On the Numerical Solution of the Adjoint Equation for Computing Sensitivities in Optimal Control

Kathrin Bernd

Bachelor's thesis
2015:K11

Lund University

Faculty of Science
Centre for Mathematical Sciences
Numerical Analysis

Centrum Scientiarum Mathematicarum

Lund University

Bachelor Thesis

On the Numerical solution of the Adjoint Equation for Computing Sensitivities in Optimal Control

Author:
Kathrin Bernd

Supervisor:
Claus Führer

July 1, 2015

Abstract

A short introduction to optimal control problems is presented. For these problems the calculation of sensitivity matrices with the method of adjoint equation is shown. The adjoint equations are derived with the corresponding initial values. At the end two examples are calculated.

First, I want to say thanks to my supervisor Prof. Claus Führer
for providing me with an interesting topic for my thesis.
Furthermore, I deeply appreciate the time you
took for answering all my questions
and for the discussions we had at the whiteboard.

Additionally I want to thank Peter and Adam for proof-reading
my thesis and teaching me some english along the way.

Finally, I want to say thanks to my family,
for supporting me during the writing of the thesis
and throughout my whole studies.

Contents

1 Introduction

In this thesis we look at the computation of sensitivity matrices with the adjoint method. Now the question is, what is sensitivity analysis or what are sensitivity matrices? Sensitivity analysis looks at small perturbations of the solution of a mathematical model based on how the input parameters can change the output. Here with a mathematical model a system of equations which are dependent on some variables and parameters is meant. The parameters describe some influences, which can happen in the mathematical model. The sensitivity analysis shows how big the influence of the change of the initial parameter is for the solution. So the sensitivity matrices show the influence of changing the initial conditions of the mathematical model that affect the solution of a system of ordinary differential equations (ODE) or differential algebraic equations (DAE). Applications of this sensitivity analysis can be found in optimization, in physical problems, in the field of the engineering and others.
Recent work [1] explains the forward computation method for the sensitivity matrices and how they can be calculated with CVODES.
But what is a real problem where the computation of sensitivities can be used? For example: *You have two water boxes where each of these boxes has a given volume and a controllable outflow rate of water. Then you are interested in the influence of a change in the initial outflow on the resulting outflow.* Here the sensitivity matrix describes the influences of changing the inflow.
In this paper the focus is on the adjoint sensitivity analysis for optimal control problems, the derivation of the adjoint equations and the computation of the sensitivity matrices. For this optimization problem we have a given objective function, which we want to minimize, from an optimal control problem. This objective function depends on a given parameter u, for which we want to compute the sensitivities. The sensitivities are given by the derivative of the objective function at the parameter u.

In the beginning a short overview for optimal control problems is given. After this the adjoint equation for an optimal control problem will be derived. Then the formula for the computation of sensitivity matrices will be derived. Finally, two examples are calculated.

2 Optimization Problem

This section gives a short introduction to optimal control problems, what an optimal control problem is and the definition of such a problem. The section ends with two special cases of optimal control problems. These cases are the parametrized optimal control problem and the initial value optimal control problem.

2.1 Optimal Control Problem

Optimal control problems are given, for example, *when we are searching for the right angle of holding a water towel to water a salad.* Here maybe some steps are needed to find the best angle that we catch the salad best, without loosing to much water. This means that in this case the changing of the angle is what we want to optimize. Such control problems are described by a system of ordinary differential equations

$$\dot{z} = f(z_1, \ldots, z_{n_z}, u_1, \ldots, u_{n_u}) \tag{1}$$

where z_i describes the process and u_i are the control parameters which describe the problem at the time t. In the example above the control parameter u is a vector of parameter values, rather then a vector valued function is normally called parameter identifications. In this thesis we consider these problems as (discrete) special cases of optimal control problems, where u is a vector valued function in time, which is given by

$$u_i = u_i(t) \text{ for } i = 0, \ldots, n$$

In this paper the idea behind optimal control problems is to look what the influence of changing the initial conditions has on the output. This means that the calculation of the sensitivities from the given optimization problem is of interest. The sensitivities are given by

$$\frac{\mathrm{d}}{\mathrm{d}u_i} g,$$

where g is the objective function of the optimal control problem with the constraints $\dot{z}$. So the definition of the optimal control problem is given by

