) = u(1)-20.0;
        d(2) =-20.0-u(1);
        d(3) =-(8.0*(y(3)-0.5)*(y(3)-0.5)-0.5-y(2));
}
```

```

solution_estimate {
        y0(2) = -1.0;
        /* All other variables set to zero by default */
}
```

The statements in this script are self documenting, however, we note the following. (i) The control discretization mesh has $N = 101$ nodes. This is indicated by setting `n_nodes`= 101; (ii) The time nodes are uniformly spaced in the interval $t \in [0, 1]$. Here the $t_i =$ `initial_time` $= 0.0$, and $t_f =$ `final_time` $= 1.0$; (iii) The initial state discretization mesh is $N_j = 1$, $j = 1, 2, \cdots, 100$; (iv) The initial estimate of the optimal control is $u(t) = 0$; and (v) By default, the coefficient matrix M is initialized to the identity matrix.

The listing given above is stored in a file named `ex6`. Then, to solve this problem using the package `dsoa` we the following command in a system shell;

```
dsoa ex6
```

This command combines the script in the file `ex6` with some additional code to form a C program. This code is compiled, linked with the `dsoa` library functions, and then executed.

Some of the output generated by the numerical solution method is shown below.

```
iter = 1, f = 6.004e-01, |hg| = 0.000e+00, |DL| = 1.397e+00
iter = 2, f = 5.963e-01, |hg| = 0.000e+00, |DL| = 6.388e-02
iter = 3, f = 5.763e-01, |hg| = 6.331e-19, |DL| = 6.239e-02
.
.
.
iter = 205, f = 1.698e-01, |hg| = 5.658e-13, |DL| = 2.495e-04
iter = 206, f = 1.698e-01, |hg| = 8.610e-14, |DL| = 1.219e-04
iter = 207, f = 1.698e-01, |hg| = 1.172e-12, |DL| = 1.027e-04

RSQP_solve: |f1-f0| <= SQRT_DBL_EPSILON*(1+|f0|)

$ f     = 0.169842
$ |hg|  = 1.17245e-12
$ |DL|  = 0.000102673
$
```

As can be seen the algorithm terminates after 207 iterations due to the fact that the NLP function value has a very small change in successive iterations. The program stores the solution data in an ASCII file named `ex6.data`. This file can be read to extract the optimal control and the corresponding state trajectory.

Figure 1 shows the optimal state trajectory and control obtained for this problem. The plot (a) shows the state y_1, the plot (b) shows the state y_2, and the plot (c) shows the control u. (Note that the state y_3 is simply a straight line from 0 to 1.)

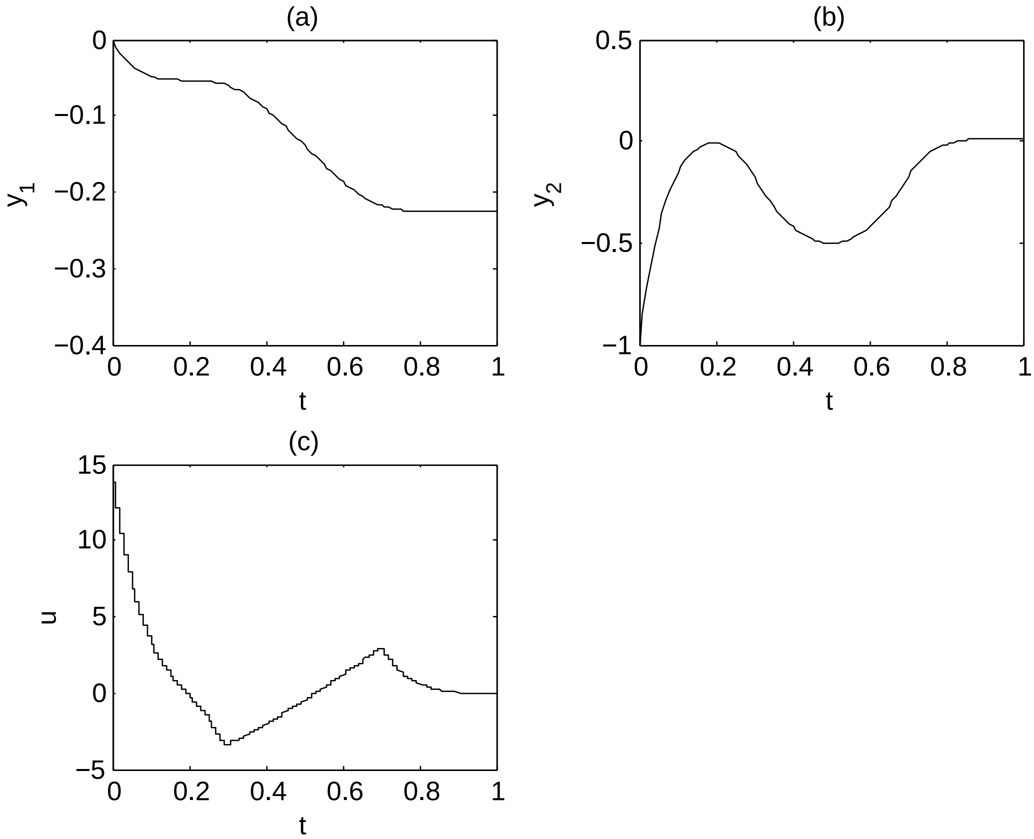

Figure 1: Optimal trajectory for the problem `ex6`

Example 3 A batch reactor control problem can be described using differential-algebraic equations as follows [20].

$$\min_{T(t)\in\mathcal{R}} -x_2(1)$$

subject to

$$\begin{aligned}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{k}_1 \\ \dot{k}_2 \end{bmatrix} &= \begin{bmatrix} -k_1 x_1^2 \\ k_1 x_1^2 - k_2 x_2 \\ k_1 - 4000e^{-2500/T} \\ k_2 - 620000e^{-5000/T} \end{bmatrix}, \\
x_1(0) &= 1, \\
x_2(0) &= 0, \\
0 &\geq T - 398, \\
0 &\geq 298 - T.
\end{aligned}$$

In this problem the states are $x_1(t)$, $x_2(t)$, $k_1(t)$, and $k_2(t)$. The control is $T(t)$, the initial time $t_i = 0$ and the final time is $t_f = 1$.

The `dsoa` input script used to solve this problem is shown below.

```
double x1, x2, k1, k2, T;
int i, j;
n_states     = 4;
n_controls   = 1;
n_initial    = 4;
n_inequality = 2;
n_nodes      = 201;
initial_time = 0.0;
final_time   = 1.0;
tolerance    = 1.0e-6;
output_file  = "ex68dae.data";

set_M {
    M(3,3) = 0.0;
    M(4,4) = 0.0;
}

initial_constraint {
    x1 = y(1);
    x2 = y(2);
    k1 = y(3);
    k2 = y(4);
    T = u(1);
    Gamma(1) = x1-1.0;
    Gamma(2) = x2;
    Gamma(3) = k1-4000.0*exp(-2500.0/T);
    Gamma(4) = k2-620000.0*exp(-5000.0/T);
```

```
28  }
29
30  dynamic_equation_and_cost_function {
31      x1 = y(1);
32      x2 = y(2);
33      k1 = y(3);
34      k2 = y(4);
35      T = u(1);
36      F(1) = -k1*x1*x1;
37      F(2) = k1*x1*x1-k2*x2;
38      F(3) = k1-4000.0*exp(-2500.0/T);
39      F(4) = k2-620000.0*exp(-5000.0/T);
40  }
41
42  inequality_constraint {
43      T = u(1);
44      d(1) = T-398.0;
45      d(2) = 298.0-T;
46  }
47
48  terminal_condition {
49      x2 = y(2);
50      phi = -x2;
51  }
52
53  solution_estimate {
54      y0(1) = 1.0;
55      for(j=1;j<n_nodes;j++)
56          for(i=1;i<=n_controls;i++)
57              u0(i,j) = 348.0;
58  }
```

Lines 13-16 initializes the coefficient matrix M for the DAEs. By default M is set equal to the identity matrix hence, in this script we simply zero the elements $M_{3,3}$ and $M_{4,4}$.

The equations in the initial constraint block (lines 18-28) ensure that $x_1(0) = 1$, $x_2(0) = 0$, and $k_1(0)$, $k_2(0)$ satisfy the DAEs at the initial time.

Lines 53-58 establish an estimate for the unknown initial condition $x_1(0)$, and the control history, $T(t)$. In particular, line 54 sets the estimate of the first state variable equal to 1.0. Lines 55-57 set the control equal to 348.0 in all $N - 1$ time intervals.

The solution of this problem required 4 state mesh refinement iteration to obtain a solution with the desired tolerance $\epsilon = \epsilon_1 = 10^{-6}$. In total, 16 iteration of the NLP solver (`rsqp`) was required to find this solution. The value of the cost functional at the solution is $J = -0.610764$, which compares well with the result reported in [20]. Figure 3a shows the optimal state trajectory for y_1 and y_2, Figure 3b shows the optimal state trajectory for the algebraic variables k_1 and k_2. The optimal control is shown in Fig. 3c. Finally, Fig. 3d shows the state discretization mesh. This plot shows the number of state discretization segments, N_j, in each interval of the control mesh. Recall that

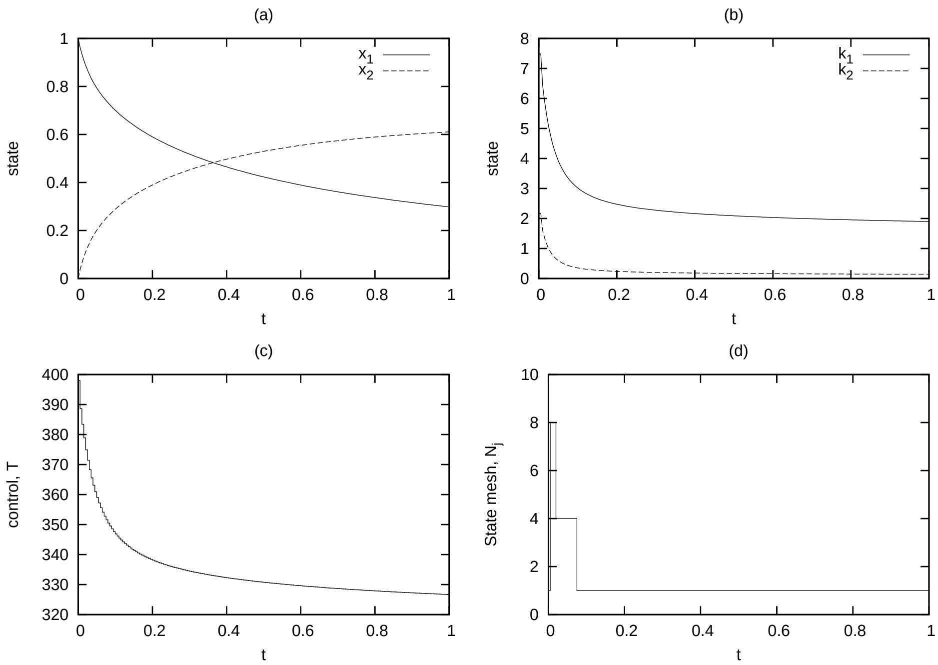

Figure 2: Solution for the problem `ex68dae`

initial state mesh is $N_j = 1$, $j = 1, 2, \cdots, 200$. Fig. 3d shows the state mesh after 4 mesh refinement iterations. As can be seen, a significant increase in the state mesh segments is required to obtain a solution with the desired accuracy.

3.2 Input/Output data files

We will use the following simple example to illustrate the structure of the input/output data files used by `dsoa`.

$$\min_{u(t)} J = \int_0^1 \frac{1}{2} u(t)^2 \, dt$$

subject to

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= u, \\ x_1(0) &= 1, \\ x_2(0) &= 1, \\ x_1(1) &= 0, \\ x_2(1) &= 0. \end{aligned}$$

In this problem we seek the energy optimal control that will take the linear second-order system from the initial state $x_1(0) = 1$, $x_2(0) = 1$ to the origin $x_1(1) = 0$, $x_2(1) = 0$ in 1 second. The `dsoa` script that describes this problem is given below.

```
double x1, x2, u1;
output_file = "ex2test.data";
n_states = 2;
n_controls = 1;
n_parameters = 0;
n_initial = 2;
n_inequality = 0;
n_terminal = 2;
n_nodes = 5;
initial_time = 0.0;
final_time = 1.0;
tolerance = 1.0e-4;

initial_constraint {
       x1 = y(1);
       x2 = y(2);
       u1 = u(1);
       Gamma(1) = x1-1.0;
       Gamma(2) = x2-1.0;
}
dynamic_equation_and_cost_function {
       x1 = y(1);
       x2 = y(2);
       u1 = u(1);
       F(1) = x2;
       F(2) = u1;
       L = 0.5*u1*u1;
}
```

```
29 terminal_condition {
30        x1 = y(1);
31        x2 = y(2);
32        u1 = u(1);
33        Psi(1) = x1;
34        Psi(2) = x2;
35 }
36 solution_estimate {
37        y0(1) = 1.0;
38        y0(2) = 1.0;
39 }
```

This problem is solved using a control discretization mesh with $N = 5$ nodes. As indicated in the script (line 2), the output data is stored in the file `ex2test.data`. The contents of this file is shown in the listing below.

