

Numerical Methods and Software for Sensitivity Analysis of Differential-Algebraic Systems

Timothy Maly¹,

*McHugh, Freeman and Associates, Inc., 20975 Swenson Drive, Suite 400,
Waukesha, WI 53186*

Linda R. Petzold²,

*Department of Computer Science and Army High Performance Computing
Research Center, University of Minnesota, Minneapolis MN 55455*

Abstract

In this paper we present some new algorithms and software for sensitivity analysis of differential-algebraic equation (DAE) systems. The algorithms have several novel features which are described and analyzed. The codes, which are extensions of DASSL and DASPK, are easy to use, highly efficient, and well-suited for large-scale problems.

1 Introduction

Many engineering and scientific problems are described by systems of differential-algebraic equations (DAEs). Parametric sensitivity analysis of the (DAE)

¹The work of this author was partially supported by DOE contract number DE-FG02-92ER25130 and by the Minnesota Supercomputer Institute and by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

²The work of this author was partially supported by DOE contract number DE-FG02-92ER25130 and by NIST contract number 60NANB2D1272 and by ARO contract number DAAL03-92-G-0247 and by Exxon Research and Engineering Co..

model may yield information useful for parameter estimation, optimization, process sensitivity, model simplification and experimental design. Consequently, algorithms which perform such an analysis in an efficient and rapid manner are invaluable to researchers in many fields. In this paper we present two such codes: DASSLSO and DASPKSO. The codes are modifications of the DAE solvers DASSL and DASPK ([2]). The algorithms used in these codes have several novel features. They make use of an adaptive difference directional derivative approximation to (or alternatively a user supplied expression for) the sensitivity equations. The ability to adapt the increment as time progresses is important because the solution and sensitivities can sometimes change drastically. The sensitivity equations are solved simultaneously with the original system, yielding a nonlinear system at each time step. We will show via theory and numerical results that the resulting Jacobian can be approximated by a block diagonal matrix while retaining rapid Newton convergence, and that a block-diagonal preconditioner is highly effective in DASPK. The new codes are easy to use, highly efficient, and well-suited for large-scale problems.

We assume that the reader is already familiar with the codes DASSL [2] and DASPK [3] and the algorithms used in these codes, in order to concentrate on the extensions for sensitivity analysis.

2 Sensitivity Analysis

To illustrate the basic approach, consider the general DAE system with parameters,

$$F(t, y, y', p) = 0, \quad y(0) = y_0$$

where $y \in \mathfrak{R}^{n_y}$, $p \in \mathfrak{R}^{n_p}$. Here, n_y and n_p are the dimension and the number of parameters in the original DAE system, respectively. Sensitivity analysis entails finding the derivative of the above system with respect to each parameter. This produces an additional $n_s = n_p \cdot n_y$ sensitivity equations which, together with the original system, yields

$$\begin{aligned} & F(t, y, y', p) = 0 \\ & \frac{\partial F}{\partial y} s_i + \frac{\partial F}{\partial y'} s_i' + \frac{\partial F}{\partial p_i} = 0, \quad i = 1, \dots, n_p, \end{aligned} \tag{1}$$

where $s_i = \frac{dy}{dp_i}$. Defining $Y = [y, s_1, \dots, s_{n_p}]^T$ and $\mathbf{F} = [F(t, y, p), \frac{\partial F}{\partial y} s_1 + \frac{\partial F}{\partial y'} s_1' + \frac{\partial F}{\partial p_1}, \dots, \frac{\partial F}{\partial y} s_{n_p} + \frac{\partial F}{\partial y'} s_{n_p}' + \frac{\partial F}{\partial p_{n_p}}]^T$ the combined system can be rewritten

as

$$\mathbf{F}(t, Y, Y', p) = 0, \quad Y(0) = \begin{pmatrix} y_0 \\ \frac{dy_0}{dp_1} \\ \vdots \\ \frac{dy_0}{dp_{n_p}} \end{pmatrix}.$$

We note that the initial conditions for this DAE system must be chosen to be consistent.

Approximating the solution to the combined system by a numerical method, for example the implicit Euler method with stepsize h , yields the nonlinear system

$$G(Y_{n+1}) = \mathbf{F}(t_{n+1}, Y_{n+1}, \frac{Y_{n+1} - Y_n}{h}, p) = 0. \quad (2)$$

Newton's method for the nonlinear system produces the iteration

$$Y_{n+1}^{(k+1)} = Y_{n+1}^{(k)} - \mathbf{J}^{-1}G(Y_{n+1}^{(k)}),$$

where

$$J = \begin{pmatrix} J & & & & \\ J_1 & J & & & \\ J_2 & 0 & J & & \\ \vdots & \vdots & \ddots & \ddots & \\ J_{n_p} & 0 & \dots & 0 & J \end{pmatrix} \quad (3)$$

and $J = \frac{1}{h} \frac{\partial F}{\partial y'} + \frac{\partial F}{\partial y}$, $J_i = \frac{\partial J}{\partial y} s_i + \frac{\partial J}{\partial p_i}$.

A number of codes for ODEs and DAEs solve the sensitivity system (1), or its special case for ODEs, directly (see [4,11]). If the partial derivative matrices are not available analytically, they are approximated by finite differences. The nonlinear system is usually solved by a staggered scheme, where the first block is solved for the state variables y via Newton's method, and then the block-diagonal linear system for the sensitivities s is solved at each time step.