Problem 1. *Let $\vartheta := [t_0, t_f] \subset \mathbb{R}$ be a compact interval with fixed time points $t_0 < t_f$ and let*

$$\begin{aligned} g &: \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \to \mathbb{R} \\ \varphi &: \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \to \mathbb{R} \\ f &: \vartheta \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_z} \end{aligned}$$

be sufficiently smooth functions.
Minimize

$$\begin{aligned} & g(z,u) = \int_0^T \varphi(z(t), u(t)) \\ & \text{with respect to } z : \vartheta \to \mathbb{R}^{n_z} \text{ and } u : \vartheta \to \mathbb{R}^{n_u} \\ & \text{such that} \begin{cases} \dot{z} = f(t, z, u) \\ z(0) = z_0 \end{cases} . \end{aligned}$$

2.2 Parametrized Optimal Control Problem

Here the focus is on a parametrized optimal control problem. This part of optimal control problems is a special case of this problem class. Here the control parameters are no longer functions which depend on time t, now these control parameter functions are discretized control parameters and are given as a vector $u \in \mathbb{R}^{n_z}$. From the discretization it follows that the initial conditions of the given system of differential equations (1) depend on the control parameters, which means that

$$z(t_0) = z_0(u).$$

Because the discretization of the control parameter functions has an influence on the system of differential equations and so it has an influences at the initial conditions from this system. From this the definition for parametrized optimal control problems follows

Problem 2. *Let $\vartheta := [t_0, t_f] \subset \mathbb{R}$ be a compact interval with fixed time points $t_0 < t_f$ and let*

$$\begin{aligned} g &: \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \to \mathbb{R} \\ \varphi &: \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \to \mathbb{R} \\ f &: \vartheta \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_z} \end{aligned}$$

be sufficiently smooth functions.
Minimize

$$\begin{aligned} & g(z) = \int_0^T \varphi(z(t)) \\ & \text{with respect to } z : \vartheta \to \mathbb{R}^{n_z} \text{ and } u \in \mathbb{R}^{n_z} \\ & \text{such that} \begin{cases} \dot{z} = f(t, z, u) \\ z(0) = z_0(u) \end{cases} \end{aligned}$$

2.2.1 Initial Value Optimal Control Problem

In the last part of this section the focus is on a more specific case of optimal control problems. Here we have an optimal control problem where only the initial conditions depend on the parameters u. The name of this special case is initial value optimal control problem. Mathematically speaking this means that the given system of differential equations is independent of the control parameters u_i. So the system looks like

$$\dot{z} = f(z_1, \ldots, z_n)$$

with the initial condition

$$z(t_0) = u,$$

where $u \in \mathbb{R}^{n_z}$. The definition of initial value optimal control problems is:

Problem 3. *Let $\vartheta := [t_0, t_f] \subset \mathbb{R}$ be a compact interval with fixed time points $t_0 < t_f$ and let*

$$\begin{aligned} g &: \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \to \mathbb{R} \\ \varphi &: \mathbb{R}^{n_z} \to \mathbb{R} \\ f &: \vartheta \times \mathbb{R}^{n_z} \to \mathbb{R}^{n_z} \end{aligned}$$

be sufficiently smooth functions.
Minimize

$$\begin{aligned} & g(z,u) = \int_0^T \varphi(z(t)) \\ & \text{with respect to } z : \vartheta \to \mathbb{R}^{n_z} \text{ and } u \in \mathbb{R}^{n_z} \\ & \text{such that} \begin{cases} \dot{z} = f(t,z) \\ z(0) = u \end{cases} \end{aligned}$$

3 Sensitivity Analysis

The goal is to derive the adjoint equations in order to compute the sensitivity matrices. Therefore this chapter starts with the derivation of adjoint equations and the sensitivity equation for optimal control problems. After this the adjoint equation for parametrized optimal control problems and initial value optimal control problems will follow. At the end of the chapter the idea of computing the sensitivities as a boundary value problem is given.

3.1 Sensitivity Analysis for Optimal Control Problem

The calculation of the sensitivity matrices depends on the objective function of Problem 1

$$g(z,u) = \int_0^T \varphi(z,u)dt$$

and its constraints. The sensitivities are the derivatives of the function $g(z)$ for the parameters $u(t)$. For this the construction of an auxiliary function will be used. The residual of the system will be multiplied with a multiplier λ. The idea behind this is that the multiplier times the residual will be equal to zero. From this the adjoint equation follows. The auxiliary function looks like

$$I(z,u) = \int_0^T \varphi(z,u)dt - \int_0^T \lambda^T(\dot{z} - f(z,u))dt$$

In this function $\dot{z} - f(z,u)$ describes the residual. If the residual is equal to 0 or λ^T makes the second integral equal to 0, then

$$\frac{\mathrm{d}}{\mathrm{d}u}g(u) = \frac{d}{du}I(u) = \int_0^T \varphi_z z_u + \varphi_u - \int_0^T \lambda^T(\dot{z}_u - f_z z_u - f_u)dt \tag{2}$$