```
n_states
2
n_controls
1
n_parameters
0
n_initial
2
n_terminal
2
n_inequality
0
n_nodes
5
tolerance
1.000000e-04
T
0 0.25 0.5 0.75 1
U
-8.200000e+00 -3.400000e+00 1.400000e+00 6.200000e+00
P

Y
1.000000e+00 9.937500e-01 6.250000e-01 1.937500e-01 -8.326673e-17
1.000000e+00 -1.050000e+00 -1.900000e+00 -1.550000e+00 -8.881784e-16
state_mesh
1 1 1 1
ns
5
Ts
0 0.25 0.5 0.75 1
Ys
1.000000e+00 9.937500e-01 6.250000e-01 1.937500e-01 -8.326673e-17
1.000000e+00 -1.050000e+00 -1.900000e+00 -1.550000e+00 -8.881784e-16
```

All output data files generated by `dsoa` will have a similar structure. The input data files to `dsoa` must also have this structure.

The contents of the input/output data files can be described as follows:

- Line 1 of the output file contains the string `n_states`.
- Line 2 contains the integer n_y.
- Line 3 contains the string `n_controls`.
- Line 4 contains the integer n_u.
- Line 5 contains the string `n_parameters`.
- Line 6 contains the integer n_p.
- Line 7 contains the string `n_initial`.
- Line 8 contains the integer n_Γ.
- Line 9 contains the string `n_terminal`.
- Line 10 contains the integer n_Ψ.
- Line 11 contains the string `n_inequality`.
- Line 12 contains the integer n_d.
- Line 13 contains the string `n_nodes`.
- Line 14 contains the integer N.
- Line 15 contains the string `tolerance`.
- Line 16 contains the floating-point number ϵ.
- Line 17 contains the string `T`.
- Line 18 contains the control mesh time nodes, t_j, $j = 1, 2, \cdots, N$. Each value t_j is separated by a space.
- Line 19 contains the string `U`.
- Lines $19 + 1$ through $19 + \bar{n}_u$ contains the control $u_i(t_j)$, $i = 1, 2, \cdots, n_u$, $j = 1, 2, \cdots, N-1$. Here, $\bar{n}_u = \max(n_u, 1)$. Line $19 + i$ contains the control for the i-th control variable at nodes t_j, $j = 1, 2, \cdots, N-1$. Each control $u_i(t_j)$ is separated by a space.

  For the example problem described above there is only one control variable, hence line 20 contains the control $u(t_j)$, $j = 1, 2, \cdots, 4$.
- Line $19 + \bar{n}_u + 1$ contains the string `P`.

- Line $19 + \bar{n}_u + 2$ contains the n_p parameters p_j, $j = 1, 2, \cdots, n_p$. Each parameter p_j is separated by a space.

  In the example given above there are no parameters in the problem, hence line 22 is blank.

- Line $19 + \bar{n}_u + 3$ contains the string `Y`.

- Lines $19+\bar{n}_u+4$ through $19+\bar{n}_u+3+\bar{n}_y$ contains the state trajectory at the control mesh nodes, i.e., $y_i(t_j)$, $i = 1, 2, \cdots, n_y$, $j = 1, 2, \cdots, N$. Here $\bar{n}_y = \max(n_y, 1)$. Line $19+\bar{n}_u+3+i$ contains the i-th state at the control nodes t_j. Each state $y_i(t_j)$ is separated by a space.

  In the example given above there are two states, hence line 24 has the states $y_1(t_j)$, and line 25 has the states $y_2(t_j)$.

- Line $19 + \bar{n}_u + \bar{n}_y + 4$ contains the string `state_mesh`.

- Line $19+\bar{n}_u+\bar{n}_y+5$ contains the state mesh nodal distribution N_j, $j = 1, 2, \cdots, N-1$. The state discretization mesh if formed by dividing each control mesh interval $[t_j, t_{j+1}]$ into N_j segments so that $t_j = t_{j,1} < t_{j,2} < \cdots < t_{j,N_j+1} = t_{j+1}$, $j = 1, 2, \cdots, N-1$. The mesh in the interval $[t_j, t_{j+1}]$ is called the state discretization mesh, and it coincides with control discretization mesh when $N_j = 1$, for all j.

- Line $19 + \bar{n}_u + \bar{n}_y + 6$ contains the string `ns`.

- Line $19+\bar{n}_u+\bar{n}_y+7$ contains the integer $N_s = 1+\sum_{j=1}^{N-1} N_j$, i.e., the total number of nodes in the state discretization mesh.

- Line $19 + \bar{n}_u + \bar{n}_y + 8$ contains the string `Ts`.

- Line $19+\bar{n}_u+\bar{n}_y+9$ contains the time nodes for the state discretization mesh, t^s_k, $k = 1, 2, \cdots, N_s$. Each t^s_k is separated by a space.

- Line $19 + \bar{n}_u + \bar{n}_y + 10$ contains the string `Ys`.

- Line $19 + \bar{n}_u + \bar{n}_y + 11$ through $19 + \bar{n}_u + 2\bar{n}_y + 10$ contain the states at the time nodes t^s_k, $k = 1, 2, \cdots, N_s$, i.e., $y_i(t^s_k)$, $i = 1, 2, \cdots, n_y$. Line $19 + \bar{n}_u + \bar{n}_y + 10 + i$ contains the i-th state at the time nodes t^s_k. Each value $y_i(t^s_k)$ is separated by a space.

  In the example given above the state discretization mesh coincides with the control discretization mesh, i.e., $N_j = 1$, $j = 1, 2, 3, 4$. Hence, the control mesh time nodes and state mesh time nodes are the same.

3.2.1 The Octave/MATLAB function `load_dsoa`

To facilitate the analysis and the visualization of the output data we have developed an Octave/MATLAB function called `load_dsoa` that reads the output data file generated by the package `dsoa`.

To illustrate the usage of the function `load_dsoa` consider the Octave/MATLAB command;

```
p = load_dsoa('ex6.data');
```

The result of this command is that the function `load_dsoa` read the data stored in the file `ex6.data`, and creates a structure `p` that now contains the problem data.

The data structure `p` has the following members;

- `p.n_states` -The number of state variables, n_y.
- `p.n_controls` -The number of control variables, n_u.
- `p.n_parameters` -The number of parameter variables, n_p.
- `p.n_initial` -The number of initial constraints, n_Γ.
- `p.n_terminal` -The number of terminal constraints, n_Ψ.
- `p.n_inequality` -The number of inequality constraints, n_d.
- `p.n_nodes` -The number of time nodes in the control mesh, N.
- `p.T` -The time nodes for the control mesh, an N matrix.
- `p.U` -The piecewise optimal control history, an $n_u \times N - 1$ matrix.
- `p.P` -The optimal parameters, an $n_p \times 1$ matrix.
- `p.Y` -The optimal state trajectory at the control mesh time nodes, an $n_y \times N$ matrix.
- `p.state_mesh` -The number of state mesh nodes in each control mesh interval, an $N - 1 \times 1$ matrix.
- `p.ns` -The total number of state mesh nodes.
- `p.Ts` -The time nodes for the state discretization mesh, an $N_s \times 1$ matrix.
- `p.Ys` -The optimal state trajectory at the state mesh time nodes, an $n_y \times N_s$ matrix.

To plot the optimal state trajectory $y_2(t)$ we can then use the Octave/MATLAB command;

```
plot(p.Ts, p.Ys(2,:))
```

To plot the optimal control $u_1(t)$ we can use the command;

```
stairs(p.T(1:p.n_nodes-1), p.U(1,:))
```

4 Automatic differentiation

The solution of the problem DSOA requires the evaluation of the derivatives of the functions ϕ, L, F, Γ, d and Ψ with respect to their arguments. The package `dsoa` uses one of two methods to compute these derivatives. The first method is a forward difference approximation of the derivative. The second method uses automatic differentiation. Here automatic differentiation is accomplished using a C++ implementation of the technique described by Rall [66]. To do so we define a class called `ADiff`. The members of this class are; (i) a function value, and (ii) the gradient of the function with respect to the independent variables in the `ADiff` class. The implementation of the `ADiff` class also includes definitions for the operations of addition, subtract, etc., to ensure that the rules of calculus are satisfied when expressions involving `ADiff` variables are computed.

The `dsoa` input scripts in the examples given above all use forward difference approximations to estimate the derivatives of problem functions. To use automatic differentiation for computing the function derivatives we must make two important modifications to the input script. Specifically,

1. The scalar functions ϕ and L must be referenced by `phi[0]` and `L[0]`, respectively. (Instead of `phi` and `L`.)

2. **All** user defined **variables** in the input script must be declared as `ADiff` instead of `double`.

The test problems that use automatic differentiation are in the directory `dsoa/test_ad`. The following example illustrates a `dsoa` input script that uses automatic differentiation.

Example 4

$$\min_{u(t)} J = 10^3 x_1(1)^2 + \int_0^1 \frac{1}{2} u(t)^2 \, dt$$

subject to

$$\begin{aligned}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= -x_1 + 10^{-4} x_2^2 + u, \\
x_1(0) &= 1, \\
x_2(0) &= 1, \\
x_2(1) &= 0.
\end{aligned}$$

The input script for this problem is as follows.

```
ADiff x1;
ADiff x2;
ADiff u1;
double constant1, constant2;
output_file = "adiff_test.data";
n_states = 2;
n_controls = 1;
```