2.1 Directional derivative sensitivity approximation

Although the direct solution of (1) is successful for many problems, there are a number of properties of this approach which are not advantageous in the

context of DASSL/DASPK. For efficiency, DASSL was designed to use its approximation to the system Jacobian over as many time steps as possible. However, sensitivity implementations using the above described staggered approach must re-evaluate this Jacobian at every step in order to ensure an accurate approximation to the sensitivity equations. If this matrix has been approximated via finite differences, which is most often the case, large errors may be introduced into the sensitivities. In addition, the staggered scheme for solving the nonlinear system is not advantageous for parallel computation [10]. To eliminate these problems, we focus on approximating the sensitivity system (1) directly, rather than via the matrices $\partial F/\partial y$, $\partial F/\partial y'$, and $\partial F/\partial p$. In the simplest case, the user can specify directly the residual of the sensitivity system at the same time as the residual of the original system. Eventually, we intend to incorporate the automatic differentiation software ADIFOR [1] for this purpose. Alternatively, we can approximate the right hand side of the sensitivity equations via a directional derivative finite difference approximation. As an example, define

$$s_i = \frac{dy}{dp_i}$$

and solve

$$\frac{F(t, y + \delta_i s_i, y' + \delta_i s'_i, p + \delta_i e_i) - F(t, y, y', p)}{\delta_i} = 0, \quad (4)$$

where δ_i is a small scalar quantity, and e_i is the i^{th} unit vector. Proper selection of the scalar δ_i is crucial to maintaining acceptable round-off and truncation error levels, and will be discussed in greater detail in section 2.2. By Taylor's theorem, it is easily seen that (4) approximates the i^{th} sensitivity in (1) with an error of order $O(\delta_i)$. We can also approximate the sensitivity system via a second order central difference with an error of order $O(\delta_i^2)$. Using either of the latter two strategies, approximations to the sensitivity equations are generated at the same time as the residual of the original system, via n_p additional calls to the user function routine (In DASSL/DASPK, this is the routine RES). The resulting system is discretized by a numerical method (in DASSL/DASPK this is the BDF method of orders 1-5), yielding an iteration matrix of the form (3).

In general, for a Newton or Newton-Krylov iteration, one should be able to approximate the iteration matrix \mathbf{J} by its block diagonal part provided that the error matrix for the Newton/modified Newton steps is nilpotent. To illustrate this idea, consider the problem formulation (2)

$$G(Y) = 0$$

and apply a Newton step

$$Y^{(k+1)} = Y^{(k)} - \hat{\mathbf{J}}^{-1}G(Y^{(k)}), \quad (5)$$

where the Newton matrix \mathbf{J} has been approximated by its block-diagonal part, $\hat{\mathbf{J}}$. The true solution Y^* satisfies

$$Y^* = Y^* - \hat{\mathbf{J}}^{-1}G(Y^*). \quad (6)$$

Subtracting (6) from (5) and defining $e^k = Y^{(k+1)} - Y^*$, the iteration errors satisfy

$$Y^{(k+1)} - Y^* = e^{k+1} \approx e^k - \hat{\mathbf{J}}^{-1}\mathbf{J}e^k = (I - \hat{\mathbf{J}}^{-1}\mathbf{J})e^k.$$

The error matrix has the form

$$I - \hat{\mathbf{J}}^{-1}\mathbf{J} = \begin{pmatrix} 0 & & & & \\ J^{-1}J_1 & 0 & & & \\ J^{-1}J_2 & 0 & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ J^{-1}J_{n_p} & 0 & \dots & 0 & 0 \end{pmatrix}$$

In section 3, we show that because this matrix is nilpotent, the Newton iteration achieves 2-step quadratic convergence for nonlinear problems.

2.2 Problem scaling and increment selection

The increment selection is critical to the success of the finite difference approximation of the sensitivity equations. Here we describe and motivate the adaptive strategy used in DASSLSO.

In the absence of scaling problems, one might choose the increment based on the size of the parameter. For example,

$$\delta_i = \sqrt{u}|p_i|, \quad (7)$$

where u is the unit roundoff error, perturbs half the digits of p_i . This type of strategy usually works provided p_i is not near zero. An important consideration for any increment selection strategy is that it should scale with p_i . That is, if the user were to change the units of the parameter and solve instead for $\hat{p}_i = cp_i$, the increment should scale as well. We will assume in any such

scaling that the error tolerances corresponding to $s_i = \frac{dy}{dp_i}$ have also been scaled appropriately. Further indication of the scale of the problem is the relative size of s_i to y . Recall that the directional difference increments y to obtain $y + \delta_i s_i$. Using a rule of thumb which suggests perturbing half the digits of y ([6]), this would mean roughly that

$$\delta_i \|s_i\|_2 = \sqrt{u} \|y\|_2,$$

and hence

$$\delta_i = \sqrt{u} \frac{\|y\|_2}{\|s_i\|_2}. \quad (8)$$

Note also that units of s are the same as units of y divided by units of p . This yields a δ_i with units which are compatible with (7) and may in some instances yield additional information which can be used by the perturbation selection strategy. However, (8) will fail if $\|s_i\|_2 = 0$ as it often is at the initial time. There is also a potential scaling problem for y , i.e. if $\|y\|_2$ is near zero (for an illustration of scaling problems see the Batch-Reactor numerical result in section 5.3). Assuming that the user has scaled the error tolerances appropriately, we modify (8) to prevent such failures and scaling difficulties by using

$$\delta_i = \sqrt{u} \|v_i\|_2, \quad i = 1, \dots, n_p, \quad (9)$$

where

$$(v_i)_j = \frac{RTOL_j \cdot |y_j| + ATOL_j}{RTOL_{in_y+j} \cdot |s_{ij}| + ATOL_{in_y+j}} = \frac{WT_j}{WT_{in_y+j}}, \quad j = 1, \dots, n_y \quad (10)$$

Combining (9) and (10), the strategy used in DASSLSO is given by

$$\delta_i = \Delta \cdot \max(|p_i|, \|v_i\|_2). \quad (11)$$

In our experience, it is possible that for some very nasty problems $\Delta = \sqrt{u}$ may be too small. The user is given the option to change this value. It should be noted that for many well-scaled problems using a scalar RTOL, the perturbation $\Delta = \sqrt{RTOL}$ may be appropriate. This implies that we are perturbing roughly half of the (locally) accurate digits of the numerical solution, hence the error in the second-order difference to the sensitivities is $O(\Delta^2) = O(RTOL)$.