In the next step we will use integration by parts for the last term of the equation

$$\int_0^T \lambda^T(\dot{x} - f_z z_u - f_u)dt = \lambda^T z_u|_0^T - \int_0^T (\dot{\lambda}^T z_u - \lambda^T f_z z_u - \lambda^T f_u)dt$$

Now we can look again at equation (2) and from the integration by parts of the last term in equation (2) it follows

$$\begin{aligned} \frac{d}{du}I(u) &= \int_0^T \varphi_z z_u + \varphi_u - \int_0^T (\dot{\lambda}^T z_u - \lambda^T f_z z_u - \lambda^T f_u)dt - [\lambda^T z_u]_0^T \qquad (3) \\ &\Leftrightarrow \\ \frac{\mathrm{d}}{\mathrm{d}u}I(x,u) &= \int_0^T \varphi_u - \lambda^T f_u - \int_0^T (\dot{\lambda}^T - \lambda^T f_z - \varphi_z)z_u dt - [\lambda^T z_u]_0^T \end{aligned}$$

In the next step the idea is that the second integral will be zero. This happens when,

$$\dot{\lambda}^T - \lambda^T f_z = -g_z \tag{4}$$

holds. So the adjoint equation for what is being sought is given by (4). The next interesting point is the last term of the equation (3), because this term

should also be zero so that the derivative of the auxiliary function is equal to the derivative of the objective function from problem (1). So from this term

$$[\lambda^T z_u]_{t=T} = 0$$

the assumption for the initial condition at the time T for the adjoint equation follows. For this we assume that $\lambda(T) = 0$, that the last term of equation (3) will be zero at T, because it is known that z_u can't be zero. So it follows for the computation of the sensitivities

$$\frac{\mathrm{d}}{\mathrm{d}u} g(u) = \int_0^T (\varphi_u - \lambda^T f_u) dt + [\lambda^T z_u]_{t=0}, \tag{5}$$

because the multiplier λ is chosen such that the second integral of equation (3) is zero. Here is it important to note that z_u at $t = 0$ is the sensitivity of the initial conditions with respect to u [2]. With this equation the calculation of the sensitivities can be done for an optimal control problem.

3.1.1 Sensitivity Analysis for a Parametrized Optimal Control Problem

The sensitivities of a parametrized optimal control problem can be calculated in an easier way. From Problem 2 it follows that, by a parametrized optimization problem, the initial conditions depend on the control parameter u,

$$z(t_0) = z_0(u).$$

This implies that in this case the system of differential equations derived at the parameter u gives the sensitivities, because the objective function is independent of the control parameter. So the sensitivities from the right-hand-side function are what is being sought. So the adjoint equation for these special systems is given by

$$\begin{aligned} \dot{\lambda}^T &= \lambda^T f_z + f_u \\ \lambda^T(t_0) &= z_{0_u} \end{aligned} \tag{6}$$

and then the sensitivities can be evaluated with equation (4). When we compare the adjoint equation (4) with the adjoint equation (6) then the term $\frac{\mathrm{d}}{\mathrm{d}z} g(z) = \frac{\mathrm{d}}{\mathrm{d}u} f(t, z, u)$, because we are interested in the sensitivities of the right-hand-side function.

3.1.2 Sensitivity Analysis for an Initial Value Optimal Control Problem

In the special case of an initial value optimal control problem, the initial values for the given system of ordinary differential equations are

$$z(t_0) = u$$

and the given right hand side of the system of ODEs is independent of the parameter u. Here the sensitivities are given from the initial conditions, because the initial value optimal control problem is a special case of the parametrized

optimal control problem. From this it follows that the last term of the adjoint equation of the parametrized problem is removed, because the right hand side $f(t,z)$ from the system of differential equations is independent of the control parameter u. So this term $f_u = 0$, from this we get the following system for the adjoint equation:

$$\begin{aligned}\dot{\lambda}^T &= \lambda^T f_z \\ \lambda(t_0) &= \mathrm{I}_i,\end{aligned}$$

where I_i stands for the i-th column of the identity matrix. In this special case the solution of the adjoint equation gives the sensitivities of the system.