```
8 n_parameters = 0;
9 n_initial = 2;
10 n_inequality = 0;
11 n_terminal = 1;
12 n_nodes = 101;
13 initial_time = 0.0;
14 final_time = 1.0;
15 tolerance = 1.0e-4;
16
17 constant1 = 1.0e3;
18 constant2 = 1.0e-4;
19
20 initial_constraint {
21        x1 = y(1);
22        x2 = y(2);
23        u1 = u(1);
24        Gamma(1) = x1-1.0;
25        Gamma(2) = x2-1.0;
26 }
27 dynamic_equation_and_cost_function {
28        x1 = y(1);
29        x2 = y(2);
30        u1 = u(1);
31        F(1) = x2;
32        F(2) = -x1 + constant2*x2*x2 + u1;
33        L[0] = 0.5*u1*u1;
34 }
35 terminal_condition {
36        x1 = y(1);
37        x2 = y(2);
38        u1 = u(1);
39        Psi(1) = x2;
40        phi[0] = constant1*x1*x1;
41 }
42 solution_estimate {
43        y0(1) = 1.0;
44        y0(2) = 1.0;
45 }
```

Notice that the user defined variables are declared using the type `ADiff` on lines 1, 2 and 3. The constants that appear in the problem description can be declared using the type `double`. Finally, note that the cost functional integrand L is referenced as `L[0]` on line 33, and the terminal penalty ϕ is referenced as `phi[0]` on line 40.

5 Test problems

To demonstrate the effectiveness of the proposed algorithm, the package `dsoa` is applied to 83 optimal control problems that have appeared in the literature. These problems are given below, and are stated using the notation defined by (1)-(5). Included in this test set are 5 problems that are described using index-1 differential-algebraic equations. (These problems have names that end with the string '`dae`'.)

Each problem definition provides the number of nodes used in the control mesh, i.e., N. In each case the initial state discretization mesh is $N_j = 1$, $j = 1, 2, \cdots, N-1$. The state mesh refinement scheme increases the number of nodes in the state discretization mesh if necessary. The problem description also provides an initial estimate of the optimal control, parameters, and initial state. Most problems use a convergence tolerance, and a state local discretization error tolerance $\epsilon = \epsilon_1 = 10^{-4}$. The exceptions are; problems `ex67` and `ex68dae` which use $\epsilon = \epsilon_1 = 10^{-8}$, and problems `ex75` and `ex76` which use $\epsilon = \epsilon_1 = 10^{-6}$.

The dimensions of these optimal control problems fall in the ranges; $1 \leq n_y \leq 10$, $1 \leq n_u \leq 4$, $0 \leq n_p \leq 1$, $1 \leq n_\Gamma \leq 10$, $0 \leq n_d \leq 10$, $0 \leq n_\Psi \leq 10$, and $51 \leq N \leq 251$. The dimensions of the resultant NLP problems fall in the ranges; $51 \leq n_x \leq 410$, $0 \leq n_h \leq 20$, and $0 \leq n_g \leq 1010$.

1 $L = y^2\cos^2 u$, $t_i = 0$, $t_f = \pi$, $F = \frac{1}{2}\sin u$, $\Gamma = [y - \pi/2]$, and $N = 51$. Initial estimate, $u = 0$. Ref. [80].

2 $L = u^2/2$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ u]^T$, $\Gamma = [y_1 - 1,\ y_2 - 1]^T$, $\Psi = [y_1,\ y_2]^T$ and $N = 51$. Initial estimate, $u = 0$. Ref. [68].

3 $\phi = p$, $t_i = 0$, $t_f = 1$, $F = [py_2,\ pu]^T$, $d = [u - 1,\ -1 - u,\ -p]^T$, $\Gamma = [y_1,\ y_2]^T$, $\Psi = [y_1 - 0.25,\ y_2]^T$ and $N = 51$. Initial estimate $p = 2$ and $u(t) = t$, $0 \le t \le 1$. Ref. [1], pp. 106.

4 $L = -y_2$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ u]^T$, $\Gamma = [y_1,\ y_2]^T$, $d = [u - 1,\ -1 - u]^T$, $\Psi = y_2$ and $N = 51$. Initial estimate, $u = 0$. Ref. [83]

5 $L = (y_1^2 + y_2^2 + u^2)/2$, $t_i = 0$, $t_f = 5$, $F = [y_2,\ -y_1 + (1 - y_1^2)y_2 + u]^T$, $\Gamma = [y_1 - 1,\ y_2]^T$, $d = -(y_2 + 0.25)$, and $N = 51$. Initial estimate, $u = 0$. Ref. [74].

6 $L = y_1^2 + y_2^2 + 0.005u^2$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ -y_2 + u,\ 1]^T$, $d = [u - 20,\ -20 - u,\ -(8.0(y_3 - 0.5)^2 - 0.5 - y_2)]^T$, $\Gamma = [y_1,\ y_2 + 1,\ y_3]^T$ and $N = 101$. Initial estimate, $u = 0$. Ref. [44].

7a $\phi = y_1^2 + y_2^2$, $L = (y_1^2 + y_2^2 + 0.1u^2)/2$, $t_i = 0$, $t_f = 0.78$, $a_1 = y_1 + 0.25$, $a_2 = y_2 + 0.5$, $a_3 = y_1 + 2$, $a_4 = a_2\exp(25y_1/a_3)$, $F = [-2a_1 + a_4 - a_1u,\ 0.5 - y_2 - a_4]^T$, $\Gamma = [y_1 - 0.05,\ y_2]^T$, and $N = 156$. Initial estimate, $u = 0$. Ref. [48].

7b $L = (y_1^2 + y_2^2 + 0.1u^2)/2$, $t_i = 0$, $t_f = 0.78$, $a_1 = y_1 + 0.25$, $a_2 = y_2 + 0.5$, $a_3 = y_1 + 2$, $a_4 = a_2\exp(25y_1/a_3)$, $F = [-2a_1 + a_4 - a_1u,\ 0.5 - y_2 - a_4]^T$, $\Gamma = [y_1 - 0.05,\ y_2]^T$, $\Psi = [y_1,\ y_2]^T$ and $N = 156$. Initial estimate, the control from problem 7a.

8 $L = (y_1^2 + y_2^2)/2$, $t_i = 0$, $t_f = 0.78$, $a_1 = y_1 + 0.25$, $a_2 = y_2 + 0.5$, $a_3 = y_1 + 2$, $a_4 = a_2\exp(25y_1/a_3)$, $F = [-2a_1 + a_4 - a_1u,\ 0.5 - y_2 - a_4]^T$, $d = [u - 1,\ -1 - u]^T$, $\Gamma = [y_1 - 0.05,\ y_2]^T$, $\Psi = [y_1,\ y_2]^T$ and $N = 156$. Initial estimate, the control from problem 7b.

9 $L = u^2/2$, $t_i = 0$, $t_f = 1$, $F = [10(V\cos u - Vy_2/h),\ 10V\sin u]^T$, $d = [u - \pi,\ -\pi - u]^T$, $\Gamma = [y_1 - 3.66,\ y_2 + 1.86]^T$, $\Psi = [y_1,\ y_2]^T$, $V = 1$, $h = 1$ and $N = 101$. Initial estimate, $u = 0$. Ref. [14].

10 $\phi = p + 10$, $t_i = 0$, $t_f = 1$, $F = [(p + 10)(V\cos u - Vy_2/h),\ (p + 10)V\sin u]^T$, $d = [u - \pi, -\pi - u,\ -(p + 10)]^T$, $\Gamma = [y_1 - 3.66,\ y_2 + 1.86]^T$, $\Psi = [y_1,\ y_2]^T$, $V = 1$, $h = 1$ and $N = 101$. Initial estimate, $p = 0$, and the control from problem 9.

11 $\phi = y_3$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ u,\ u^2/2]^T$, $d = y_1 - 1/9$, $\Gamma = [y_1,\ y_2,\ y_3]^T$, $\Psi = [y_1,\ y_2 + 1]^T$ and $N = 101$. Initial estimate, $u = 0$. Ref. [14].

12 $\phi = -y_3$, $t_i = 0$, $t_f = 5$, $F = [y_2,\ -2 + u/y_3,\ -0.01u]^T$, $d = [u - 30,\ -30 - u]^T$, $\Gamma = [y_1 - 10,\ y_2 + 2,\ y_3 - 10]^T$, $\Psi = [y_1,\ y_2]^T$ and $N = 51$. Initial estimate, $u = 0$. Ref. [80], pp. 305.

13a $\phi = 10^3(y_1^2 + y_2^2)$, $L = 10^3u^2$, $t_i = 0$, $t_f = 1$, $F = [5\cos y_3,\ 5\sin y_3,\ 5u]^T$, $\Gamma = [y_1 - 4,\ y_2,\ y_3 - \pi/2]^T$, and $N = 201$. Initial estimate, $u = 0$.

13b $\phi = p + 5$, $t_i = 0$, $t_f = 1$, $F = [(p+5)\cos y_3,\ (p+5)\sin y_3,\ (p+5)u]^T$, $d = [u-1,\ -1-u,\ -(p+5)]^T$, $\Gamma = [y_1 - 4,\ y_2,\ y_3 - \pi/2]^T$, $\Psi = [y_1,\ y_2]^T$ and $N = 201$. Initial estimate, $p = 0$, and the control from problem 13a. Ref. [80].

14 $\phi = y_2^2 + y_3^2$, $L = 10^3u^2$, $t_i = 0$, $t_f = 1$, $F = [y_3\cos u,\ y_3\sin u,\ g\sin u]^T$, $\Gamma = [y_1,\ y_2 - 6,\ y_3 - 1]^T$, $\Psi = [y_1 - 6]$, $g = -32.2$, and $N = 101$. Initial estimate, $u = 0$.

15 $\phi = p + 1$, $t_i = 0$, $t_f = 1$, $F = [(p+1)y_3\cos u,\ (p+1)y_3\sin u,\ (p+1)g\sin(u)]^T$, $d = [-y_2 - 0.5y_1 + 5,\ -(p+1)]^T$, $\Gamma = [y_1,\ y_2 - 6,\ y_3 - 1]^T$, $\Psi = y_1 - 6$, $g = -32.2$ and $N = 101$. Initial estimate, $p = 0$, and the control from problem 14. Ref. [50].

16 $L = (u_1^2 + u_2^2)/2$, $t_i = 0$, $t_f = 1$, $H_{11} = I_1 + I_2 + m_1l_1^2 + m_2(L_1^2 + l_2^2 + 2L_1l_2\cos y_2)$, $H_{12} = I_2 + m_2l_2^2 + m_2L_1l_2\cos y_2$, $H_{22} = I_2 + m_2l_2^2$, $h = m_2L_1l_2\sin y_2$, $\Delta = 1/(H_{11}H_{22} - H_{12}^2)$, $F = [y_3,\ y_4,\ \Delta(2hH_{22}y_3y_4 + hH_{22}y_4^2 + hH_{12}y_3^2 + H_{22}u_1 - H_{12}u_2),\ \Delta(-2hH_{12}y_3y_4 - hH_{11}y_3^2 - hH_{12}y_4^2 + H_{11}u_2 - H_{12}u_1)]^T$, $\Gamma = [y_1 - \theta_{1i},\ y_2 - \theta_{2i},\ y_3,\ y_4]^T$, $\Psi = [y_1 - \theta_{1f},\ y_2 - \theta_{2f}, y_3, y_4]^T$, $\theta_{1i} = 0$, $\theta_{2i} = -2$, $\theta_{1f} = 1$, $\theta_{2f} = -1$, $L_1 = L_2 = 0.4$, $m_1 = m_2 = 0.5$, $I_1 = I_2 = 0.1$, $l_1 = l_2 = 0.2$, and $N = 101$. Initial estimate, $u_1 = u_2 = 0$. Ref. [21].

17 $\phi = p+1$, $t_i = 0$, $t_f = 1$, $H_{11} = I_1+I_2+m_1l_1^2+m_2(L_1^2+l_2^2+2L_1l_2\cos y_2)$, $H_{12} = I_2 + m_2l_2^2 + m_2L_1l_2\cos y_2$, $H_{22} = I_2 + m_2l_2^2$, $h = m_2L_1l_2\sin y_2$, $\Delta = 1/(H_{11}H_{22} - H_{12}^2)$, $F = [(p+1)y_3,\ (p+1)y_4,\ (p+1)\Delta(2hH_{22}y_3y_4 + hH_{22}y_4^2 + hH_{12}y_3^2 + H_{22}u_1 - H_{12}u_2),\ (p+1)\Delta(-2hH_{12}y_3y_4 - hH_{11}y_3^2 - hH_{12}y_4^2 + H_{11}u_2 - H_{12}u_1)]^T$, $d = [u_1 - 10, -10 - u_1,\ u_2 - 10,\ -10 - u_2, -(p+1)]^T$, $\Gamma = [y_1 - \theta_{1i},\ y_2 - \theta_{2i},\ y_3,\ y_4]^T$, $\Psi = [y_1 - \theta_{1f},\ y_2 - \theta_{2f}, y_3, y_4]^T$, $\theta_{1i} = 0$, $\theta_{2i} = -2$, $\theta_{1f} = 1$, $\theta_{2f} = -1$, $L_1 = L_2 = 0.4$, $m_1 = m_2 = 0.5$, $I_1 = I_2 = 0.1$, $l_1 = l_2 = 0.2$, and $N = 101$. Initial estimate, $p = 0$, and the control from problem 16.

18 $L = (u_1^2 + u_2^2)/2$, $t_i = 0$, $t_f = 1$, $a_3 = 2/(y_3^2 + 4L_e^2m_Lm_B/m_{LB}^2 + 4I_o/m_{LB})$, $F = [u_1 + y_2^2y_3,\ a_3(u_2 - y_1y_2y_3),\ y_1,\ y_2]^T$, $\Gamma = [y_1,\ y_2,\ y_3 - L_e,\ y_4]^T$, $\Psi = [y_1,\ y_2,\ y_3 - 2(r_e + m_LL_e/m_{LB}),\ y_4 - \pi/10]^T$, $L_e = 0$, $m_B = 40$, $m_L = 0$, $m_{LB} = 40$, $I_o = 0.24$, $r_e = 0.25$, and $N = 101$. Initial estimate, $u_1 = u_2 = 0$. Ref. [78].

19 $\phi = (p+1)$, $t_i = 0$, $t_f = 1$, $a_3 = 2/(y_3^2 + 4L_e^2m_Lm_B/m_{LB}^2 + 4I_o/m_{LB})$, $F = [(p+1)(u_1 + y_2^2y_3),\ (p+1)a_3(u_2 - y_1y_2y_3),\ (p+1)y_1,\ (p+1)y_2]^T$, $d = [u_1 - 2f/m_{LB},\ -2f/m_{LB} - u_1,\ u_2 - 2M/m_{LB}, -2M/m_{LB} - u_2,\ -(p+1)]^T$, $\Gamma = [y_1,\ y_2,\ y_3 - 2(r_e + m_LL_e/m_{LB}),\ y_4]^T$, $\Psi = [y_1,\ y_2,\ y_3 - 2(r_e + m_LL_e/m_{LB}),\ y_4 - \pi/10]^T$, $L_e = 0$, $m_B = 40$, $m_L = 0$, $m_{LB} = 40$, $I_o = 0.24$, $r_e = 0.25$, $f = 5$, $M = 300$, and $N = 101$. Initial estimate, $p = 0$, and the control from problem 18.

20 $\phi = (y_3 - 4)^2$, $L = (u_1^2 + u_2^2 + u_3^2 + u_4^2)/2$, $t_i = 0$, $t_f = 1$, $F = [10y_2,\ 10((u_1 + u_3)\cos y_5 - (u_2 + u_4)\sin y_5)/M,\ 10y_4, 10((u_1+u_3)\sin y_5 + (u_2+u_4)\cos y_5)/M,\ 10y_6,\ 10((u_1+u_3)D - (u_2+u_4)L_e)/I_n]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4,\ y_5,\ y_6]^T$, $\Psi = [y_1 - 4,\ y_5]^T$,

$M = 10$, $D = 5$, $L_e = 5$, $I_n = 12$, and $N = 101$. Initial estimate, $u_1 = u_2 = u_3 = u_4 = 0$. Ref. [1].

21 Same as problem 20, except $\phi = 0$, and $\Psi = [y_1 - 4,\ y_2,\ y_3 - 4, y_5 - \pi/4,\ y_6]^T$. Initial estimate, the control from problem 20.

22 $\phi = p + 10$, $t_i = 0$, $t_f = 1$, $F = [(p + 10)y_2,\ (p + 10)((u_1 + u_3)\cos y_5 - (u_2 + u_4)\sin y_5)/M,\ (p + 10)y_4, (p + 10)((u_1 + u_3)\sin y_5 + (u_2 + u_4)\cos y_5)/M,\ (p + 10)y_6,\ (p+10)((u_1+u_3)D - (u_2+u_4)L_e)/I_n]^T$, $d = [u_1 - 5,\ -5 - u_1,\ u_2 - 5,\ -5 - u_2,\ u_3 - 5,\ -5 - u_3, u_4 - 5,\ -5 - u_4, -(p + 10)]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4,\ y_5,\ y_6]^T$, $\Psi = [y_1 - 4,\ y_2,\ y_3 - 4,\ y_5 - \pi/4, y_6]^T$, $M = 10$, $D = 5$, $L_e = 5$, $I_n = 12$, and $N = 101$. Initial estimate, $p = 0$, and the control from problem 21.