2.3 Sensitivity Analysis of Derived Quantities

In addition to the sensitivity analysis modifications to DASSL and DASPK, a stand alone routine (SENSD) which performs a sensitivity analysis of a derived quantity has been constructed. This routine approximates the analytic sensitivity equations by finite differencing the derived quantity $Q(t, y, y', p)$ ($p \in \mathbb{R}^{n_p}$, $y \in \mathbb{R}^{n_y}$ and $Q \in \mathbb{R}^{n_q}$), using

$$\frac{dQ(t, y, y', p)}{dp_i} = \frac{\partial Q}{\partial y} \frac{dy}{dp_i} + \frac{\partial Q}{\partial y'} \frac{dy'}{dp_i} + \frac{\partial Q}{\partial p_i}.$$

Expanding $Q(t, y, y', p)$ in a Taylor's series about y results in

$$Q(t, y + \delta_i s_i, y' + \delta_i s_i', p + \delta_i e_i) = Q(t, y, y', p) + \delta_i \frac{\partial Q}{\partial y} s_i + \delta_i \frac{\partial Q}{\partial p_i} + \delta_i \frac{\partial Q}{\partial y'} s_i' + O(\delta_i^2)$$

so that

$$\frac{dQ(t, y, y', p)}{dp_i} \approx \frac{Q(t, y + \delta_i s_i, y' + \delta_i s_i', p + \delta_i e_i) - Q(t, y, y', p)}{\delta_i}.$$

This, of course, is one of many possible finite difference schemes which can be used. In the code, central differencing is also an option. The routine SENSD can be called after a successful return from a call to DASSLSO or DASPKSO and must be provided with a function (DRVQ) which defines the derived quantity Q .

3 Convergence of the Sensitivity Iteration

In this section we analyze the convergence of the Newton method for the combined system (1), using the Jacobian approximation described in section 2.1. We will show that the method is 2-step quadratic convergent for full Newton, and convergent for modified Newton. We begin by introducing the required definitions and assumptions. Suppose we are solving $F(x) = 0$. Define $J = \frac{\partial F}{\partial x}$ and \hat{J} the approximate Jacobian, such that $\hat{J}^{-1}J - I = N$, where $N^2 = 0$. Let F be continuously differentiable in a convex set $D \subset \mathbb{R}^n$. Assume that there exists $x_* \in \mathbb{R}^n$ and $r > 0$ such that $\mathcal{N}(x_*, r) \subset D$, where $F(x_*) = 0$. Suppose that $\hat{J}^{-1}(x_*)$ exists and is bounded, $\|\hat{J}^{-1}(x_*)\| \leq \beta$. Suppose $\hat{J} \in Lip_{\gamma'} \mathcal{N}(x_*, r)$, $J \in Lip_{\gamma} \mathcal{N}(x_*, r)$, where $\|N\|$ is bounded, and $\|N\| \leq M$ in D .

Lemma 2.1

Let $\bar{x} \in \mathcal{N}(x_*, \epsilon)$ and $\|\hat{J}^{-1}(x_*)\| \leq \beta$, where $\epsilon \leq \frac{1}{2\beta\gamma'}$. Then $\|\hat{J}(\bar{x})^{-1}\| \leq 2\beta$.

Proof

First note that

$$\begin{aligned} \|\hat{J}(x_*)^{-1}(\hat{J}(\bar{x}) - \hat{J}(x_*))\| &\leq \|\hat{J}(x_*)^{-1}\| \|\hat{J}(\bar{x}) - \hat{J}(x_*)\| \\ &\leq \beta\gamma' \|\bar{x} - x_*\| \\ &\leq \beta\gamma'\epsilon \leq \frac{1}{2}. \end{aligned} \tag{12}$$

Then by the perturbation relation (3.1.20 in [5]), $\hat{J}(\bar{x})$ is nonsingular and

$$\begin{aligned} \|\hat{J}(\bar{x})^{-1}\| &\leq \frac{\|\hat{J}(x_*)^{-1}\|}{1 - \|\hat{J}(x_*)^{-1}(\hat{J}(\bar{x}) - \hat{J}(x_*))\|} \\ &\leq 2\beta. \square \end{aligned} \tag{13}$$

Lemma 2.2

Let the conditions of Lemma 2.1 hold and $x_i \in \mathcal{N}(x_*, r)$. Define η to be the maximum distance between x_i and \bar{x}_i . Then

$$\|\hat{J}(\bar{x})^{-1}(J(x_i) - \hat{J}(\bar{x}))\| \leq 2\beta\gamma\|x_i - \bar{x}\| + M \leq O(\eta) + M.$$

Proof

$$\hat{J}(\bar{x})^{-1}(J(x_i) - \hat{J}(\bar{x})) = \hat{J}(\bar{x})^{-1}(J(x_i) - J(\bar{x})) + \hat{J}(\bar{x})^{-1}(J(\bar{x}) - \hat{J}(\bar{x})). \tag{14}$$

The norm of the first term is bounded by $2\beta\gamma\|x_i - \bar{x}\|$, using the bound on $\|\hat{J}(\bar{x})\|^{-1}$ (Lemma 2.1) and Lipschitz continuity of J . The second term is

$$\hat{J}(\bar{x})^{-1}J(\bar{x}) - I = N, \tag{15}$$

where $\|N\| \leq M$ by assumption. \square

Lemma 2.3

Let the conditions of Lemma 2.1 hold, and define η to be the maximum distance between x_i and \bar{x}_i . Then

$$\|\hat{J}(\bar{x}_i)^{-1}(\hat{J}(\bar{x}_i) - J(x_i))\hat{J}(\bar{x}_{i-1})^{-1}(\hat{J}(\bar{x}_{i-1}) - J(x_{i-1}))\| \leq O(\eta).$$