3.2 Solve Sensitivity Analysis with Boundary Value Problem

The idea behind this section is to solve the given system of differential equations at the same time as the system of the adjoint differential equations. So the system of differential equations from the optimal control problem is given

$$\begin{aligned}\dot{z} &= f(t,z,u) \\ z(t_0) &= z_0\end{aligned}$$

and the system from the adjoint differential equations is given

$$\begin{aligned}\dot{\lambda}^T &= \lambda^T f_z - g_z \\ \lambda(T)^T &= 0.\end{aligned}$$

Let $h(z,\lambda,u) = \lambda^T f_z - g_z$ and

$$\dot{x} = \begin{pmatrix}\dot{z} \\ \dot{\lambda}^T\end{pmatrix}$$

with boundary values

$$x(t_0) = \begin{pmatrix}z_0 \\ ?\end{pmatrix} \text{ and } x(T) = \begin{pmatrix}? \\ 0\end{pmatrix}$$

The question marks are there, because there is no initial condition for $z(T)$ and $\lambda(t_0)$, but we can solve this problem with the approach of a boundary value problem for differential equations. The idea is to make a piecewise interpolation for the solution of the given system $\dot{x}$ at the time points t_i with the conditions

$$\begin{aligned}x_i(t_i) &= x_{i-1}(t_i) \\ \dot{x} &= \begin{pmatrix}f(z_i(t_{i+1}), u_i(t_{i+1})) \\ h(z_i(t_{i+1}), \lambda_i(t_{i+1}), u(t_{i+1}))\end{pmatrix}.\end{aligned}$$

Then the piecewise interpolated polynomials here of order 3 look like

$$x_i(t) = \begin{pmatrix}\sigma_3^i(t-t_i)^3 + \sigma_2^i(t-t_i)^2 + \sigma_1^i(t-t_i) + \sigma_0^i \\ a_3^i(t-t_i)^3 + a_2^i(t-t_i)^2 + a_1^i(t-t_i) + a_0^i\end{pmatrix}.$$

Now the idea is that the missing initial conditions can be evaluated. So a solution for the system of the adjoint differential equations is found, which can be used for computing the sensitivities with the given formula (5).
In this paper a different method for the computation of the solution of the two systems is used. This method will be explained in the next section.

3.3 Solve Sensitivity Analysis with Forward and Backward Integration

By the sensitivity analysis two systems of differential equations are given. The first one is the system of differential equations of the optimal control problem and the second one is the adjoint system of differential equations. Then the two systems be given by

$$\begin{aligned} \dot{y} &= f(t, y) \\ y(0) &= y_0 \\ t &\in [t_0, T] \end{aligned}$$

and

$$\begin{aligned} \dot{\lambda}^T &= h(\tau, y, u) \\ \lambda^T(T) &= 0 \\ \tau &\in [t_0, T]. \end{aligned}$$

In this case the solution of the second system of differential equations depend on the solution of the first system. For this a forward integration can be used to compute the solutions for the first system of differential equation and a backward integration, which includes the solution from the forward integration, gives the solution of the second system of differential equation. The method is shown in figure 2. Here the long green arrow describes the solution of the forward

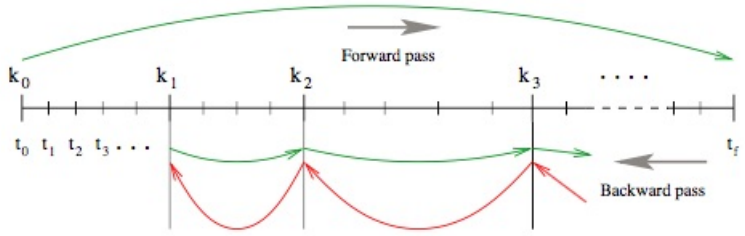

Figure 1: Illustration of the forward and backward integration during calculation of the adjoint system [4]

integration and the small right and green arrows stand for the forward and the backward integration steps which will be done. But it can happen that different time steps are used for the forward and the backward integration, for example when no fixed step size is used. Thus an interpolation over the solution from the forward integration is needed. The forward integration gives solutions $y_0, \ldots, y_N$ for $t_0, \ldots t_N = T$, but for the backward integration $y(\tau)$ is needed, because maybe different time points from the forward integration are used in the backward integration. So it is easy to think that

$$t_i < \tau \leq t_{i+1}$$

and then an interpolation over the given t_i and the given $y_i(t_i)$ can be done to evaluate the function $y(t)$ at the point τ. With this interpolation the backward integration can be computed and then the sensitivity equation (5) from above can be evaluated.