23 $L = 4.5(y_3^2 + y_6^2) + 0.5(u_1^2 + u_2^2)$, $t_i = 0$, $t_f = 1$, $F = [9y_4,\ 9y_5,\ 9y_6,\ 9(u_1 + 17.2656y_3),\ 9u_2,\ -9(u_1 + 27.0756y_3 + 2y_5y_6)/y_2]^T$, $\Gamma = [y_1,\ y_2 - 22,\ y_3,\ y_4,\ y_5 + 1,\ y_6]$, $\Psi = [y_1 - 10,\ y_2 - 14,\ y_3,\ y_4 - 2.5,\ y_5,\ y_6]^T$, and $N = 101$. Initial estimate, $u_1 = u_2 = 0$. Ref. [80].

24 Same as 23 except, $L = 4.5(y_3^2 + y_6^2)$, $d = [u_1 - 2.83374,\ -2.83374 - u_1,\ u_2 - 0.71265,\ -0.80865 - u_2]^T$ Initial estimate, the control from problem 23.

25 Same as 24 except, $d = [y_4 - 2.5,\ -2.5 - y_4,\ y_5 - 1,\ -1 - y_5,\ u_1 - 2.83374,\ -2.83374 - u_1,\ u_2 - 0.71265,\ -0.80865 - u_2]^T$. Initial estimate, the control from problem 24.

26 $\phi = p + 9$, $t_i = 0$, $t_f = 1$, $F = [(p + 9)y_4,\ (p + 9)y_5,\ (p + 9)y_6,\ (p + 9)(u_1 + 17.2656y_3),\ (p + 9)u_2,\ -(p + 9)(u_1 + 27.0756y_3 + 2y_5y_6)/y_2]^T$, $\Gamma = [y_1,\ y_2 - 22,\ y_3,\ y_4,\ y_5 + 1, 0]$, $d = [y_4 - 2.5,\ -2.5 - y_4,\ y_5 - 1,\ -1 - y_5,\ u_1 - 2.83374,\ -2.83374 - u_1,\ u_2 - 0.71265,\ -0.80865 - u_2,\ -(p + 9)]^T$, $\Psi = [y_1 - 10,\ y_2 - 14,\ y_3,\ y_4 - 2.5,\ y_5,\ y_6]^T$, and $N = 101$. Initial estimate, $p = 0$, and the control from problem 25.

27 $\phi = 5y_1^2 + y_2^2$, $t_i = 0$, $t_f = 2.9$, $F = [y_2,\ u - 0.1(1 + 2y_1^2)y_2]^T$, $\Gamma = [y_1 - 1,\ y_2 - 1]$, $d = [1 - 9(y_1 - 1)^2 - ((y_2 - 0.4)/0.3)^2,\ -0.8 - y_2,\ u - 1,\ -1 - u]$, $t \in [0, 1)$, and $N = 101$. Initial estimate, $u = 0$. (Note that the inequality constraints are defined in the interval $0 \leq t < 1$.) Ref. [71].

28 $\phi = (y_1 - 0.70106)^2 + (y_2 - 0.0923)^2 + (y_3 - 0.56098)^2 + (y_4 - 0.43047)^2 + y_5^2 + y_6^2 + y_7^2$, $L = (T_1^2 + T_2^2 + T_3^2)/2$, $t_i = 0$, $t_f = 1$, $F = [100(y_5y_4 - y_6y_3 + y_7y_2)/2,\ 100(y_5y_3 + y_6y_4 - y_7y_1)/2,\ 100(-y_5y_2 + y_6y_1 + y_7y_4)/2,\ -100(y_5y_1 + y_6y_2 + y_7y_3)/2,\ 100((I_2 - I_3)y_6y_7 + u_1S_1)/I_1,\ 100((I_3 - I_1)y_7y_5 + u_2S_2)/I_2,\ 100((I_1 - I_2)y_5y_6 + u_3S_3)/I_3]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4 - 1,\ y_5 - 0.01,\ y_6 - 0.005,\ y_7 - 0.001]^T$, $I_1 = 10^6$, $I_2 = 833333$, $I_3 = 916667$, $S_1 = 550$, $S_2 = 50$, $S_3 = 550$, and $N = 101$. Initial estimate, $u_1 = u_2 = u_3 = 0$. Ref. [47].

29 Same as problem 28 except, $\phi = 0$, $\Psi = [y_1 - 0.70106,\ y_2 - 0.0923,\ y_3 - 0.56098,\ y_4 - 0.43047,\ y_5,\ y_6,\ y_7]^T$. Initial estimate, the control from problem 28.

30 $\phi = p+100$, $t_i = 0$, $t_f = 1$, $F = [(p+100)(y_5y_4 - y_6y_3 + y_7y_2)/2,\ (p+100)(y_5y_3+y_6y_4-y_7y_1)/2,\ (p+100)(-y_5y_2+y_6y_1+y_7y_4)/2,\ -(p+100)(y_5y_1+y_6y_2+y_7y_3)/2,\ (p+100)((I_2-I_3)y_6y_7+u_1S_1)/I_1,\ (p+100)((I_3-I_1)y_7y_5+u_2S_2)/I_2,\ (p+100)((I_1-I_2)y_5y_6+u_3S_3)/I_3]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4-1,\ y_5-0.01,\ y_6-0.005,\ y_7-0.001]^T$, $d = [u_1-1,\ -1-u_1,\ u_2-1,\ -1-u_2,\ u_3-1,\ -1-u_3,\ -(p+100)]^T$, $\Psi = [y_1-0.70106,\ y_2-0.0923,\ y_3-0.56098,\ y_4-0.43047,\ y_5,\ y_6,\ y_7]^T$, $I_1 = 10^6$, $I_2 = 833333$, $I_3 = 916667$, $S_1 = 550$, $S_2 = 50$, $S_3 = 550$, and $N = 101$. Initial estimate, $p = 0$, and the control from problem 29.

31 $L = u^2/2$, $t_i = 0$, $t_f = 1$, $F = [u,\ y_1]^T$, $\Gamma = [y_1-1,\ y_2]^T$, $d = y_2 - l$, $\Psi = [y_1+1,\ y_2]^T$, $l = 0.12$, and $N = 101$. Initial estimate, $u = 0$.

32 $L = y_1^2 + y_2^2$, $t_i = 0$, $t_f = 5$, $F = [y_2,\ u]^T$, $\Gamma = [y_1,\ y_2-1]^T$, $d = [u-1,\ -1-u]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [45].

33 $L = (y_2 - (\frac{3}{4}y_3+1))^2 + (y_1 - (\frac{3}{8}y_3^2 + y_3))^2$, $t_i = 0$, $t_f = 5$, $F = [y_2,\ u, 1]^T$, $\Gamma = [y_1,\ y_2-1,\ y_3]^T$, $d = [u-1,\ -1-u]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [45].

34 $L = y_1^2$, $t_i = 0$, $t_f = 5$, $F = [y_2,\ u]^T$, $\Gamma = [y_1,\ y_2-1]^T$, $d = [u-1,\ -1-u]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [45].

35 $\phi = y_1^2 + y_2^2 + y_3^2 + y_4^2$, $t_i = 0$, $t_f = 4.2$, $F = [-0.5y_1 + 5y_2,\ -5y_1 - 0.5y_2 + u,\ -0.6y_3 + 10y_4,\ -10y_3 - 0.6y_4 + u]^T$, $\Gamma = [y_1-10,\ y_2-10,\ y_3-10,\ y_4-10,]^T$, $d = [u-1,\ -1-u]^T$, and $N = 201$. Initial estimate, $u = 0$. Ref. [43].

36 $L = (y_2^2 - y_1^2)$, $t_i = 0$, $t_f = 2.985$, $F = [y_2,\ u]^T$, $\Gamma = [y_1,\ y_2-1]^T$, $d = [u-1,\ -1-u]^T$, $\Psi = [y_1-0.065,\ y_2+1.336]$, and $N = 201$. Initial estimate, $u = 0$.

37 $L = y_1^2$, $t_i = 0$, $t_f = 2$, $F = u$, $\Gamma = [y_1-1]$, $d = [u-1,\ -1-u]^T$, $\Psi = y_1 - 0.5$, and $N = 101$. Initial estimate, $u = 0$. Ref. [23].

38 $L = y_4 + y_5 + B_1u_1^2/2 + B_2u_2^2/2$, $t_i = 0$, $t_f = 5$, $F = [\gamma - b_1y_1y_3/\eta - b_sy_1y_5/\eta - \mu y_1,\ b_1y_1y_3/\eta - (\mu+k_1)y_2 - u_1r_1y_2 + (1-u_2)pr_2y_3 + b_2y_6y_3/\eta - b_sy_2y_5/\eta,\ k_1y_2 - (\mu+d_1)y_3 - r_2y_3,\ (1-u_2)qr_2y_3 - (\mu+k_2)y_4 + b_s(y_1+y_2+y_6)y_5/\eta,\ k_2y_4 - (\mu+d_2)y_5,\ u_1r_1y_2 + (1-(1-u_2)(p+q))r_2y_3 - b_2y_6y_3/\eta - b_sy_6y_5/\eta - \mu y_6]^T$, $\Gamma = [y_1 - 76\eta/120,\ y_2 - 36\eta/120,\ y_3 - 4\eta/120,\ y_4 - 2\eta/120,\ y_5 - \eta/120,\ y_6 - \eta/120]^T$, $d = [u_1-0.95,\ 0.05-u_1,\ u_2-0.95,\ 0.05-u_2]^T$, $b_1 = 13$, $b_2 = 13$, $b_s = 0.029$, $\mu = 0.0143$, $d_1 = d_2 = 0$, $k_1 = 0.5$, $k_2 = 1$, $r_1 = 2$, $r_2 = 1$, $p = 0.4$, $q = 0.1$, $B_1 = 50$, $B_2 = 500$, $\eta = 30000$, $\gamma = \mu\eta$, and $N = 51$. Initial estimate, $u_1 = u_2 = 0$. Ref. [46].

39 $L = y_1^2 + y_2^2 + y_3^2 + y_4^2 + u^2$, $t_i = 0$, $t_f = 1$, $F = [20y_2,\ 20(y_3^2/y_1 - r_\mu/y_1^2 + a(\rho)\sin u),\ 20(-y_2y_3/y_1 + a(\rho)\cos u),\ 20y_3/y_1]^T$, $a(\rho) = 0.01 + \rho$, $\Gamma = [y_1 - 6,\ y_2,\ y_3 - \sqrt{r_\mu/y_1},\ y_4]^T$, $\Psi = [y_1 - 6.6,\ y_2,\ \sqrt{r_\mu/y_1}]^T$, $r_\mu = 62.5$, $\rho = 0$, and $N = 201$. Initial estimate, $u = 0$. Ref. [16].

40 $\phi = p+20$, $t_i = 0$, $t_f = 1$, $F = [(p+20)y_2,\ (p+20)(y_3^2/y_1 - r_\mu/y_1^2 + a(\rho)\sin u),\ (p+20)(-y_2y_3/y_1 + a(\rho)\cos u),\ (p+20)y_3/y_1]^T$, $a(\rho) = 0.01+\rho$, $\Gamma = [y_1-6,\ y_2,\ y_3 - \sqrt{r_\mu/y_1},\ y_4]^T$, $d = [u-1,\ -1-u,\ -(p+20)]^T$, $\Psi = [y_1 - 6.6,\ y_2,\ \sqrt{r_\mu/y_1}]^T$, $r_\mu = 62.5$, $\rho = -10^{-3}$, and $N = 201$. Initial estimate, $p = 0$, and the control from problem 39.

41 $L = u_1^2+u_2^2+u_3^2+u_4^2$, $t_i = 0$, $t_f = 1$, $F = [y_7\cos(y_6)\cos(y_5) + R_x, y_7\sin(y_6)\cos(y_5), -y_7\sin(y_5) + R_z, y_8 + y_9\sin(y_4)\tan(y_5) + y_{10}\cos(y_4)\tan(y_5), y_9\cos(y_4) - y_{10}\sin(y_4), (y_9\sin(y_4) + y_{10}\cos(y_4))/\cos(y_5), u_1, u_2, u_3, u_4]^T$, $\Gamma = [y_1, y_2, y_3 - 0.02, y_4 - \pi/2, y_5 - 0.1, y_6 + \pi/4, y_7 - 1, y_8, y_9 - 0.5, y_{10} - 0.1]^T$, $\Psi[y_1 - 1,\ y_2 - 0.5,\ y_3,\ y_4 - \pi/2,\ y_5,\ y_6,\ y_7,\ y_8,\ y_9,\ y_{10}]^T$, $R_x = -u_{x_{\max}}(y_1 - c_x)d_2^2e^{-d_1^2}$, $R_z = -(u_{z_{\max}} + \rho)d_2^2e^{-d_1^2}$, $d_1 = (y_1 - c_x)/r_x$, $d_2 = (y_3 - c_z)/c_z$, $u_{x_{\max}} = 2$, $u_{z_{\max}} = 1$, $r_x = 0.1$, $c_x = 0.5$, $c_z = 0.1$, $\rho = 0$, and $N = 101$. Initial estimate, $u_1 = u_2 = u_3 = u_4 = 0$. Ref. [16].

42 Same as problem 41 with $d = [\ u_1 - 15,\ -15 - u_1,\ u_2 - 15,\ -15 - u_2,\ u_3 - 15,\ -15 - u_3,\ u_4 - 15,\ -15 - u_4,\ y_4 - \pi/2 - 0.02,\ -0.02 + \pi/2 - y_4\ ]^T$. Initial estimate, the control from problem 41.

43 Same as problem 42 with $\rho = 0.1$. Initial estimate, the control from problem 42.

44 $L = u^2/2$, $t_i = 0$, $t_f = 1$, $F = [y_1(y_4-\beta)/l + \lambda_1y_2 + \lambda_2y_3,\ \beta\mu_1y_1/l - \lambda_1y_2,\ \beta\mu_2y_1/l - \lambda_2y_3,\ u]^T$, $\Gamma = [y_1 - 5000,\ y_2 - 110,\ y_3 - 18.5,\ y_4]^T$, $d = [u-1,\ -1-u]^T$, $\Psi = y_1 - 20000$, $\lambda_1 = 0.154$, $\lambda_2 = 0.456$, $\mu_1 = 0.65$, $\mu_2 = 0.35$, $l = 0.0015$, $\beta = 0.0076$, and $N = 101$. Initial estimate, $u = 0$. Ref. [80].

45 $\phi = (y_2 - Y)^2$, $L = u^2/2$, $t_i = 0$, $t_f = 1$, $F = [10y_3,\ 10y_4,\ 10a\cos u,\ 10a\sin u]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T$, $Y = 5$, $a = 2$, and $N = 101$. Initial estimate, $u = 0$. Ref. [80].

46 $\phi = p + 10$, $t_i = 0$, $t_f = 1$, $F = [(p+10)y_3,\ (p+10)y_4,\ (p+10)a\cos u,\ (p+10)a\sin u]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T$, $d = [-(p+10)]$, $\Psi = [y_2 - Y,\ y_1 - V(p+10)]^T$, $Y = 5$, $a = 2$, $V = 3$, and $N = 101$. Initial estimate, $p = 0$, and the control from problem 45.

47 $L = u^2/2 + 10y_2^2$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ -(g/l)\sin y_1 + u/(ml^2)]^T$, $\Gamma = [y_1,\ y_2]^T$, $\Psi = [y_1 - \pi,\ y_2]$, $m = 1$, $l = 0.25$, $g = 9.81$, and $N = 101$. Initial estimate, $u = 0$.

48 $\phi = p + 1$, $t_i = 0$, $t_f = 1$, $F = [(p+1)y_2,\ -(p+1)(g/l)\sin y_1 + (p+1)u/(ml^2)]^T$, $\Gamma = [y_1,\ y_2]^T$, $d = [u-4,\ -4-u,\ -(p+1)]^T$, $\Psi = [y_1 - \pi,\ y_2]$, $m = 1$, $l = 0.25$, $g = 9.81$, and $N = 101$. Initial estimate, $p = 0$, and the control from problem 47.

49 $t_i = 0$, $t_f = 1$, $\Gamma = [y_1,\ y_2,\ y_3 - r_0,\ y_4]^T$, $L = 0.5u^2$, $F = [y_2,\ (-my_3(2y_2y_4) + u)/(my_3^2 + I),\ y_3,\ (y_3y_2^2 - g\sin y_1)/(1+k)]^T$, $\Psi = [y_1 + r_0,\ y_2,\ y_3,\ y_4]$, $m = 0.1$, $I = 0.01$, $k = 2/5$, $r_0 = 0.01$, $g = 9.81$, and $N = 101$. Initial estimate, $u = 0$. Ref [52], pp. 110.

50 $t_i = 0$, $t_f = 1$, $L = u^2$, $\Gamma = [y_1,\ y_2,\ y_3]^T$, $F = [y_2,\ 1 - \gamma_1^2(1 + y_3)^2/(\gamma_1 + y_1)^2,\ (1 + y_3)/(\gamma_1 + y_1) - \gamma_2(\gamma_1 + y_1)y_3 + \gamma_2(\gamma_1 + y_1)u]^T$, $\Psi = [y_1 - 0.1,\ y_2]$, $\gamma_1 = 1.11$, $\gamma_2 = 1.162$, and $N = 101$. Initial estimate, $u = 0$. Ref [25].

51 $\phi = p$, $t_i = 0$, $t_f = 1$, $\Gamma = [y_1,\ y_2,\ y_3]^T$, $F = [py_2,\ p(1 - \gamma_1^2(1 + y_3)^2/(\gamma_1 + y_1)^2),\ p((1+y_3)/(\gamma_1+y_1) - \gamma_2(\gamma_1+y_1)y_3+\gamma_2(\gamma_1+y_1)u)]^T$, $d = [u-1,\ -1-u,\ -p]^T$, $\Psi = [y_1 - 0.1,\ y_2]$, $\gamma_1 = 1.11$, $\gamma_2 = 1.162$, and $N = 101$. Initial estimate, $p = 1$ and u from problem 50.

52 $\phi = y_1^2/2$, $L = y_1^2 + u^2 - 2u$, $t_i = 0$, $t_f = 1$, $F = [u,\ 1]^T$, $\Gamma = [y_1,\ y_2]^T$, $d = [-(y_1^2 + u^2 - y_2^2 - 1)]$, and $N = 101$. Initial estimate, $u = 0$. Ref. [82]

53a $L = \beta(y_1^2 + y_2^2 + 0.0005(y_2 + 16y_4 - 8 - 0.1y_3u^2)^2)$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ -y_3u + 16y_4 - 8,\ u,\ 1]^T$, $\Gamma = [y_1,\ y_2 + 1,\ y_3 + \sqrt{5},\ y_4]^T$, $d = [-4 - u,\ u - 10]^T$, $\beta = 1$, and $N = 51$. Initial estimate, $u = 0$. Ref. [67].