Proof

$$\begin{aligned} & \hat{J}(\bar{x}_i)^{-1}(\hat{J}(\bar{x}_i) - J(x_i))\hat{J}(\bar{x}_{i-1})^{-1}(\hat{J}(\bar{x}_{i-1}) - J(x_{i-1})) \tag{16} \\ &= (\hat{J}(\bar{x}_i)^{-1}(J(\bar{x}_i) - J(x_i)) + \hat{J}(\bar{x}_i)^{-1}(\hat{J}(\bar{x}_i) - J(\bar{x}_i))) \\ & \quad \cdot (\hat{J}(\bar{x}_{i-1})^{-1}(J(\bar{x}_{i-1}) - J(x_{i-1})) + \hat{J}(\bar{x}_{i-1})^{-1}(\hat{J}(\bar{x}_{i-1}) - J(\bar{x}_{i-1}))) \\ &= \hat{J}^{-1}(\bar{x}_i)(J(\bar{x}_i) - J(x_i))(\hat{J}(\bar{x}_{i-1})^{-1}(J(\bar{x}_{i-1}) - J(x_{i-1}))) \\ & \quad + \hat{J}(\bar{x}_i)^{-1}(\hat{J}(\bar{x}_i) - J(x_i))\hat{J}(\bar{x}_{i-1})^{-1}(J(\bar{x}_{i-1}) - J(x_{i-1})) \\ & \quad + \hat{J}(\bar{x}_i)^{-1}(J(\bar{x}_i) - J(x_i))\hat{J}(\bar{x}_{i-1})^{-1}(\hat{J}(\bar{x}_{i-1}) - J(\bar{x}_{i-1})) \\ & \quad + \hat{J}(\bar{x}_i)^{-1}(\hat{J}(\bar{x}_i) - J(\bar{x}_i))\hat{J}(\bar{x}_{i-1})^{-1}(\hat{J}(\bar{x}_{i-1}) - J(\bar{x}_{i-1})). \end{aligned}$$

The last term is zero because $\hat{J}(\bar{x})^{-1}(\hat{J}(\bar{x}) - J(\bar{x}))$ is nilpotent. The norm of the first term is bounded by

$$(2\beta\gamma'\eta_i)(2\beta\gamma'\eta_{i-1}) = 4\beta^2(\gamma')^2\eta_i\eta_{i-1}. \tag{17}$$

The norm of the second term is bounded by $4\beta^2 M\eta_{i-1}$. The norm of the third term is bounded by $4\beta^2 M\eta_i$. \square

Theorem 1 *Suppose we are solving $F(x) = 0$ by modified Newton iteration*

$$x_{i+1} = x_i - \hat{J}^{-1}(\bar{x}_i)F(x_i), \tag{18}$$

where $J = \frac{\partial F}{\partial x}$ and the approximate Jacobian \hat{J} satisfies $\hat{J}^{-1}J - I = N$, where $N^2 = 0$. Let F be continuously differentiable in a convex set $D \subset \mathbb{R}^n$. Assume that there exists $x_* \in \mathbb{R}^n$ and $r > 0$ such that $\mathcal{N}(x_*, r) \subset D$, where $F(x_*) = 0$. Suppose that $\hat{J}^{-1}(x_*)$ exists and is bounded, $\|\hat{J}^{-1}(x_*)\| \leq \beta$. Suppose $\hat{J} \in \text{Lip}_{\gamma'}\mathcal{N}(x_*, r)$, $J \in \text{Lip}_{\gamma}\mathcal{N}(x_*, r)$, $\|N\|$ bounded, $\|N\| \leq M$ in D .

Then $\exists \varepsilon > 0$ such that for all $x_* \in \mathcal{N}(x_*, \varepsilon')$, where $\varepsilon' = \varepsilon/(M+1)^2$, the sequence generated by (18) is well-defined and converges to x_* . The convergence is 2-step quadratic for $\eta = 0$, where η is the maximum distance between x_i and \bar{x}_i . Note that $\eta = 0$ for the ‘full Newton’ version.

Proof: The proof is an extension of the Newton convergence proof in [5], p.90. The main complication is due to the inaccurate matrix. We will show that if

$x_0 \in \mathcal{N}(x_*, \varepsilon')$, then

$$x_{2i} \in \mathcal{N}\left(x_*, \varepsilon' \left(\frac{3}{4}\right)^i\right), \quad (19a)$$

$$x_{2i+1} \in \mathcal{N}\left(x_*, \varepsilon'' \left(\frac{3}{4}\right)^i\right), \quad (19b)$$

where $\varepsilon' = \varepsilon/(M+1)^2$ as above, and $\varepsilon'' = \varepsilon/(M+1)$. Thus the iteration converges, and the Jacobians are defined and invertible. Finally, we will show that the convergence is 2-step quadratic for $\eta = 0$.

We will choose ε so that $\hat{J}(x)$ is nonsingular for any $x \in \mathcal{N}(x_*, \varepsilon)$, and then show the convergence result. Let

$$\varepsilon = \min\left\{r, \frac{1}{8}, \frac{1}{8(\beta\gamma + \beta\gamma' + \beta^2\gamma)}\right\}. \quad (20)$$

By Lemma 2.1, $\|\hat{J}(\bar{x}_0)^{-1}\| \leq 2\beta$. Therefore, x_1 is well-defined and

$$\begin{aligned} x_1 - x_* &= x_0 - x_* - \hat{J}(\bar{x}_0)^{-1}F(x_0) \\ &= x_0 - x_* - \hat{J}(\bar{x}_0)^{-1}[F(x_0) - F(x_*)] \\ &= \hat{J}(\bar{x}_0)^{-1}[F(x_*) - F(x_0) - \hat{J}(\bar{x}_0)(x_* - x_0)] \\ &= \hat{J}(\bar{x}_0)^{-1}[F(x_*) - F(x_0) - J(x_0)(x_* - x_0)] \\ &\quad + \hat{J}(\bar{x}_0)^{-1}(J(x_0) - \hat{J}(\bar{x}_0))(x_* - x_0). \end{aligned} \quad (21)$$