4 Examples

In this chapter are two examples for the computation of sensitivity matrices. The first example is about a system of two water boxes with a controllable outflow rate and the second example is a Sundials test example. For the first example only the analytic calculation is shown and for the second example the analytic calculation combined with the numerical calculation is given. For the numerical calculation a program with forward-backward integration is used.

4.1 System of two Water Boxes

Here we consider a system of two water boxes, where $x_1(t)$ and $x_2(t)$ stand for the volume of water in the reservoirs and $v_1(t)$ and $v_2(t)$ stand for the outflow rate of water for each reservoir, respectively, at time t.

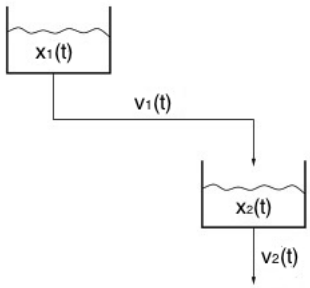

Figure 2: System of two water boxes with controllable outflow rate [3]

The system describes the influence of an controllable outflow from reservoir 1 in reservoir 2 for the outflow of reservoir 2. For this the optimal control problem is given by

Problem 4. *Let* $\vartheta = [0, 10]$.

$$Minimize_{v_1,v_2}$$

$$g(x_1, x_2, v_1, v_2) = -\int_0^{10} (10 - t)v_1(t) + tv_2(t)dt$$

such that

$$\begin{cases} \dot{x}_1(t) = -v_1(t) \\ \dot{x}_2(t) = v_1(t) - v_2(t) \\ x_1(t_0) = x_{1_0} \\ x_2(t_0) = x_{2_0} \\ x_1(t), x_2(t) \leq 0 \\ v_1(t) = v_2(t) \in [0, 1] \end{cases}$$

In Problem 4 the control parameters are given by v_1, v_2 and the system is independent of the states, it depends only on the control parameters. Now by the formulae from the section before it follows

$$f(x, \dot{x}, t; u_1, u_2) = \begin{pmatrix} \dot{x_1}(t) + v_1 \\ \dot{x_2}(t) - v_1 + v_2 \end{pmatrix}$$

The adjoint equation for this problem is given by

$$\dot{\lambda}(t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$\text{with } \lambda(T) = 0$$

Now we can compute the sensitivities for this problem with

$$\frac{\mathrm{d}}{\mathrm{d}u} I(x, u) = \int_0^T (\varphi_u - \lambda^T f_u) dt + (\lambda^T x_u)|_0$$

The last term $(\lambda^T x_u)|_0 = 0$, because we have no sensitivity for the initial condition. So it follows that the equation reduces to

$$\frac{\mathrm{d}}{\mathrm{d}u} I(x, u) = \int_0^T (\varphi_u - \lambda^T f_u) dt.$$

For this we need the solution of the adjoint equation, but it is easy to see that the solution is given by

$$\lambda(t) = 0,$$

because of the initial condition $\lambda(T) = 0$. So now we have all the information we need for the calculation of the sensitivity matrix.

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}v} I(x, v) &= \int_0^{10} \varphi_u dt \\ &= \int_0^{10} \begin{pmatrix} -10 + t, & t \end{pmatrix} dt \\ &= \begin{pmatrix} -10t + \frac{1}{2}t^2, & \frac{1}{2}t^2 \end{pmatrix} |_0^{10} \\ &= \begin{pmatrix} -50, & 50 \end{pmatrix} \end{aligned}$$

This means that the solution of the integral will be changed. In the sense that the initial outflow v_1 changes then the first term of the integral $(10 - t)v_1(t)$. In this case that these term will be multiplied with a factor of -50 and when the initial outflow v_2 is changed then the last term of the integral $tv_2(t)$ will be multiplied with a factor of 50.