53b Same as problem 53a with $\beta = 100$. Initial estimate, the control from problem 53a.

53c Same as problem 53a with $\beta = 1000$. Initial estimate, the control from problem 53a.

54 $\phi = -1000y_2$, $t_i = 0$, $t_f = 1$, $F = [-(p + 9)(k_1y_1 + (k_3 + k_4 + k_5)y_1y_2,\ (p + 9)(k_1y_1 - k_2y_2 + k_3y_1y_2)]^T$, $\Gamma = [y_1 - 1,\ y_2]^T$, $d = [7 - (p+9),\ (p+9) - 11,\ 698.15 - (710 + u),\ (710 + u) - 48.15]$, $k_1 = A_1e^{-B_1/w}$, $k_2 = A_2e^{-B_2/w}$, $k_3 = A_3e^{-B_3/w}$, $k_4 = A_4e^{-B_4/w}$, $k_5 = A_5e^{-B_5/w}$, $A_1 = e^{8.86}$, $A_2 = e^{24.25}$, $A_3 = e^{23.67}$, $A_4 = e^{18.75}$, $A_5 = e^{20.70}$, $B_1 = 10215.4$, $B_2 = 18820.5$, $B_3 = 17008.9$, $B_4 = 14190.8$, $B_5 = 15599.8$, $w = 710 + u$, and $N = 101$. Initial estimate, $u = 0$ and $p = 0$. Ref. [67].

55 $L = 10y_1^3\sqrt{r}$, $t_i = 0$, $t_f = 1$, $F = [\ T(-SCry_1^2 - (g \sin y_2)/(1 + y_3)^2),\ T(SC_ary_1 + (y_1 \cos y_2)/(R(1+y_3)) - (g \cos y_2)/(y_1(1+y_3)^2)),\ T((y_1 \sin y_2)/R),\ T((y_1 \cos y_2)/(1+y_3))\ ]^T$, $\Gamma = [y_1 - 0.35,\ y_2 + 5.75\pi/180,\ y_3 - 4/209.0352,\ y_4]^T$, $d = [-\pi/2 - u,\ u - \pi/2]^T$, $\Psi = [\ y_1 - 0.01239929,\ y_2 + 26.237124\pi/180,\ y_3 - 0.7553/209.0352,\ y_4 - 51.10198\ ]^T$, $T = 100 + p$, $R = 209.0352$, $r_0 = 0.0023769$, $b = 1/0.235$, $S = 25000$, $g = 3.2172 \times 10^{-4}$, $C = 0.88 + 0.52 \cos u$, $C_a = -0.505 \sin u$, $r = r_0e^{-bRy_3}$, and $N = 101$. Initial estimate, $u = 0$ and $p = 0$. Ref. [56].

56 $L = y_1^2 + 0.001u^2$, $t_i = 0$, $t_f = 4$, $F = [y_2,\ u,\ 1]^T$, $d = [-(y_1 - 15 + (y_3 - 4)^4)]$, $\Gamma = [y_1 - 5,\ y_2,\ y_3]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [10].

57 $L = 0.5(\rho(y_1 - 1)^2 + (y_4 + p)^2)$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ y_3,\ y_4 + p,\ u]^T$, $\Gamma = [y_1,\ y_2 - 1,\ y_3 - 2,\ y_4]^T$, $d = [y_1 - l]$, $\Psi = [y_1,\ y_2 + 1,\ y_3 - 2]^T$, $\rho = 1$, $l = 0.134$, and $N = 201$. Initial estimate, $u = 0$. Ref. [11].

58 $L = 0.5(Ru^2 + w_q(y_2^2 + y_4^2 - 2y_2y_4) + 2w_su(y_2 - y_4))$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ -r_0y_1 - r_1y_2 + r_0y_3,\ y_4,\ y_5,\ u]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4,\ y_5]^T$, $\Psi = [y_1 - 1,\ y_2,\ y_3 - 1,\ y_4,\ y_5]^T$, $R = 2.081$, $r_0 = 3.106 \times 10^2$, $r_1 = 0$, $w_s = 6.337 \times 10^2$, $w_q = 3.865 \times 10^5$, and $N = 101$. Initial estimate, $u = 0$. Ref. [1], pp. 120.

59 $L = u^2/2$, $t_i = 0$, $t_f = 1$, $F = [y_2,\ -g + u/y_3,\ -ku]^T$, $\Gamma[y_1 - 1,\ y_2,\ y_3 - 1]^T$, $d = [-y_1,\ -y_3]^T$, $\Psi = [y_1,\ y_2]^T$, $k = 0.01$, $g = 9.8$, and $N = 101$. Initial estimate, $u = 0$. Ref. [80].

60 $\phi = p + 1$, $t_i = 0$, $t_f = 1$, $F = [(p + 1)y_2,\ (p + 1)(-g + u/y_3),\ -(p + 1)ku]^T$, $\Gamma = [y_1 - 1,\ y_2,\ y_3 - 1]^T$, $d = [-y_1,\ -y_3,\ -(p+1),\ u - 12, -u]^T$, $\Psi = [y_1,\ y_2,\ y_3 - 0.9]^T$, $k = 0.01$, $g = 9.8$, and $N = 101$. Initial estimate, $p = 0$ and the control from problem 59.

61dae $L = u^2/2$, $t_i = 0$, $t_f = 1$, $M = \text{diag}(1,\ 1,\ 1,\ 0)$, $F = [y_2,\ y_4,\ -ku,\ -gy_3 + u - y_3y_4]^T$, $\Gamma[y_1 - 1,\ y_2,\ y_3 - 1,\ -gy_3 + u - y_3y_4]^T$, $d = [-y_1,\ -y_3]^T$, $\Psi = [y_1,\ y_2]^T$, $k = 0.01$, $g = 9.8$, and $N = 101$. Initial estimate, $u = 0$.

62dae $L = u^2/2$, $t_i = 0$, $t_f = 1$, $M = \text{diag}(1,\ 1,\ 1,\ 0)$, $F = [y_3,\ y_4,\ (u - c(y_3 - y_4))/m,\ -c(y_4 - y_3) - ky_2]^T$, $\Gamma = [y_1,\ y_2,\ -c(y_4 - y_3) - ky_2]^T$, $\Psi = [y_1 - 1,\ y_3,\ y_4]^T$, $m = 1$, $k = 10$, $c = 0.1$, and $N = 101$. Initial estimate, $u = 0$.

63dae $\phi = p + 1$, $t_i = 0$, $t_f = 1$, $M = \text{diag}(1,\ 1,\ 1,\ 0)$, $F = [(p + 1)y_3,\ (p + 1)y_4,\ (p + 1)(u - c(y_3 - y_4))/m,\ (p + 1)(-c(y_4 - y_3) - ky_2)]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T$, $d = [u - 10,\ -10 - u]^T$, $\Psi = [y_1 - 1,\ y_3,\ y_4]^T$, $m = 1$, $k = 10$, $c = 0.1$, and $N = 101$. Initial estimate, $p = 0$ and the control from problem 62dae.

64dae Same as problem 63dae but with $d = [u - 10,\ -10 - u,\ y_2 - 0.025]^T$. Initial estimate, p and the control from problem 63dae.

65a $L = y_1^2 + u^2$, $t_i = 0$, $t_f = 1$, $F = [y_1 - u]$, $\Gamma = [y_1 - 1]$, $d = [-y_1 + 0.9]$, $\Psi = [y_1 - 1]$, and $N = 101$. Initial estimate, $u = 0$. Ref. [56].

65b $L = y_1^2 + u^2$, $t_i = 0$, $t_f = 1$, $F = [y_1^2 - u]$, $\Gamma = [y_1 - 1]$, $d = [-y_1 + 0.9]$, $\Psi = [y_1 - 1]$, and $N = 101$. Initial estimate, $u = 0$. Ref. [56].

66a $L = 10^3u^2$, $t_i = 0$, $t_f = 1$, $F = [10y_2,\ 10(u - y_1),\ 10y_4,\ 10u]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T$, $\Psi = [y_1,\ y_2,\ y_3 - 1,\ y_4]^T$, and $N = 251$. Initial estimate, $u = 0$. Ref. [56].

66b $\phi = p + 10$, $t_i = 0$, $t_f = 1$, $F = [(p + 10)y_2,\ (p + 10)(u - y_1),\ (p + 10)y_4,\ (p + 10)u]^T$, $\Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T$, $d = [-(p + 10),\ u - 1,\ -1 - u,\ y_4 - \alpha]^T$, $\Psi = [y_1,\ y_2,\ y_3 - 1,\ y_4]^T$, $\alpha = 0.4$, and $N = 251$. Initial estimate, $u = 0$. Ref. [56].

67 $\phi = -y_2$, $t_i = 0$, $t_f = 1$, $\Gamma = [y_1 - 1,\ y_2]^T$, $F = [-k_1y_1^2,\ k_1y_1^2 - k_2y_2]^T$, $k_1 = 4000e^{-2500/u}$, $k_2 = 620000e^{-5000/u}$, $d = [u - 398,\ 298 - u]^T$, $N = 201$, and $\epsilon = 10^{-8}$. Initial estimate, $u = 348$.

68dae $\phi = -y_2$, $t_i = 0$, $t_f = 1$, $\Gamma = [y_1 - 1,\ y_2,\ y_3 - 4000e^{-2500/u},\ y_4 - 620000e^{-5000/u}]^T$, $M = \text{diag}(1,\ 1,\ 0,\ 0)$, $F = [-y_3y_1^2,\ y_3y_1^2 - y_4y_2,\ y_3 - 4000e^{-2500/u},\ y_4 - 620000e^{-5000/u}]^T$, $d = [u - 398,\ 298 - u]^T$, $N = 201$, and $\epsilon = 10^{-8}$. Initial estimate, $u = 348$.

69 $\phi = -1 + y_1 + y_2$, $t_i = 0$, $t_f = 12$, $\Gamma = [y_1 - 1,\ y_2]^T$, $F = [u(10y_2 - y_1),\ -u(10y_2 - y_1) - (1 - u)y_2]^T$, $d = [u - 1,\ -u]^T$, and $N = 101$. Initial estimate, $u = 0$.

70 $t_i = 0,\ t_f = 0.2,\ L = -(5.8(qy_1 - u_4) - 3.7u_1 - 4.1u_2 + q(23y_4 + 11y_5 + 28y_6 + 35y_7) - 5u_3^2 - 0.099),\ \Gamma = [y_1 - 0.1883,\ y_2 - 0.2507,\ y_3 - 0.0467,\ y_4 - 0.0899,\ y_5 - 0.1804,\ y_6 - 0.1394,\ y_7 - 0.1046]^T,\ F = [u_4 - qy_1 - 17.6y_1y_2 - 23y_1y_6u_3,\ u_1 - qy_2 - 17.6y_1y_2 - 146y_2y_3,\ u_2 - qy_3 - 73y_2y_3,\ -qy_4 + 35.2y_1y_2 - 51.3y_4y_5,\ -qy_5 + 219y_2y_3 - 51.3y_4y_5,\ -qy_6 + 102.6y_4y_5 - 23y_1y_6u_3,\ -qy_7 + 46y_1y_6u_3]^T,\ q = u_1 + u_2 + u_4,\ d = [u_1 - 20,\ -u_1,\ u_2 - 6,\ -u_2,\ u_3 - 4,\ -u_3,\ u_4 - 20,\ -u_4]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [3].

71 $\phi = -y_2,\ t_i = 0,\ t_f = 1,\ \Gamma = [y_1 - 1,\ y_2]^T,\ F = [-(1 + 0.5u)uy_1,\ uy_1]^T,\ d = [u - 5,\ -u]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [20].

72 $\phi = y_2,\ t_i = 0,\ t_f = 1,\ \Gamma = [y_1 - 1,\ y_2]^T,\ F = [u,\ y_1^2 + u^2]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [55].

73 $t_i = 0,\ t_f = 1,\ L = (u_1 + 1)^2 + (u_2 + 1)^2,\ \Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T,\ F = [y_3,\ y_4,\ u_1 + 1,\ u_2 + 1]^T,\ d = [0.1 - ((y_1 - 0.4)^2 + (y_2 - 0.5)^2),\ 0.1 - ((y_1 - 0.8)^2 + (y_2 - 1.5)^2)]^T,\ \Psi = [y_1 - 1.2,\ y_2 - 1.6]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [4].

74 $t_i = 0,\ t_f = 1,\ L = u_1^2 + u_2^2,\ \Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T,\ F = [5y_3,\ 5y_4,\ 5u_1,\ 5u_2]^T,\ \Psi = [y_1 - 1.2,\ y_2 - 1.6,\ y_3,\ y_4]^T$, and $N = 101$. Initial estimate, $u = 0$. Ref. [4].

75 $\phi = p_1 + 5,\ t_i = 0,\ t_f = 1,\ \Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T,\ F = [(p_1 + 5)y_3,\ (p_1 + 5)y_4,\ (p_1 + 5)u_1,\ (p_1 + 5)u_2]^T,\ d = [-(p_1 + 5),\ u_1 - 6,\ -4 - u_1,\ u_2 - 4,\ -1 - u_2]^T,\ \Psi = [y_1 - 1.2,\ y_2 - 1.6,\ y_3,\ y_4]^T$, and $N = 101$. Initial estimate, $p_1 = 0$ and the solution form problem 74.

76 $\phi = p_1 + 5,\ t_i = 0,\ t_f = 1,\ \Gamma = [y_1,\ y_2,\ y_3,\ y_4]^T,\ F = [(p_1 + 5)y_3,\ (p_1 + 5)y_4,\ (p_1 + 5)u_1,\ (p_1 + 5)u_2]^T,\ d = [-(p_1 + 5),\ u_1 - 6,\ -4 - u_1,\ u_2 - 4,\ -1 - u_2,\ 0.1 - ((y_1 - 0.4)^2 + (y_2 - 0.5)^2),\ 0.1 - ((y_1 - 0.8)^2 + (y_2 - 1.5)^2)]^T,\ \Psi = [y_1 - 1.2,\ y_2 - 1.6,\ y_3,\ y_4]^T$, and $N = 101$. Initial estimate, the solution form problem 75.

77 $L = y_1^2 + y_2^2,\ t_i = 0,\ t_f = 0.85,\ F = [(y_2 - y_4^2 - y_1)/y_4,\ -((y_4 s'_{22} u + s_{22})y_2 + (y_4 s'_{12} u - s_{11})y_1 - y_4^2 s_{12})/(y_4 s_{22}),\ u,\ 1]^T,\ \Gamma = [y_1,\ y_4 - 0.15]^T,\ d = [-\pi/2 - y_3,\ y_3 - \pi/2]^T,\ \Psi = [y_1],\ s_{11}(y_3) = a_{22}(y_3)/\Delta(y_3),\ s_{22}(y_3) = a_{11}(y_3)/\Delta(y_3),\ s_{12}(y_3) = -a_{12}(y_3)/\Delta(y_3),\ a_{11}(y_3) = \lambda Q_{22} + (1 - \lambda)Q_{rr}(y_3),\ a_{12}(y_3) = \lambda Q_{12} + (1 - \lambda)Q_{rt}(y_3),\ a_{22}(y_3) = \lambda Q_{11} + (1 - \lambda)Q_{tt}(y_3),\ \Delta(y_3) = a_{11}(y_3)a_{22}(y_3) - a_{12}^2(y_3),\ Q_{rr}(y_3) = Q_{11}\cos^4 y_3 + Q_{22}\sin^4 y_3 + (2Q_{12} + 4Q_{66})\cos^2 y_3 \sin^2 y_3,\ Q_{rt}(y_3) = Q_{12}(\cos^4 y_3 + \sin^4 y_3) + (Q_{11} + Q_{22} - 4Q_{66})\cos^2 y_3 \sin^2 y_3,\ Q_{tt}(y_3) = Q_{11}\sin^4 y_3 + Q_{22}\cos^4 y_3 + (2Q_{12} + 4Q_{66})\cos^2 y_3 \sin^2 y_3,\ s'_{12}(y_3) = ds_{12}/dy_3,\ s'_{22}(y_3) = ds_{22}/dy_3,\ Q_{11} = 1.00003,\ Q_{12} = 1.07643 \times 10^{-4},\ Q_{22} = 4.3057 \times 10^{-4},\ Q_{66} = 1.63613 \times 10^{-4},\ \lambda = 0.71$, and $N = 201$. Initial estimate, $u = 1$. Ref. [29].

5.1 Numerical results

All computations, reported below, are performed using the Solaris operating system (`uname -a: SunOS solaris 5.11 snv_86 i86pc i386 i86pc`) on a workstation equipped with an AMD Athlon 64 X2 Dual Core Processor 5200+. The `dsoa` C library is compiled using the Sun C compiler (`cc -V: Sun C 5.9 SunOS_i386 2007/11/15`) with the options `-O -m64 -xlibmieee`. The `dsoa` C++ library is compiled using the Sun C++ compiler (`CC -V: Sun C++ 5.9 SunOS_i386 2007/11/15`) with the options `-O -m64 -xlibmieee`.

Here we evaluate the performance of the `dsoa` package using two different NLP solvers, namely, `linfsqp` [30] and `rsqp` [31]. The solver `linfsqp` is a sequential quadratic programming technique that uses an L_∞ merit function, and strictly convex quadratic programming problems to determine minimizing search directions. The solver `rsqp` is a sequential quadratic programming technique that uses an L_1 merit function, and solves an expanded convex quadratic programming problem in the cases where the linearized constraints of the NLP problem are inconsistent.

We also compare the use of forward difference approximations (FD), and automatic differentiation (AD) for computing the derivatives of the functions ϕ, L, F, Γ, d and Ψ, with respect to their arguments. To evaluate the performance of the package `dsoa` we consider the four 'methods':