Using Lemma 2.1, Lemma 2.2, Lemma 4.1.12 of [5] and the definition of η ,

$$\|x_1 - x_*\| \leq \beta\gamma\|x_0 - x_*\|^2 + (M + O(\eta))\|x_0 - x_*\|. \quad (22)$$

Using (20) and $\|x_0 - x_*\| \leq \varepsilon'$,

$$\begin{aligned} \|x_1 - x_*\| &\leq \beta\gamma\|x_0 - x_*\|^2 \\ &\quad + (M + O(\eta))\|x_0 - x_*\| \\ &\leq (\beta\gamma\|x_0 - x_*\| + M + O(\eta))\|x_0 - x_*\| \\ &\leq \left(\frac{\beta\gamma\varepsilon}{(M+1)^2} + M + O(\eta)\right)\|x_0 - x_*\| \\ &\leq \left(\frac{\beta\gamma}{8(M+1)^2(\beta\gamma + \beta\gamma' + \beta^2\gamma)} + M + O(\eta)\right)\|x_0 - x_*\| \\ &\leq (M+1)\|x_0 - x_*\| \\ &\leq \frac{\varepsilon}{M+1}. \end{aligned} \quad (23)$$

Thus, $x_1 \in \mathcal{N}(x_*, \varepsilon'')$, so the Jacobian at x_1 is well-defined.

Now consider x_2 ,

$$\begin{aligned}
x_2 - x_* &= x_1 - x_* - \hat{J}(\bar{x}_1)^{-1}F(x_1) \\
&= x_1 - x_* - \hat{J}(\bar{x}_1)^{-1}[F(x_1) - F(x_*)] \\
&= \hat{J}(\bar{x}_1)^{-1}[F(x_*) - F(x_1) - \hat{J}(\bar{x}_1)(x_* - x_1)] \\
&= \hat{J}(\bar{x}_1)^{-1}[F(x_*) - F(x_1) - J(\bar{x}_1)(x_* - x_1)] \\
&\quad + \hat{J}(\bar{x}_1)^{-1}(J(\bar{x}_1) - \hat{J}(\bar{x}_1))(x_* - x_1) \\
&= \hat{J}(\bar{x}_1)^{-1}[F(x_*) - F(x_1) - J(x_1)(x_* - x_1)] \\
&\quad + \hat{J}(\bar{x}_1)^{-1}(J(x_1) - \hat{J}(\bar{x}_1))\hat{J}(\bar{x}_0)^{-1}[F(x_*) - F(x_0) - J(x_0)(x_* - x_0)] \\
&\quad + \hat{J}(\bar{x}_1)^{-1}(J(x_1) - \hat{J}(\bar{x}_1))\hat{J}(\bar{x}_0)^{-1}(J(x_0) - \hat{J}(\bar{x}_0))(x_* - x_0).
\end{aligned} \tag{24}$$

Thus, using (20), Lemmas 2.1, 2.2, 2.3, 4.1.12 of [5] and the definition of η ,

$$\|x_2 - x_*\| \leq \beta\gamma\|x_1 - x_*\|^2 + \beta\gamma(M + O(\eta))\|x_0 - x_*\|^2 + O(\eta)\|x_0 - x_*\|. \tag{25}$$

In addition, using (20) and $\|x_0 - x_*\| \leq \varepsilon'$,

$$\|x_2 - x_*\| \leq \frac{1}{2}\|x_0 - x_*\| \leq \frac{3}{4}\|x_0 - x_*\|. \tag{26}$$

Thus, $x_2 \in \mathcal{N}(x_*, \varepsilon' \cdot \frac{3}{4})$.

The iteration for $x_i, i \geq 3$ proceeds similarly, i.e.

$$\begin{aligned}
x_i - x_* &= \hat{J}(\bar{x}_{i-1})^{-1}([F(x_*) - F(x_{i-1}) - J(x_{i-1})(x_* - x_{i-1})] \\
&\quad + (J(x_{i-1}) - \hat{J}(\bar{x}_{i-1}))\hat{J}(\bar{x}_{i-2})^{-1}[F(x_*) - F(x_{i-2}) - J(x_{i-2})(x_* - x_{i-2})] \\
&\quad + (J(x_{i-1}) - \hat{J}(\bar{x}_{i-1}))\hat{J}(\bar{x}_{i-2})^{-1}(J(x_{i-2}) - \hat{J}(\bar{x}_{i-2}))(x_* - x_{i-2})).
\end{aligned} \tag{27}$$

Thus,

$$\|x_i - x_*\| \leq \beta\gamma\|x_{i-1} - x_*\|^2 + \beta\gamma(M + O(\eta))\|x_{i-2} - x_*\|^2 + O(\eta)\|x_{i-2} - x_*\|. \tag{28}$$

Now, we would like to show that

$$\|x_{2i} - x_*\| \leq \frac{\varepsilon}{(M+1)^2} \left(\frac{3}{4}\right)^i, \quad \|x_{2i+1} - x_*\| \leq \frac{\varepsilon}{M+1} \left(\frac{3}{4}\right)^i. \tag{29}$$