4.2 Sundials test Example

This is an example of an initial value problem. In this problem the initial values depend on the sensitivity u. The given problem is

$$\begin{aligned} \dot{y_1} &= -0.0712y_1 + 0.0111y_2 + 0.039y_3 + 49.3 \\ \dot{y_2} &= 0.0111y_1 - 0.0286y_2 \\ \dot{y_3} &= 0.039y_1 - 0.000035y_3 \end{aligned}$$

with initial condition $y(0) = u$ [5]. So the computation of the sensitivities can be done with

$$\frac{\mathrm{d}}{\mathrm{d}u} I(u) = \int_0^T \varphi_u - \lambda^T f_u dt \tag{7}$$

and the adjoint equation

$$\dot{\lambda}^T = \lambda^T f_y - \varphi_y y_u$$

So f is given by

$$f(y, \dot{y}, t) = \begin{pmatrix} \dot{y}_1 + 0.0712y_1 - 0.0111y_2 - 0.039y_3 - 49.3 \\ \dot{y}_2 - 0.0111y_1 + 0.0286y_2 \\ \dot{y}_3 - 0.039y_1 + 0.000035y_3 \end{pmatrix}$$

and f_y is given by

$$f_y = \begin{pmatrix} 0.0712 & -0.0111 & -0.039 \\ -0.0111 & 0.0286 & 0 \\ -0.039 & 0 & 0.000035 \end{pmatrix}$$

Now for this example the question is: what is the function φ_y? The function is given by

$$\varphi_y = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix},$$

because it is searched for the sensitivities of the inital conditions from the system, but here is only a example given for the first vector. Here it can be choosen every row of the identity matrix for the first term of φ_y. So the sensitivities for the these system can be calculated with the equation (5).

4.2.1 Numerical Solution

The problem will be solved with a program which is written in Python. Here the package Assimulo is used. Assimulo can be used for solving differential equations. For this, a right-hand-side function is given, which will be transformed in an explicit problem and for the simulation a CVODE solver is used.

In this example Assimulo is used to calculate the forward integration and the backward integration, where in the right-hand-side function the interpolation of the forward solution is done. In the next lines the code for the forward integration will be explained and then the code for the backward integration will be explained.
The program of the forward integration starts with the right-hand-side function. This function contains the system of differential equations from the optimal control problem. Here y_1, y_2, y_3 are written as a vector y. By the definition of the right-hand-side function this function will be transformed into an explicit problem, with the function called Explicit_Problem. This function needs as an input the right-hand-side function and the initial values for the given system, here also

a name for the problem can be given. For this example $y_0 = \begin{pmatrix} 0, & 0, & 0 \end{pmatrix}$. After the problem is discretized a CVode solver is used to solve the problem. This works with the function called CVode and finally this problem will be simulated, for this the function called simulate is used.
From this program the solution of the forward integration follows: In this plot

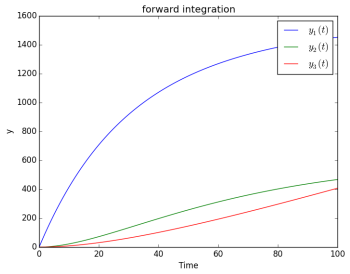

Figure 3: Solution of the forward integration

the solution of the problem is given, where the parameter is chosen as $u = 0$. This means that the solution without the influence of the parameter is shown. The program for the backward integration is the same, only two small differences are in it. The first difference is that in the right-hand-side function the solution of the forward integration will be interpolated, with a degree three polynomial. The second difference is that we need a special variable that the interpolation in backward time can be done. This variable is called *backward* and it has to be true. From this the following solution for the adjoint equations are shown in figures 4-6.

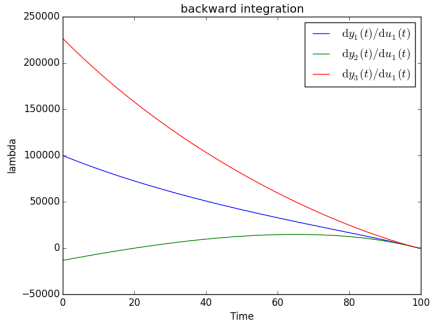

Figure 4: Solution of the backward Integration where $\frac{\mathrm{d}y}{\mathrm{d}y_{0_1}}$

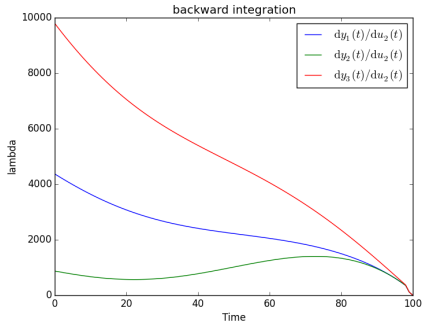

Figure 5: Solution of the backward Integration where $\frac{\mathrm{d}y}{\mathrm{d}y_{0_2}}$