1. The NLP solver `rsqp` using forward difference gradient approximations.
2. The NLP solver `rsqp` using automatic differentiation.
3. The NLP solver `linfsqp` using forward difference gradient approximations.
4. The NLP solver `linfsqp` using automatic differentiation.

The tables below summarize the results obtained using `dsoa` to solve the test problems. The first column in the tables gives the problem name. Columns two through four gives `f`, `|hg|`, and `|DL|`, respectively. (Recall that `f` is the value of the NLP cost function, `|hg|` is the infinity norm of the constraints, and `|DL|` is the norm of the gradient of the Lagrangian.) Column five in tables gives the number of function evaluations performed, and column six gives the number of gradient evaluations performed. The 'exit code' from function is given in column seven. A positive exit code indicates that the algorithm has met one of its convergence criteria, while a negative exit code indicates that the algorithm has terminated unsuccessfully. Finally, column eight in tables gives the solution time in seconds. (Note that the computation time is measured in 1 second intervals. Hence, solution times less than 1 second are recorded as 0.)

References

[1] Agrawal, S. K. and Fabien, B. C., *Optimization of Dynamic Systems*, Kluwer Academic Publishers, 1999.

[2] Ascher, U. M, and Petzold, L. R., *Computer methods for ordinary differential equations and differential-algebraic equations*, SIAM, 1998.

[3] Balsa-Canto, M., Banga, J., Alonso, A., and Vassiliadis, V., Dynamic optimization of chemical and biochemical processes using restricted second-order information. *Computers and Chemical Engineering*, Vol. 25, pp. 539–546, 2001.

[4] Bhattacharya, R., OPTRAGEN: A MATLAB Toolbox for Optimal Trajectory Generation, *45th IEEE Conference on Decision and Control*, pp. 6832 – 6836, 2006.

[5] Bauer, I., Bock, H., Korkel, S. and Schlöder, J., Numerical methods for optimum experimental design in DAE systems, *Journal of Computational and Applied mathematics*, Vol. 120, pp. 1–25, 2000.

[6] Bertsekas, D. P., *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[7] Betts, J. T., Survey of numerical methods for trajectory optimization, *AIAA Journal Guidance, Control, and Dynamics*, Vol. 21, 193–207, 1998.

[8] Betts, J. T. and Huffman, W. P., Application of sparse nonlinear programming to trajectory optimization, *AIAA Journal Guidance, Control and Dynamics*, Vol. 15, 198–206, 1992.

[9] Betts, J. T. and Huffman, W. P., Mesh refinement in direct transcription methods for optimal control, *Optimal Control Applications and methods*, Vol. 19, 1–21, 1998.

[10] Betts, J. T., Biehn, N. and Campbell, S. L., Convergence of nonconvergent IRK discretizations of optimal control problems with state inequality constraints, *SIAM J. Sci. Comput.*, pp. 1981–2007, 2002.

[11] Betts, J. T., Campbell, S. L. and Engelsone, A., Direct transcription solution of optimal control problems with higher order state constraints: theory vs. practice, *Optimization and Engineering*, pp. 1–19, vol. 8, 2007.

[12] Bock, H. G., Numerical solution of nonlinear multipoint boundary value problems with application to optimal control, *ZAMM*, Vol. 58, T407–T409, 1978.

[13] Bock, H. G. and Plitt, K. J., A multiple shooting algorithm for direct solution of optimal control problems, *IFAC 9-th Triennial World Congress*, 1984, 1603–1608.

[14] Bryson, A. E. & Ho, Y., *Applied Optimal Control*, Hemisphere, 1975.

Prob.	f	|hg|	|DL|	#fun	#grad	exit	time (s)
ex1	1.69142e-05	0	8.11132e-05	221	221	0	2
ex2	14.0054	2.44249e-15	2.9917e-09	2	2	0	0
ex3	1	4.78155e-08	4.4638e-07	48	48	0	1
ex4	-0.25	4.44089e-16	1.00218e-14	58	58	0	1
ex5	1.79688	3.90467e-09	0.000116415	22	22	1	0
ex6	0.169842	1.17245e-12	0.000102673	210	208	3	14
ex7a	0.014215	4.41092e-20	0.000223874	35	32	1	3
ex7b	0.0167044	4.11449e-08	0.00357329	14	11	3	2
ex8	0.00100151	9.00726e-10	0.0261838	6	6	3	1
ex9	0.377186	1.41224e-06	0.000109481	37	35	3	1
ex10	5.45798	1.85769e-06	0.000352381	60	57	3	2
ex11	2.21248	1.11022e-16	4.05871e-10	2	2	0	0
ex12	-8.8692	2.89159e-07	0.0287717	5	5	3	0
ex13a	280.033	7.42576e-22	0.0120482	87	30	3	8
ex13b	4.7391	1.35769e-07	0.000238623	82	82	3	31
ex14	171.82	6.57068e-08	0.00904803	60	20	3	1
ex15	0.741945	2.515e-07	6.88965e-05	51	51	0	4
ex16	1.38125	6.35427e-09	7.62154e-05	16	14	0	3
ex17	0.392341	8.35981e-08	0.000125505	1064	1064	3	431
ex18	1.24003	7.52729e-11	0.00011452	108	58	3	11
ex19	2.13503	1.1608e-06	0.147409	1045	559	3	209
ex20	0.239117	6.62394e-09	0.00079298	9	8	1	8
ex21	0.969282	3.29752e-07	0.000218752	45	40	3	39
ex22	4.16072	6.18084e-08	0.000380213	535	533	3	1655
ex23	0.343924	1.59854e-07	1.55203e-05	12	12	0	3
ex24	0.00521918	1.47507e-07	7.26563e-05	78	78	0	21
ex25	0.00521575	5.05416e-09	6.75261e-05	3	3	0	1
ex26	8.7019	4.36089e-07	1.64968e-06	17	17	0	9
ex27	3.84597e-11	2.02282e-10	6.90773e-05	16	16	0	1
ex28	0.0792513	2.51466e-19	0.00329332	16	16	1	8
ex29	0.714178	1.32564e-08	1.1293e-05	13	13	0	7
ex30	98.8302	6.43601e-09	0.000429937	72	72	1	128
ex31	3.70442	3.60822e-16	6.05971e-10	2	2	0	0
ex32	0.753991	4.95618e-19	0.000314628	95	94	3	5
ex33	6.04328e-06	2.89321e-20	7.01945e-05	83	83	0	5
ex34	0.268489	1.0407e-18	9.09173e-05	126	125	0	7
ex35	1.00633	4.44089e-16	9.72192e-05	109	109	0	52
ex36	0.0221832	8.32667e-16	8.67149e-05	36	36	0	9
ex37	0.375023	0	7.13755e-05	96	96	0	5
ex38	5152.36	5.55112e-16	0.124809	73	26	3	3
ex39	84.5717	3.15504e-07	0.0011541	201	148	3	44
ex40	-1.87396	3.07449e-09	0.000201457	67	64	3	19
ex41	222.185	3.16875e-09	0.00275962	45	44	1	58
ex42	228.433	4.10476e-10	0.0915751	20	13	3	22
ex43	228.52	3.34278e-09	0.00267232	8	8	1	14
ex44	0.00012382	8.11728e-05	0.00119983	18	8	3	1
ex45	0.000938975	1.18066e-18	9.37449e-05	9	4	0	0
ex46	3.35066	2.14838e-06	0.000395924	76	76	3	6
ex47	103.051	2.58072e-09	0.0236336	87	58	1	2
ex48	0.460076	1.354e-06	0.000117301	241	233	3	16
ex49	4.12277e-05	4.50687e-10	1.36563e-05	3	3	0	1

`rsqp` with forward difference gradients.

Prob.	f	|hg|	|DL|	#fun	#grad	exit	time (s)
ex50	0.668774	1.81117e-08	1.01928e-05	11	11	0	0
ex51	0.978342	1.39361e-10	1.92689e-05	95	93	0	8
ex52	0.921548	5.55112e-16	1.37259e-06	90	79	0	5
ex53a	0.119836	2.11995e-19	0.00024368	158	147	3	12
ex53b	11.9382	7.35583e-18	0.0375762	44	34	3	3
ex53c	119.289	4.76114e-19	0.205298	60	50	3	4
ex54	-353.716	1.46444e-30	10.3602	286	282	3	19
ex55	5.19894e-05	9.59816e-06	0.916215	35	19	3	5
ex56	323.138	7.77156e-16	0.000892182	222	222	1	15
ex57	969.855	4.61769e-10	0.00167019	63	62	1	18
ex58	1765.82	4.17328e-11	9.53044	30	28	3	2
ex59	48.2338	3.19032e-12	0.000222181	9	9	1	0
ex60	1.07511	1.20739e-10	1.98864e-07	4	4	0	1
ex61dae	48.2335	4.30162e-07	4.40845e-05	14	14	0	3
ex62dae	6.40161	7.94476e-13	1.37948e-09	2	2	0	0
ex63dae	0.647171	3.71858e-08	0.000104227	452	452	3	44
ex64dae	0.664495	1.95402e-08	0.000207858	334	331	3	32
ex65a	1.78266	1.11022e-16	7.22692e-05	24	24	0	1
ex65b	1.65624	4.38066e-10	9.64904e-05	23	23	0	1
ex66a	1.26511	4.66294e-15	1.06312e-09	4	3	0	2
ex66b	4.67784	1.56564e-10	0.000136687	246	242	1	304
ex67	-0.608469	2.48369e-20	3.65168e-05	13	13	1	2
ex68dae	-0.608471	7.47842e-09	3.69291e-05	17	17	2	8
ex69	-0.477692	1.44825e-18	9.68561e-05	53	53	0	3
ex70	-21.8227	6.69658e-25	0.00364251	944	942	3	2343
ex71	-0.573531	0	8.90343e-05	139	139	0	6
ex72	0.761599	0	3.85441e-05	9	9	0	0
ex73	24.255	7.4829e-14	0.00688301	42	42	1	13
ex74	0.0768077	4.44089e-16	9.14475e-11	2	2	0	0
ex75	2	9.48354e-08	1.39791e-07	97	97	0	58
ex76	2	1.37467e-09	0.000423621	163	163	1	200
ex77	0.144122	1.76643e-08	6.26336e-05	7	6	0	2

`rsqp` with forward difference gradients, continued.

Prob.	f	|hg|	|DL|	#fun	#grad	exit	time (s)
ex1	1.69142e-05	0	8.11131e-05	221	221	0	2
ex2	14.0054	2.44249e-15	2.77546e-14	2	2	0	0
ex3	1	4.78391e-08	4.46401e-07	48	48	0	1
ex4	-0.25	4.44089e-16	1.00218e-14	58	58	0	1
ex5	1.79688	3.90297e-09	0.000116413	22	22	1	1
ex6	0.169843	9.25627e-20	0.000122067	211	209	3	16
ex7a	0.014215	1.43416e-19	0.000223863	35	32	1	4
ex7b	0.0167044	4.11523e-08	0.00357315	14	11	3	1
ex8	0.00100151	9.03663e-10	0.0261832	6	6	3	1
ex9	0.377186	1.41224e-06	0.000109481	37	35	3	2
ex10	5.45798	1.85769e-06	0.000352381	60	57	3	3
ex11	2.21248	1.11022e-16	1.58892e-14	2	2	0	0
ex12	-8.8692	2.89158e-07	0.0287715	5	5	3	0
ex13a	280.033	1.04324e-19	0.0120473	87	30	3	8
ex13b	4.7391	1.38016e-07	0.000238593	82	82	3	32
ex14	171.82	6.55859e-08	0.00903934	60	20	3	2
ex15	0.741945	2.51755e-07	6.88709e-05	51	51	0	4
ex16	1.38125	6.34722e-09	7.62165e-05	16	14	0	2
ex17	0.392421	1.19841e-05	0.000170623	1020	1020	3	418
ex18	1.24003	5.48009e-11	0.000114549	108	59	3	12
ex19	2.15019	0.000324641	0.0907043	1201	678	3	264
ex20	0.239117	6.62277e-09	0.000792978	9	8	1	8
ex21	0.969282	3.29741e-07	0.000218752	45	40	3	39
ex22	4.16061	1.09029e-07	0.000206733	565	563	3	1819
ex23	0.343924	1.59856e-07	1.55199e-05	12	12	0	3
ex24	0.00521918	1.47498e-07	7.26555e-05	78	78	0	21
ex25	0.00521575	5.04723e-09	6.75261e-05	3	3	0	1
ex26	8.7019	4.36101e-07	1.64968e-06	17	17	0	10
ex27	3.82392e-11	1.98922e-10	6.90772e-05	16	16	0	1
ex28	0.0792513	1.76098e-19	0.00329332	16	16	1	9
ex29	0.714178	1.32567e-08	1.13732e-05	13	13	0	8
ex30	98.8302	6.43566e-09	0.000429938	72	72	1	132
ex31	3.70442	3.60822e-16	5.58999e-14	2	2	0	0
ex32	0.753991	1.62195e-19	0.00031463	95	94	3	5
ex33	6.04781e-06	1.42486e-20	7.01945e-05	83	83	0	5
ex34	0.268489	1.46539e-18	9.0885e-05	126	125	0	7
ex35	1.00633	8.88178e-16	9.72192e-05	109	109	0	54
ex36	0.0221832	1.06859e-15	8.67703e-05	36	36	0	9
ex37	0.375023	0	7.13665e-05	96	96	0	5
ex38	5152.36	5.55112e-16	0.124858	74	26	3	3
ex39	84.5717	3.06937e-07	0.00112612	201	148	3	47
ex40	-1.87396	3.07545e-09	0.000201523	67	64	3	21
ex41	222.185	3.1828e-09	0.00275923	45	44	1	60
ex42	228.433	4.10738e-10	0.0915822	20	13	3	22
ex43	228.52	3.34421e-09	0.00267165	8	8	1	15
ex44	0.00012382	8.13806e-05	0.00119983	18	8	3	1
ex45	0.000938975	1.13948e-19	9.37451e-05	9	4	0	1
ex46	3.35066	2.14838e-06	0.000395924	76	76	3	7
ex47	103.051	2.60036e-09	0.0236323	87	58	1	3
ex48	0.460076	1.35459e-06	0.000117301	241	233	3	17
ex49	4.12277e-05	4.5106e-10	1.36563e-05	3	3	0	1

`rsqp` with automatic differentiation.

Prob.	f	|hg|	|DL|	#fun	#grad	exit	time (s)
ex50	0.668774	1.81126e-08	1.01923e-05	11	11	0	1
ex51	0.978342	1.391e-10	1.92688e-05	95	93	0	9
ex52	1.25232	1.09468e-20	8.94134e-08	73	62	0	4
ex53a	0.119836	2.8823e-18	0.000243848	158	146	3	13
ex53b	11.9382	2.13125e-19	0.0382507	44	33	3	3
ex53c	119.289	1.26564e-18	0.227904	61	49	3	5