We have shown this for $i = 0, 1$. The rest will be by induction. For i odd, (28) yields

$$\begin{aligned}
\|x_{2i+1} - x_*\| &\leq \beta\gamma\|x_{2i} - x_*\|^2 + \beta\gamma(M + O(\eta))\|x_{2i-1} - x_*\|^2 \\
&\quad + O(\eta)\|x_{2i-1} - x_*\| \\
&\leq \frac{\beta\gamma\varepsilon^2}{(M+1)^4} \left(\frac{3}{4}\right)^{2i} + \beta\gamma(M + O(\eta)) \cdot \frac{\varepsilon^2}{(M+1)^2} \left(\frac{3}{4}\right)^{2(i-2)} \\
&\quad + O(\eta) \left(\frac{3}{4}\right)^{i-2} \frac{\varepsilon}{M+1} \\
&\leq \frac{\varepsilon}{M+1} \left(\frac{3}{4}\right)^i \left[\left(\frac{3}{4}\right)^i \cdot \frac{\beta\gamma\varepsilon}{(M+1)^3} + \left(\frac{3}{4}\right)^{i-2} \frac{\beta\gamma\varepsilon(M + O(\eta))}{M+1} + O(\eta) \left(\frac{3}{4}\right)^{-2} \right] \\
&\leq \frac{\varepsilon}{M+1} \left(\frac{3}{4}\right)^i \left[\frac{1}{8} + \frac{1}{3} + O(\eta) \cdot \left(\frac{4}{3}\right)^2 \right] \\
&\leq \frac{\varepsilon}{M+1} \left(\frac{3}{4}\right)^i,
\end{aligned} \tag{30}$$

assuming a choice of neighborhood \mathcal{N} such that η is sufficiently small. For i even, (28) yields

$$\begin{aligned}
\|x_{2i} - x_*\| &\leq \beta\gamma\|x_{2i-1} - x_*\|^2 + \beta\gamma(M + O(\eta))\|x_{2i-2} - x_*\|^2 \\
&\quad + O(\eta)\|x_{2i-2} - x_*\| \\
&\leq \frac{\beta\gamma\varepsilon^2}{(M+1)^2} \left(\frac{3}{4}\right)^{2(i-1)} + \beta\gamma(M + O(\eta)) \frac{\varepsilon^2}{(M+1)^4} \left(\frac{3}{4}\right)^{2(i-1)} \\
&\quad + O(\eta) \frac{\varepsilon}{(M+1)^2} \left(\frac{3}{4}\right)^{i-1} \\
&\leq \frac{\varepsilon}{(M+1)^2} \left(\frac{3}{4}\right)^i \cdot \left[\left(\frac{3}{4}\right)^{i-2} \beta\gamma\varepsilon + \left(\frac{3}{4}\right)^{i-2} \frac{\beta\gamma\varepsilon(M + O(\eta))}{(M+1)^2} + \frac{4}{3}O(\eta) \right] \\
&\leq \frac{\varepsilon}{(M+1)^2} \left(\frac{3}{4}\right)^i \cdot \left[\frac{1}{8} + \frac{1}{8} + \frac{4}{3}O(\eta) \right] \\
&\leq \frac{\varepsilon}{(M+1)^2} \cdot \left(\frac{3}{4}\right)^i.
\end{aligned} \tag{31}$$

Hence,

$$x_{2i} \in \mathcal{N}\left(x_*, \varepsilon' \left(\frac{3}{4}\right)^i\right), \quad x_{2i+1} \in \mathcal{N}\left(x_*, \varepsilon'' \left(\frac{3}{4}\right)^i\right),$$

so that by (28) the iteration converges, and this convergence is 2-step quadratic if $\eta = 0$. \square

4 DASSLSO Numerical Results

DASSLSO was tested on several problems of varying size and complexity. The problems described in the following subsections proved to be the most interesting. When feasible, the approximate solution produced via finite differences was compared with the solution of the analytic sensitivity equations found using DASSL. The results appeared to be within the order of the error tolerances used. Solution plots are displayed in Appendices 1 through 3.

4.1 Gas-Oil Cracking Problem

The Gas-Oil Cracking problem is a 2 dimensional ODE with 3 parameters which has appeared as a test example in many papers. The problem has the form

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} -(p_1 + p_3)x_1^2 \\ p_1x_1^2 - p_2x_2 \end{pmatrix}, \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \quad (32)$$

Note that for this problem $n_y = 2$ and $n_p = 3$. Thus, the sensitivity solution vector Y in DASSLSO will have $n_y(n_p + 1) = 8$ elements. The resulting combined system (i.e. the system (5) of the introduction) defines an 8×8 system of equations. Analytically, this system takes the form:

$$\begin{pmatrix} x_1' \\ x_2' \\ s_1' \\ s_2' \\ s_3' \\ s_4' \\ s_5' \\ s_6' \end{pmatrix} = \begin{pmatrix} -(p_1 + p_3)x_1^2 \\ p_1x_1^2 - p_2x_2 \\ -2s_1(p_1 + p_3)x_1 - x_1^2 \\ 2s_1p_1x_1 - s_2p_2 + x_1^2 \\ -2s_3(p_1 + p_3)x_1 \\ 2s_3p_1x_1 - s_4p_2 - x_2 \\ -2s_5(p_1 + p_3)x_1 - x_1^2 \\ 2s_5p_1x_1 - p_2s_6 \end{pmatrix}, \quad \begin{pmatrix} x_1(0) \\ x_2(0) \\ s_1(0) \\ s_2(0) \\ s_3(0) \\ s_4(0) \\ s_5(0) \\ s_6(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

In the DASSLSO approximation to the sensitivity equations of (32), the relative and absolute error tolerances were set at 10^{-7} , the parameter values were set to $p_1 = 0.9875$, $p_2 = 0.2566$, $p_3 = 0.3323$ and the first order forward finite-difference approximation to the sensitivity equations was used. The approximate solution is displayed in Appendix 1.

4.2 Single Pendulum Problem

This classical second order ODE has been reformulated as an index-two DAE for the purpose of testing. The problem takes the form

$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ 0 \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \\ -x_1x_5 \\ x_2x_5 - g \\ x_1x_3 + x_2x_4 \end{pmatrix}, \quad \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \end{pmatrix} = \begin{pmatrix} 0.5 \\ -(p_1^2 - x_1^2)^{\frac{1}{2}} \\ 0.0 \\ 0.0 \end{pmatrix}. \quad (33)$$

Here, $n_y = 5$, $n_p = 1$ and $\text{NEQ} = n_y(n_p + 1) = 10$ within the driver routine for this problem. It should be noted that the constraint corresponding to the variable x_5 tends to adversely effect the performance of DASSL/DASSLSO, as is usually the case with index-2 DAEs. This effect can be mitigated by eliminating this variable from the error test, which is an option in DASSLSO. For this example, we applied the error test only to the variables x_1 through x_4 , and set all RTOL and ATOL values to 10^{-6} . The centered difference approximation to the sensitivity equations was used in DASSLSO to produce the numerical solutions to (33), which are displayed in Appendix 2.