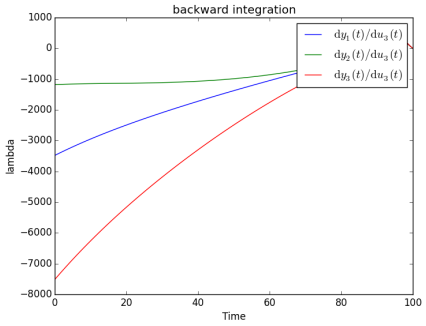

Figure 6: Solution of the backward integration where $\frac{\mathrm{d}y}{\mathrm{d}y_{0_3}}$

In these three plots, the influence of the parameter at the time is shown. This means for example that when the parameter u_1 is chosen for which the computation of the sensitivities should be done. Then the influence for the solution of the system when the parameter u_1 is changed at $t = 0$ is, for example, that the solution of y_3 will be multiplied with a factor of 225000, the solution of y_1 will be multiplied with a factor of 100000 and the solution of y_2 will be multiplied with a factor of -10000. This means that in the plot the factor for changing parameters at a given time point is shown, which means that the solution of this system will be multiplied with this factor if the parameter is changed.
If the influence of the sensitivities is searched for a given time interval, then the computation can be done with the following formula

$$\frac{\mathrm{d}}{\mathrm{d}u} I(u) = \int_0^T \lambda^T \cdot I_3 dt.$$

In the next step the solution of this integral is computed where $T = 100$, from this the following sensitivity matrix follows

$$\frac{\mathrm{d}}{\mathrm{d}u} I(u) = \begin{pmatrix} 148389.47134845212 & 148569.16737826829 & 149028.31931999006 \\ -19327.387537068535 & -19149.415402086557 & -18698.284831337056 \\ 335924.48444081156 & 336109.53173379973 & 336592.77106828295 \end{pmatrix},$$

here in the first collum the sensitivities $\frac{\mathrm{d}}{\mathrm{d}y_{1_0}} y = u_1$ are given, in the second collum the sensitivities for $\frac{\mathrm{d}}{\mathrm{d}y_{2_0}} y = u_2$ and in the last collum for $\frac{\mathrm{d}}{\mathrm{d}y_{3_0}} y = u_3$. So in the matrix are the factors shown, which multiplie the solution at $T = 100$, when the inital parameters will be changed.

5 Conclusion

In this paper, first a short introduction for optimal control problems was given. For these problems the adjoint equations which for the regular problem looks like

$$\begin{aligned} \dot{\lambda}^T - \lambda^T f_z &= -g_z \\ \lambda(T) &= 0 \end{aligned}$$

and the formula for the computation of the sensitivity matrices

$$\frac{\mathrm{d}}{\mathrm{d}u} g(u) = \int_0^T (\varphi_u - \lambda^T f_u) dt + [\lambda^T f_z z_u]_{t=0}$$

were derived. We found out that for initial value optimization problems the calculation of the sensitivities is equal to the integral over the solution of their adjoint differential equation system, because there the sensitivities depend on the initial conditions of the system of ODEs.

References

[1] Rien Quirynen, Milan Vulkov, Moritz Diehl,*Auto Generation of Implicit Integrators for Embedded NMPC with Microsecond Sampling Times*

[2] Yang Cao, Shengtal Li, Linda Petzold,Radu Serban, *Adjoint Sensitivity Analysis fir Differntial-Algebraic Equations: The Adjoint DAE System and its Numerical Solution*

[3] M.Gerdts, *Optimal Control of ODES and DAES*, 1. Edition (2012), De Gruyter

[4] Radu Serban, Alan C. Hindmarsh,*CVODES: An ODE Solver with Sensitivity Analysis Capabilities*,2003

[5] *http://sundials.2283335.n4.nabble.com/Forward-sensitivities-for-initial-conditions-td3239724.html*

[6] *The Mathematical Theory of Optimal Processes* 3.Edition (1986) ,Gordon and Breanch Scince Publishers S.A.

6 Appendix

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jun 17 16:27:59 2015

@author: kathrin
"""

from scipy import *
import numpy as N
import pylab as P
from assimulo.problem import Explicit_Problem
from assimulo.solvers import CVode
import warnings
from numpy.polynomial.hermite import hermfit

def run_example(with_plots=True):
    """ forward integration to compute the sensitivity matrices in the next step"""
    #Define the rhs
    def f(t, y):
        y1,y2,y3 = y
        k01 = 0.0211
        k02 = 0.0162
        k21 = 0.0111
        k12 = 0.0124
        k31 = 0.0039
        k13 = 0.000035
        b1 = 49.3

        yd_0 = -(k01+k21+k31)*y1+k12*y2+k13*y3+b1
        yd_1 = k21*y1-(k02+k12)*y2
        yd_2 = k31*y1-k13*y3

        return N.array([yd_0,yd_1,yd_2])