ex54	-353.75	1.22941e-30	2.42999	355	347	3	29
ex55	4.66554e-05	2.41183e-06	0.144121	33	19	3	8
ex56	323.138	5.06379e-28	0.00133406	214	214	3	15
ex57	969.855	5.04041e-14	0.00167019	62	62	1	19
ex58	1765.82	9.36553e-15	9.49778	30	28	3	3
ex59	48.2338	5.11952e-08	5.64798e-06	9	9	0	2
ex60	1.07511	1.2081e-10	1.98856e-07	4	4	0	1
ex61dae	48.2335	4.3715e-07	5.1043e-05	13	13	0	2
ex62dae	6.40161	1.44329e-15	1.99162e-14	2	2	0	0
ex63dae	0.647556	5.76696e-08	0.000163689	434	434	3	45
ex64dae	0.664454	6.41522e-09	0.000237021	339	337	3	36
ex65a	1.78266	1.77636e-15	7.22694e-05	24	24	0	1
ex65b	1.65624	4.38171e-10	9.64895e-05	23	23	0	1
ex66a	1.26511	1.77636e-15	1.46936e-13	4	3	0	2
ex66b	4.67784	1.77167e-11	0.000277679	245	241	3	310
ex67	-0.608469	9.93502e-20	3.65168e-05	13	13	1	3
ex68dae	-0.608471	7.46949e-09	3.69291e-05	17	17	2	8
ex69	-0.477692	6.68174e-21	9.68562e-05	53	53	0	2
ex70	-21.8212	1.77971e-24	0.00125276	733	732	3	1666
ex71	-0.573531	0	8.90342e-05	139	139	0	7
ex72	0.761599	0	3.85441e-05	9	9	0	1
ex73	24.255	6.66134e-16	0.00688287	42	42	1	12
ex74	0.0768077	4.44089e-16	4.86356e-16	2	2	0	0
ex75	2	9.47431e-08	1.40136e-07	97	97	0	30
ex76	2	4.18461e-10	4.53699e-08	179	179	0	166
ex77	0.144122	1.76633e-08	6.26342e-05	7	6	0	2

`rsqp` with automatic differentiation, continued.

Prob.	f	|hg|	|DL|	#fun	#grad	exit	time (s)
ex1	1.69142e-05	0	8.11132e-05	221	221	0	2
ex2	14.0054	2.44249e-15	2.9917e-09	2	2	0	0
ex3	1	8.40994e-14	4.50126e-07	52	49	0	1
ex4	-0.25	4.44089e-16	1.00218e-14	58	58	0	1
ex5	1.79688	1.15524e-12	0.00136574	41	24	3	0
ex6	0.169842	1.17245e-12	0.000102673	210	208	3	14
ex7a	0.014215	4.41092e-20	0.000223874	35	32	1	4
ex7b	0.0167153	1.73314e-09	0.00333938	30	11	3	1
ex8	0.00100105	5.48543e-10	0.0304644	6	6	3	0
ex9	0.377186	3.79968e-12	9.76957e-05	99	38	0	2
ex10	5.45798	2.06579e-11	0.000188703	144	68	3	3
ex11	2.21248	1.11022e-16	4.05871e-10	2	2	0	1
ex12	-8.8692	2.89159e-07	0.0287717	5	5	3	0
ex13a	280.033	7.42576e-22	0.0120482	87	30	3	7
ex13b	4.73911	3.93439e-10	0.000220962	94	76	3	28
ex14	171.82	5.05594e-08	0.0100249	74	26	3	2
ex15	0.741945	5.12312e-10	1.53683e-05	71	46	0	3
ex16	1.41162	1.65384e-06	0.0618398	26	7	3	1
ex17	0.39221	4.66903e-08	8.39156e-05	1130	1102	0	425
ex18	1.41615	2.78748e-05	0.61548	1951	233	3	49
ex19	2.45888	6.15481e-06	0.166108	47	16	3	5
ex20	0.239117	2.40611e-10	0.00263489	13	8	3	7
ex21	1.01514	3.34701e-05	0.0786098	52	10	3	12
ex22	4.16148	2.73332e-05	0.000684166	493	457	3	1372
ex23	0.343924	1.17809e-09	1.2783e-05	17	11	0	2
ex24	0.00521918	1.12724e-07	9.97519e-05	103	74	0	19
ex25	0.00521576	5.05662e-09	6.76208e-05	3	3	0	0
ex26	8.7019	2.54596e-10	1.66181e-06	20	18	0	9
ex27	3.84597e-11	2.02282e-10	6.90773e-05	16	16	0	1
ex28	0.0792513	2.51466e-19	0.00329332	16	16	1	9
ex29	0.71418	4.45043e-08	0.00626107	12	9	3	5
ex30	98.8305	2.63801e-11	0.000894739	80	64	3	105
ex31	3.70442	3.60822e-16	6.05971e-10	2	2	0	0
ex32	0.753991	4.95618e-19	0.000314628	95	94	3	5
ex33	6.04328e-06	2.89321e-20	7.01945e-05	83	83	0	5
ex34	0.268489	1.0407e-18	9.09173e-05	126	125	0	7
ex35	1.00633	4.44089e-16	9.72192e-05	109	109	0	52
ex36	0.0221832	8.32667e-16	8.67149e-05	36	36	0	9
ex37	0.375023	0	7.13755e-05	96	96	0	5
ex38	5152.36	5.55112e-16	0.124809	73	26	3	3
ex39	84.9583	2.16807e-06	4.78446	12160	1230	3	446
ex40	-1.87396	2.20012e-13	0.000434225	178	77	3	25
ex41	228.716	6.15689e-07	31.6152	153	35	3	48
ex42	228.433	1.26151e-10	0.00636685	37	21	3	36
ex43	228.52	7.93063e-10	0.000241872	10	7	1	12
ex44	0.00012382	8.11728e-05	0.00119983	18	8	3	1
ex45	0.000938975	1.18066e-18	9.37449e-05	9	4	0	0
ex46	3.35065	4.68017e-11	0.000146979	247	127	3	10
ex47	103.063	9.2909e-06	0.51149	112	38	3	2
ex48	0.460079	1.28192e-06	0.000118729	1416	437	3	29
ex49	4.12277e-05	4.50687e-10	1.36563e-05	3	3	0	0

`linfsqp` with finite difference gradients.

Prob.	f	|hg|	|DL|	#fun	#grad	exit	time (s)
ex50	0.668774	8.9507e-13	1.0551e-05	16	11	0	1
ex51	0.978342	3.41148e-11	6.62103e-06	176	98	0	8
ex52	0	2	0.2	1	1	-4	0
ex53a	0.119836	2.11995e-19	0.00024368	158	147	3	11
ex53b	11.9382	7.35583e-18	0.0375762	44	34	3	2
ex53c	119.289	4.76114e-19	0.205298	60	50	3	3
ex54	-353.716	9.00059e-10	0.764218	294	285	3	18
ex55	-7.85218e+87	807.419	3.00813e+98	31	14	-4	2
ex56	323.138	7.77156e-16	0.000892182	222	222	1	15
ex57	992.703	2.19302e-12	0.0183749	33	14	3	5
ex58	1765.82	4.17328e-11	9.53044	29	28	3	3
ex59	48.2338	3.23128e-12	2.48296e-06	9	8	0	0
ex60	1.07511	1.20817e-10	1.98848e-07	4	4	0	0
ex61dae	48.2335	1.52615e-10	0.000652212	16	13	3	1
ex62dae	6.40161	7.94476e-13	1.37948e-09	2	2	0	0
ex63dae	0.647776	8.36438e-08	0.000213576	433	431	3	42
ex64dae	0.706406	8.7918e-10	0.0192284	54	10	3	1
ex65a	1.78266	1.11022e-16	7.22692e-05	24	24	0	1
ex65b	1.65624	4.38066e-10	9.64904e-05	23	23	0	1
ex66a	1.26511	5.10703e-15	4.32214e-10	2	2	0	1
ex66b	4.67784	1.2452e-11	0.000105013	298	253	1	323
ex67	-0.608469	2.48369e-20	3.65168e-05	13	13	1	3
ex68dae	-0.608471	7.47842e-09	3.73887e-05	18	18	-4	8
ex69	-0.477692	1.44825e-18	9.68561e-05	53	53	0	3
ex70	-21.8227	6.69658e-25	0.00364251	944	942	3	2300
ex71	-0.573531	0	8.90343e-05	139	139	0	6
ex72	0.761599	0	3.85441e-05	9	9	0	1
ex73	24.255	1.85407e-13	0.00467865	32	31	3	6
ex74	0.0768077	4.44089e-16	9.14475e-11	2	2	0	1
ex75	2	9.48057e-08	1.6582e-07	98	96	0	30
ex76	2	9.48189e-08	5.1045e-06	177	115	2	41
ex77	0.144122	1.55532e-07	7.12183e-05	8	6	0	2

`linfsqp` with finite difference gradients, continued.

Prob.	f	|hg|	|DL|	#fun	#grad	exit	time (s)
ex1	1.69142e-05	0	8.11131e-05	221	221	0	2
ex2	14.0054	2.44249e-15	2.77546e-14	2	2	0	0
ex3	1	2.27596e-15	4.50322e-07	52	49	0	1
ex4	-0.25	4.44089e-16	1.00218e-14	58	58	0	1
ex5	1.79688	5.26412e-13	0.00136574	42	23	3	0
ex6	0.169843	9.25627e-20	0.000122067	211	209	3	15
ex7a	0.014215	1.43416e-19	0.000223863	35	32	1	4
ex7b	0.0167153	1.7335e-09	0.00333927	30	11	3	2
ex8	0.00100105	5.52116e-10	0.0304637	6	6	3	1
ex9	0.377186	5.78598e-12	9.76951e-05	99	38	0	2
ex10	5.45798	1.6075e-11	0.000188697	144	68	3	4
ex11	2.21248	1.11022e-16	1.58892e-14	2	2	0	0
ex12	-8.8692	2.89158e-07	0.0287715	5	5	3	0
ex13a	280.033	1.04324e-19	0.0120473	87	30	3	8
ex13b	4.73911	3.93763e-10	0.000220961	94	76	3	29
ex14	171.82	5.04969e-08	0.0100236	74	26	3	2
ex15	0.741945	5.12451e-10	1.53703e-05	71	46	0	4
ex16	1.41162	1.65382e-06	0.0618399	26	7	3	2
ex17	0.392386	6.47009e-08	0.000100156	1057	1036	3	420
ex18	1.41916	2.86136e-05	1.17874	1766	214	3	51
ex19	2.47679	0.000183937	0.491071	85	21	-2	8
ex20	0.239117	2.40639e-10	0.00263489	13	8	3	9
ex21	1.01514	3.347e-05	0.0786087	52	10	3	13
ex22	4.15942	2.04096e-07	0.000251617	562	519	3	1624
ex23	0.343924	1.17877e-09	1.27819e-05	17	11	0	3
ex24	0.00521918	1.12561e-07	9.97448e-05	103	74	0	21
ex25	0.00521576	5.04801e-09	6.76209e-05	3	3	0	0
ex26	8.7019	2.5803e-10	1.66471e-06	20	18	0	9
ex27	3.82392e-11	1.98922e-10	6.90772e-05	16	16	0	1
ex28	0.0792513	1.76098e-19	0.00329332	16	16	1	9
ex29	0.71418	4.45044e-08	0.00626109	12	9	3	5
ex30	98.8305	2.6474e-11	0.000894865	80	64	3	108
ex31	3.70442	3.60822e-16	5.58999e-14	2	2	0	0
ex32	0.753991	1.62195e-19	0.00031463	95	94	3	5
ex33	6.04781e-06	1.42486e-20	7.01945e-05	83	83	0	5
ex34	0.268489	1.46539e-18	9.0885e-05	126	125	0	7
ex35	1.00633	8.88178e-16	9.72192e-05	109	109	0	54
ex36	0.0221832	1.06859e-15	8.67703e-05	36	36	0	9
ex37	0.375023	0	7.13665e-05	96	96	0	5
ex38	5152.36	5.55112e-16	0.124858	74	26	3	3
ex39	84.9743	2.23977e-06	16190.1	8281	872	3	417
ex40	-1.87396	3.65994e-14	0.000558078	163	74	3	27
ex41	228.716	6.157e-07	31.6152	153	35	3	52
ex42	228.433	1.25699e-10	0.00636729	37	21	3	37
ex43	228.52	7.92408e-10	0.00024185	10	7	1	13
ex44	0.00012382	8.13806e-05	0.00119983	18	8	3	1
ex45	0.000938975	1.13948e-19	9.37451e-05	9	4	0	1
ex46	3.35065	4.46878e-11	0.000146979	247	127	3	12
ex47	103.063	9.29052e-06	0.511488	112	38	3	3
ex48	0.460308	1.26347e-06	0.000204106	1398	420	3	33
ex49	4.12277e-05	4.5106e-10	1.36563e-05	3	3	0	0

`linfsqp` with automatic differentiation.

Prob.	f	|hg|	|DL|	#fun	#grad	exit	time (s)
ex50	0.668774	1.51403e-12	1.05518e-05	16	11	0	0
ex51	0.978342	3.48932e-11	6.6152e-06	176	98	0	10
ex52	-0.166666	4.32048e-20	1.99055e-06	37	37	0	3
ex53a	0.119836	2.8823e-18	0.000243848	158	146	3	13
ex53b	11.9382	2.13125e-19	0.0382507	44	33	3	3
ex53c	119.289	1.26564e-18	0.227904	61	49	3	5
ex54	-353.75	1.22941e-30	2.42999	355	347	3	29
ex55	4.72844e-05	0.316318	630438	110	22	-2	5
ex56	323.138	5.06379e-28	0.00133406	214	214	3	15
ex57	969.855	4.21885e-15	0.00167481	76	62	3	20
ex58	1765.82	9.36553e-15	9.49778	30	28	3	3
ex59	48.2338	3.09454e-12	2.48272e-06	9	8	0	1
ex60	1.07511	1.20804e-10	1.98856e-07	4	4	0	1
ex61dae	48.2335	2.31027e-10	0.000652158	16	13	3	1
ex62dae	6.40161	1.44329e-15	1.99162e-14	2	2	0	0
ex63dae	0.648238	2.88545e-07	0.000338764	419	417	3	43
ex64dae	0.703967	2.22162e-09	0.0176336	57	10	3	2
ex65a	1.78266	1.77636e-15	7.22694e-05	24	24	0	1
ex65b	1.65624	4.38171e-10	9.64895e-05	23	23	0	0
ex66a	1.26511	5.10703e-15	3.00766e-14	2	2	0	1
ex66b	4.67784	6.0939e-11	0.000298139	297	252	3	314
ex67	-0.608469	9.93502e-20	3.65168e-05	13	13	1	3
ex68dae	-0.608471	7.46949e-09	3.73887e-05	18	18	-4	9
ex69	-0.477692	6.68174e-21	9.68562e-05	53	53	0	3
ex70	-21.8212	1.77971e-24	0.00125276	733	732	3	1612
ex71	-0.573531	0	8.90342e-05	139	139	0	7
ex72	0.761599	0	3.85441e-05	9	9	0	0
ex73	24.255	4.44089e-16	0.00467657	31	31	3	6
ex74	0.0768077	4.44089e-16	4.86356e-16	2	2	0	0
ex75	2	9.47433e-08	1.65992e-07	98	96	0	30
ex76	2	9.4721e-08	5.08261e-06	160	113	2	40
ex77	0.144122	1.55538e-07	7.12186e-05	8	6	0	3

`linfsqp` with automatic differentiation, continued.

[15] Burke, J. V. and Han, S. P., A Robust SQP Algorithm, *Mathematical Programming*, vol. 43, pp. 207–303, 1989.

[16] Buskens, C. and Maurer, H., SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control, *Journal of Computational and Applied Mathematics*, Vol. 120, pp. 85–108, 2000.

[17] Charalambous, C., On Conditions for Optimality of the Nonlinear l_1 Problem, *Mathematical Programming*, Vol. 17, pp. 123–135, 1979.

[18] Conn, A. R., Gould, N. I. M. and Toint, P. L., *Trust-Region Methods*, MPS-SIAM Series on Optimization, SIAM, 2000.

[19] Cuthrell, J. E. and Biegler, L. T., On the optimization of differential-algebraic process systems, *AIChE Journal*, **33**, 1257–1270, 1987.

[20] Dadebo, S. A. and McAuley, K. B., Dynamic optimization of constrained chemical engineering problems using dynamic programming. *Computers and Chemical Engineering*, **19**, pp. 513-525, 1995.

[21] Dissanayake, M., Goh, C. J., and Phan-Thien, N., Time-optimal trajectories for robot manipulators, *Robotica*, Vol. 9, pp. 131–138, 1991.

[22] Dueflhard, P. and Bornemann, F., *Scientific computing with ordinary differential equations*, Springer, 2002.