4.3 Batch-Reactor Problem

This interesting problem, which consists of a series of very stiff ODEs and constraints, was solved in [4] using the sensitivity code DASSAC. The problem has the form

$$\begin{pmatrix} u_1' \\ u_2' \\ u_3' \\ u_4' \\ u_5' \\ u_6' \\ u_7' \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -p_3 u_2 u_8 \\ -p_1 u_2 u_6 + p_2 u_{10} - p_3 u_2 u_8 \\ -p_3 u_2 u_8 + p_4 u_4 u_6 - p_5 u_9 \\ -p_4 u_4 u_6 - p_5 u_9 \\ p_1 u_2 u_6 - p_2 u_{10} \\ -p_1 u_2 u_6 - p_4 u_4 u_6 p_2 u_{10} + p_5 u_9 \\ -0.0131 + u_6 + u_8 + u_9 + u_{10} \\ u_8 - p_7 u_1 / (p_7 - u_7) \\ u_9 - p_8 u_3 / (p_8 - u_7) \\ u_{10} - p_6 u_5 / (p_6 - u_7) \end{pmatrix}$$

with initial conditions,

$$\begin{pmatrix} u_1(0) \\ u_2(0) \\ u_3(0) \\ u_4(0) \\ u_5(0) \\ u_6(0) \\ u_7(0) \\ u_8(0) \\ u_9(0) \\ u_{10}(0) \end{pmatrix} = \begin{pmatrix} 1.5776 \\ 8.32 \\ 0 \\ 0 \\ 0 \\ 0.0131 \\ 0.5(-p_7 + (p_7^2 + 4p_7 u_1(0))) \\ 0.5(-p_7 + (p_7^2 + 4p_7 u_1(0))) \\ 0 \\ 0 \end{pmatrix}.$$

As in [4], the following rate and equilibrium constant values were used,

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{pmatrix} = \begin{pmatrix} 21.893 \text{ hr}^{-1} K g m o l e^{-1} \\ 2.14 \times 10^9 \text{ hr}^{-1} \\ 32.318 \text{ hr}^{-1} K g m o l e^{-1} \\ 21.893 \text{ hr}^{-1} K g m o l e^{-1} \\ 1.07 \times 10^9 \text{ hr}^{-1} \\ 7.65 \times 10^{-18} g m o l e K g^{-1} \\ 4.03 \times 10^{-11} g m o l e K g^{-1} \\ 5.32 \times 10^{-18} g m o l e K g^{-1} \end{pmatrix}.$$

Since the problem parameters have such disparate values, scaling difficulties will arise during any attempted numerical solution of this example. To resolve these difficulties, we scaled the RTOL and ATOL values with respect to each individual parameter. For example, when solving for the sensitivity $s_8 = dy/dp_8$ we note that p_8 is on the order of 10^{-18} , so s_8 will be on the order of 10^{18} . Accordingly, we set $RTOL(i) = 10^{-7}$ and $ATOL(i) = 10^{11}$ for $i = 81, \dots, 90$ in order to ensure 7 digits of accuracy (here we used the rule $RTOL(i)/O(p_8) = 10^{11}$). The results are displayed exactly as in [4] in order to facilitate comparison, i.e. the sensitivity approximation has been multiplied by the parameter value in order to scale the results within $[-1,1]$ on the $x_i(t)$ axis, where $x_i(t)$ is the concentration of quantity i . These calculations were carried out using both finite difference approximation options, which yield similar results. The results of the forward difference approximation to the solution of the sensitivity equations and the DASSL approximation to the state variable equations have been displayed in Appendix 3.

5 DASPCKSO Description and Numerical Results

DASPCK is an extension of DASSL specializing in the solution of large-scale systems of DAEs which uses the time-stepping algorithms of DASSL ([2]), with the preconditioned GMRES iterative method to solve the linear system at each Newton iteration, or optionally with direct methods as in DASSL. A detailed description of the algorithms and the code can be found in [3]. Here we will concentrate on its modification for sensitivity analysis, DASPCKSO. DASPCK was designed to require minimal storage for the iterative method option. To this end, it neither generates nor stores the Jacobian matrix explicitly. This is possible because the GMRES iteration requires only the product of the Jacobian and a given vector, which is approximated in DASPCK by a directional difference of F . By using a directional difference approximation to the sensitivity equations as well, we are able to maintain the low storage require-

ment.

DASPKSO utilizes the same sensitivity subroutine as DASSLSO. Indeed, the extensions to the user interface are identical to those of DASSLSO with a few very minor exceptions. DASPKSO was tested on the same problems as DASSLSO, as well as a variety of method of lines (MOL) problems in order to take full advantage of Krylov iteration routine contained within the code. Performance statistics for a two-dimensional MOL problem are given below.

5.1 Two dimensional Heat equation

Consider the two dimensional heat equation with the following parameters

$$\frac{\partial u}{\partial t} = p_1 \frac{\partial^2 u}{\partial x^2} + p_2 \frac{\partial^2 u}{\partial y^2}.$$

We approximate the solution to this equation on the square $0 \leq x, y \leq 1$ with homogeneous boundary conditions ($u = 0$) and the initial condition $u(x, y, 0) = 16xy(1 - x)(1 - y)$. A uniform (x, y) grid with $M = 10$ internal points was used, which implies that $\Delta x, \Delta y = 1/(M + 1) = 0.091$. Thus, the numerical approximation

$$y_{j,k} \approx u(j\Delta x, k\Delta y), \quad 0 \leq j, k \leq M + 1$$

leads to the discrete problem (applying the method of lines)

$$\begin{aligned} y'_{j,k} &= (\Delta x)^{-2}(y_{j+1,k} + y_{j-1,k} + y_{j,k+1} + y_{j,k-1} - 4y_{j,k}), \quad 1 \leq j, k \leq M \\ 0 &= y_{j,k} \quad \textit{otherwise} \end{aligned}$$