    #Define an Assimulo problem
    forward_exp_mod = Explicit_Problem(f, y0=array([0,0,0]), name = r'forward integration')

    #Define an explicit solver
    forward_exp_sim = CVode(forward_exp_mod) #Create a CVode solver
    #Sets the parameters
    #forward_exp_sim.inith=-1.e-15
    forward_exp_sim.iter = 'Newton' #Default 'FixedPoint'
    forward_exp_sim.discr = 'BDF' #Default 'Adams'
    forward_exp_sim.atol = [1e-6] #Default 1e-6
    forward_exp_sim.rtol = 1e-6 #Default 1e-6

    #Simulate
    t1, x1 = forward_exp_sim.simulate(100,100) #Simulate 5 seconds
    #Plot
    P.figure()
```

```
52      if with_plots:
53          legend_text=r"${}$"
54          P.plot(t1, x1)
55          P.title(forward_exp_mod.name)
56          P.legend((legend_text.format('y_1(t)'),
57                    legend_text.format('y_2(t)'),
58                    legend_text.format('y_3(t)')))
59          P.ylabel('y')
60          P.xlabel('Time')
61          P.show()
62
63      return forward_exp_mod, forward_exp_sim
64
65  if __name__=='__main__':
66      forward_mod,forward_sim = run_example()
67
68  def run_example2(with_plots=True):
69
70      """ backward integration to compute the sensitivities of dx/dp_1"""
71      t=N.array(forward_sim.t_sol)#time of the forward integratio
72      u=N.array(forward_sim.y_sol)#solution points of the forward integration
73
74      def lambda_xy(tau,lambda1):
75          upp_index = lambda tau: argmax(array(t)>tau)#gives the upper index
76          index=upp_index(tau)#gives us the upper index for the t
77          try:
78              u_pol=polyfit(t[index:index+3],u[index:index+3,i],3)
79          except IndexError:
80              u_pol=polyfit(t[index-3:index],u[index-3:index,i],3)
81
82          #adjoint equation
83          lambda_xdot=0.0712*lambda1[0]-0.0111*lambda1[1]-0.039*lambda1[2]-polyval(u_pol,u[0,i])
84          lambda_ydot=+0.0111*lambda1[0]+0.0286*lambda1[1]-polyval(u_pol,u[0,i])
85          lambda_zdot=-0.039*lambda1[0]+0.000035*lambda1[2]-polyval(u_pol,u[0,i])
86
87
88          return N.array([lambda_xdot,lambda_ydot,lambda_zdot])
89
90      #Define an Assimulo problem
91      exp_mod = Explicit_Problem(lambda_xy, y0=array([0,0,0]),t0=100, name = r'backward integration')
92
93      #Define an explicit solver
94      exp_sim = CVode(exp_mod) #Create a CVode solver
95      #Sets the parameters
96      exp_sim.backward=True
97      #exp_sim.inith=-1.e-5
98      exp_sim.iter = 'Newton' #Default 'FixedPoint'
99      exp_sim.discr = 'BDF' #Default 'Adams'
100     exp_sim.atol = [1e-7] #Default 1e-6
101     exp_sim.rtol = 1e-6 #Default 1e-6
102
103     #Simulate
104     tau1,x1 = exp_sim.simulate(0,100) #Simulate 5 seconds
105     #Plot
```

```
106     P.figure()
107     if with_plots:
108         legend_text=r"$\mathrm{{d}}{}/\mathrm{{d}}{}$"
109         P.plot(tau1,x1)
110         P.title(exp_mod.name)
111         P.legend((legend_text.format('y_1(t)','u_1(t)'),
112                 legend_text.format('y_2(t)','u_1(t)'),
113                 legend_text.format('y_3(t)','u_1(t)')))
114         P.ylabel('lambda')
115         P.xlabel('Time')
116         P.show()
117     return exp_mod, exp_sim
118 for i in range(3):
119     if __name__=='__main__':
120         with warnings.catch_warnings():
121             warnings.filterwarnings('ignore',category=FutureWarning)
122             warnings.filterwarnings('ignore',category=RankWarning)
123             mod,sim= run_example2()
```

Bachelor's Theses in Mathematical Sciences 2015:K11
ISSN 1654-6229

LUNFNA-4005-2015

Numerical Analysis
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/