[23] Edge, E. R. and Powers, W. F., Optimal Control Problems with Bounded Controls and Singular Arcs, *Journal of Optimization Theory and Applications*, Vol. 20, pp. 455-479, 1976.

[24] Fabien, B. C., Indirect numerical solution of constrained optimal control problems with parameters, *Applied Mathematics and Computation*, Vol. 80, 43–62, 1996.

[25] Fabien, B. C., Observer-based feedback linearizing control of an electromagnetic suspension, *ASME Journal of Dynamic Systems Measurement and Control*, Vol. 118, 615–619, 1996.

[26] Fabien, B. C., Some tools for the direct solution of optimal control problems, *Advances in Engineering Software*, 1998, Vol. 29, 45–61.

[27] Fabien, B. C., Ramulu, M. and Tremblay, M., Dynamic Modelling and identification of a waterjet cutting machine, *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 9, pp 45–63, 2003.

[28] Fabien, B. C., A comparison of optimized stacked-ply flywheel designs that are based on different failure criteria, *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, September 2005.

[29] Fabien, B. C., The influence of failure criteria on the design optimization of stacked-ply composite flywheels, *Structural and Multidisciplinary Optimization*, Vol. 33, pp. 507–517, 2007.

[30] Fabien, B. C., Parameter optimization using the L_∞ exact penalty function and strictly convex quadratic programming problems, *Applied Mathematics and Computation*, doi:10.1016/j.amc.2007.09.028, 2008.

[31] Fabien, B. C., Implementation of a Robust SQP Algorithm, *Optimization Methods and Software*, doi:10.1080/10556780801996228, 2008.

[32] Fabien, B. C., Direct optimization of dynamic systems described by differential-algebraic equations, *Optimal Control Applications and Methods*, doi:10.1002/oca.838, 2008.

[33] `http://abs-5.me.wasington.edu/pub/dsoa/dsoa.zip`.

[34] Fabien, B. C., `dsoa`: A direct method for dynamic system optimization, *Journal of Computational and Applied Mathematics*, submitted.

[35] Fletcher, R., *Practical Methods for Optimization*, John Wiley, 1987.

[36] Gerdts, M. Direct shooting method for the numerical solution of higher-index DAE optimal control problems, *Journal of Optimization Theory and Applications*, Vol. 117, pp. 267-294, 2003.

[37] Gill, P. E., Jay, L. O., Leonard, M. W., Petzold, L. R. and Sharma, V., An SQP method for the optimal control of large-scale dynamical systems, *Journal of Computational and Applied mathematics*, Vol. 120, pp. 197–213, 2000.

[38] Goh, C. J. and Teo, K. L., MISER: a FORTRAN program for solving optimal control problems, *Adv. Eng. Software*, 1988, Vol. 10, 319–322.

[39] Goldfarb, D. and Idnani, A., A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs, *Mathematical Programming*, Vol. 27, pp. 1–33, 1983.

[40] Grippo, L., Lampariello, F. and Lucidi, S., A Nonmonotone Line Search Technique for Newton's Method, *SIAM Journal of Numerical Analysis*, Vol. 23, pp. 707–716, 1986.

[41] Hairer, E. and Wanner, G., *Solving ordinary differential equations II: Stiff and Differential-Algebraic problems*, 2nd. Edition, Springer, 1996.

[42] Hargraves, C. R. and Paris, S. W., Direct trajectory optimization using nonlinear programming and collocation, *AIAA Journal Guidance, Control and Dynamics*, **10**, 338–342, 1987.

[43] Jacobson, D. H., Differential Dynamic Programming Methods for Solving Bang-Bang Control Problems *IEEE Trans. AC*, Vol. 13, 1968.

[44] Jacobson, D. H. and Lele, M. M., A transformation technique for optimal control problems with state variable inequality constraints, *IEEE Trans. AC*, Vol. 14, pp. 457–460, 1969.

[45] Jacobson, D, Gershwin, S. and Lele, M., Computation of Optimal Singular Controls Jacobson, D. H. and Lele, M. M., *IEEE Trans. AC*, Vol. 15, pp. 67–73, 1970.

[46] Jung, E, Lenhart, S and Feng, Z., Optimal Control of Treatments in a Two-Strain Tuberculosis Model, *Discrete and Continuous Dynamical Systems-Series B*, Vol. 2, pp. 473–482, 2002.

[47] Junkins, J. L., and Turner, J. D., Optimal Continuous Torque Attitude Maneuvers, *Journal of Guidance, Control, and Dynamics*, Vol. 3, No. 3, pp. 210–217, 1980.

[48] Kirk, D. E., *Optimal Control Theory: An Introduction*, Prentice-Hall, 1970.

[49] Kraft, D., Algorithm 733: TOMP-Fortran Modules for Optimal Control Calculations, *ACM Trans. Math. Software*, 1994, Vol. 20, 262–307.

[50] Lasdon, L. S., Waren, A. D. and Rice, R. K., Interior penalty method for inequality constrained optimal control problems, *IEEE Trans. AC*, Vol. 12, pp. 388–395, 1967.

[51] Lang, J. and Verwer, J., ROS3P-An accurate third-order Rosenbrock solver designed for parabolic problems, *BIT*, Vol. 41, pp. 731–738, 2001.

[52] Lewis, F. L., *Applied Optimal Control & Estimation*, Prentice Hall, 1992.

[53] Logsdon, J. S. and Biegler, L. T., Accurate solution of differential-algebraic optimization problems, *Ind. Eng. Chem. Res.*, Vol. 28, pp. pp. 1628–1639, 1989.

[54] Lubich, Ch. and Roche, M., Rosenbrock methods for differential-algebraic systems with solution-dependent singular matrix multiplying the derivative, *Computing*, Vol. 43, pp. 325–342, 1990.

[55] Luus, R., Application of iterative dynamic to state constrained optimal control problems. *Hungarian Journal of Industrial Chemistry*, Vol. 19, pp. 245-254, 1991.

[56] Maurer, H. and Gillessen, W., Application of Multiple Shooting to the Numerical Solution of Optimal Control Problems with Bounded State Variables, *Computing*, Vol. 15, pp. 105–126, 1975.

[57] Miele, A., Gradient algorithms for the optimization of dynamic systems, *Control and Dynamic Systems*, pp. 1–52, 1980.

[58] Murali, B. N., Ebbesen, L. R. and Sebesta, H. R., Optimization of constrained dynamic systems using the sequential unconstrained minimization technique, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 29, pp. 319–322, 1972.

[59] Nagurka, M. L. and Yen, V., Fourier-based optimal control of nonlinear dynamic systems, *ASME Trans. Dynamic Systems, Measurement and Control*, Vol. 112, 17–26, 1990.

[60] Ostermann, A., Continuous Extensions of Rosenbrock-Type Methods, *Computing*, Vol. 44, 59–68, 1990.

[61] Panier, E. and Tits, A., Avoiding the Maratos Effect by Means of Nonmonotone Line Search I. General Constrained Problems, *SIAM Journal of Numerical Analysis*, Vol. 28, pp. 1183–1195, 1991.

[62] Pantoja, J. and Mayne, D. Q., Exact Penalty Function Algorithm with Simple Updating of the Penalty Parameter, *Journal Optimization Theory and Applications*, Vol. 69, pp. 441–467, 1991.

[63] Powell, M., A Fast Algorithm for Nonlinearly Constrained Optimization Calculations, in: Watson, G. A., ed., *Numerical Analysis, Dundee 1977*, Lecture Notes in Mathematics, No. 630, pp. 144–157, Springer, 1978.

[64] Pshenichnyi, B. N., *The Linearization Method for Constrained Optimization*, translated from the Russian by Stephen S. Wilson, Springer-Verlag, New York, 1994.

[65] Pytlak, R., *Numerical Methods for Optimal Control Problems with State Constraints* Lecture Notes in Mathematics, 1707, Springer-Verlag, 1999.

[66] Rall, L. B., Differentiation in Pascal-SC: Type GRADIENT, *ACM Transactions on Mathematical Software*, Vol. 10, pp. 161–184, 1984.

[67] Rosen, O. and Luus R., Global optimization approach to nonlinear optimal control, *J. Optimization Theory and Applications*, Vol.73, pp. 547–562, 1992.

[68] Sage, A. P. and White, C. C., *Optimum Systems Control*, Prentice-Hall, 1977.

[69] Sahba, M., Globally Convergent Algorithm for Nonlinearly Constrained Optimization Problems, *Journal Optimization Theory and Applications*, Vol. 52, pp. 291–309, 1987.

[70] Schulz, V., *Reduced SQP Methods for Large-Scale Optimal Control Problems in DAE with Application to Path Planning Problems for Satellite Mounted Robots* Ph.D. Thesis, University of Heidelberg, 1996.

[71] Schwartz, A., *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems* Ph.D. Thesis, University of California, Berkeley, 1996.

[72] Serban, R. and Petzold, L., COOPTA software package for optimal control of large-scale differentialalgebraic equation systems, *Mathematics and Computers in Simulation*, Vol. 56, pp. 187-203, 2001.

[73] Shampine, L. F., Implementation of Rosenbrock Methods, *ACM Transactions on Mathematical Software*, Vol. 8, pp. 93–113, 1982.

[74] Shimizu, K. and Ito, S., Constrained optimization in Hilbert space and a generalized dual quasi-newton algorithm for state-constrained optimal control problems, *IEEE Trans. AC*, Vol. 39, pp. 982–986, 1994.

[75] Snyman, J. A., Frangos, C. and Yavin, Y., Penalty function solutions to optimal control problems with general constraints via a dynamic optimization method, *Computers and Mathematics with Applications*, Vol. 23, pp. 47–55, 1992.

[76] Spellucci, P., A New Technique for Inconsistent QP Problems in the SQP Method, *Mathematical Methods of Operations Research*, Vol. 47, 355–400, 1998.

[77] Steinebach, G. and Rentrop, P., An Adaptive Method of Lines Approach for Modeling Flow and Transport in Rivers, in *Adaptive Method of Lines*, eds. A. Vande Wouwer, Ph. Saucez, W. E. Schiesser, Chapman and Hall, 2001.

[78] Stienbach, M., Bock, H. G. and Longman, R. W., Time-optimal extension or retraction in polar coordinate robots: A numerical analysis of the switching structure. In *Proc. AIAA Guid. Nav. and Cont. Conf.*, pp. 883–894, 1989

[79] Stryk, O. and Bulirsch, R., Direct and indirect methods for trajectory optimization, *Annals Oper. Research*, 1992, Vol. 37, 357–373.

[80] Teo, K. L., Goh, C. J. and Wong, K. H., *A unified computational approach to optimal control problems*, Longman Scientific, 1991.

[81] Tone, K., Revisions of Constraint Approximations in the Successive QP Method for Nonlinear Programming Problems, *Mathematical Programming*, Vol. 26, pp. 144–152, 1983.

[82] Xing, A. Q., Chen, Z. H., Wang, C. L. and Yao, Y. Y., Exact penalty function approach to constrained optimal control problems, *Optimal Control Applications and Methods*, Vol. 10, pp. 173-180, 1989.

[83] Zhao, Y., Bryson, A. E. and Slattery, R., Generalized gradient algorithm for trajectory optimization, *AIAA Journal of Guidance, Control and Dynamics*, Vol. 13, pp. 1166–1169, 1990.

[84] Zhang, J-L., and Zhang, X-S., Sequential Penalty Algorithm for Nonlinear Constrained Optimization, *Journal Optimization Theory and Applications*, Vol. 118, pp. 635–655, 2003.

[85] Zedan, H., Avoiding the exactness of the Jacobian matrix in Rosenbrock formulae, *Computers and Mathematics with Applications*, Vol. 18, 82–89, 1990.