Hence, the total size of the state variable system becomes $(M + 2)^2$. Performing a sensitivity analysis upon the two parameters in this problem adds an additional $2(M + 2)^2$ equations to the system, resulting in a total of $3(M + 2)^2 = 432$ equations for the combined system. The solution of this test problem was carried out with the error tolerances $RTOL = ATOL = 10^{-4}$. The default iterative method values for EPLI, MAXL and KMP (0.05, MIN(5,NEQ) and MAXL, respectively) were used, as well as a time dependent, (lumped) banded preconditioner (bandwidth of 5). The following table compares a number of DASPK/DASPKSO operational parameters for the solution of the state variable problem (SV), a sensitivity analysis of the problem using a forward-difference approximation to the sensitivity equations (SA FDM = 1), and a sensitivity analysis of the problem using the centered-difference approximation to the sensitivity equations (SA FDM = 2). For this computation the error

control test was applied to all variables (both state and sensitivity).

Operation	SV	SA FDM = 1	SA FDM = 2
RWORK size	3749	9509	9509
IWORK size	328	328	328
# of Steps	71	76	74
RES eval.	430	1307	2080
Prec. eval.	19	19	19
Prec. solv.	335	404	397
Nonlinear its.	144	151	148
Linear its.	194	253	249
Avg. Krylov dimension	1.3759	1.6755	1.6824

The table below contains the results for the same computation, but with the error control test restricted to the state variables only.

Operation	SA FDM = 1	SA FDM = 2
RWORK size	9509	9509
IWORK size	328	328
# of Steps	66	66
RES eval.	1253	2155
Prec. eval.	19	19
Prec. solv.	386	412
Nonlinear its.	131	132
Linear its.	255	280
Avg. Krylov dimension	1.9466	2.1212

In the above tables, a RES evaluation is considered to be an evaluation of the original system or any n_y -dimensional subsystem. Hence, one evaluation of the combined system (2) involves $n_p + 1$ RES evaluations. Similarly, one preconditioner solve for the combined system involves $n_p + 1$ preconditioner solves. Note that the performance parameters for the sensitivity code DASPKSO are close in value to those produced by DASPK for this problem, with the predictable exception of the number of RES evaluations. DASPKSO does not work significantly harder in order to compute the additional information required for a sensitivity analysis of this problem.

6 Conclusion

Three new codes have been introduced: DASSLSO, DASPKSO and SENSD. The first two allow the user to perform a sensitivity analysis upon the parameters of a given problem in a fast and efficient manner, and are modifications to the ODE/DAE solvers DASSL and DASPK. The third code is an auxiliary routine which allows a user to perform a sensitivity analysis upon a derived quantity (for example, the L2 norm of the solution vector). All of the codes are flexible and easy to use. Each code contains a finite difference option that employs an adaptive-increment directional difference approximation to the sensitivity system which saves storage and increases flexibility for the step-size selection. This option is especially advantageous in the context of DASPKSO since the simultaneous solution of the ODE/DAE system and sensitivity system occurs in a very efficient manner. In addition, the increment in the finite difference approximation is chosen adaptively during the solution process, and the non-linear system generated at each time step is solved via an efficient and parallelizable Newton-type iteration which achieves 2-step quadratic convergence. The codes have been tested on a variety of problems and appear to be robust and efficient, provided the user has taken care to scale the error tolerances appropriately.

The codes DASSLSO, DASPKSO and SENSD, as well as the driver routines for all of the test problems, are available via anonymous FTP from the site [ftp.cs.umn.edu](ftp://ftp.cs.umn.edu), in the `/users/tmaly` directory.

References

- [1] C. Bischof, A. Carle, G. Corliss, A. Griewank and P. Hovland, *ADIFOR - Generating derivative codes from Fortran programs*, Scientific Programming 1 (1992), 11-29.

- [2] K. E. Brenan, S. L. Campbell and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Elsevier Science Publishers, 1989.
- [3] P. N. Brown, A. C. Hindmarsh and L. R. Petzold, *Using Krylov methods in the solution of large-scale differential-algebraic systems*, to appear, SIAM J. Sci. Comput.
- [4] M. Caracotsios and W. E. Stewart, *Sensitivity analysis of initial value problems with mixed ODEs and algebraic equations*, Computers and Chemical Engineering Vol. 9, No. 4, pp 359-365, 1985.
- [5] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983.
- [6] P. E. Gill, W. Murray and M. H. Wright, *Practical Optimization*, Academic Press, 1981.
- [7] E. J. Haug and P. E. Ehle, *Second-order design sensitivity analysis of mechanical system dynamics*, International Journal for Numerical Methods in Engineering, Vol. 18, 1699-1717, 1982.
- [8] E. J. Haug, R. Wehage and N.C. Barman, *Design sensitivity analysis of planar mechanism and machine dynamics*, Transactions of the ASME, Vol. 103, July 1981.
- [9] M. A. Kramer and J. R. Leis, *The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations*, ACM Trans. Math Software **14**, 1 (March 1988) 45-60.
- [10] R. S. Maier, L. R. Petzold and W. Rath, *Parallel solution of large-scale differential-algebraic systems*, to appear, Concurrency: Practice and Experience.
- [11] H. Rabitz, M. Kramer and D. Dacol, *Sensitivity analysis in chemical kinetics*, Ann. Rev. Phys. Chem. 34, 419-461 (1983).
- [12] Y. P. Saad and M. H. Schultz, *A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp. 7(1986), 856-869.
- [13] R. Tomovic and M. Vukobratovic, *General Sensitivity Theory*, American Elsevier, New York, 1972.

7 Appendix 1: Solution plots for the Gas-Oil Cracking problem

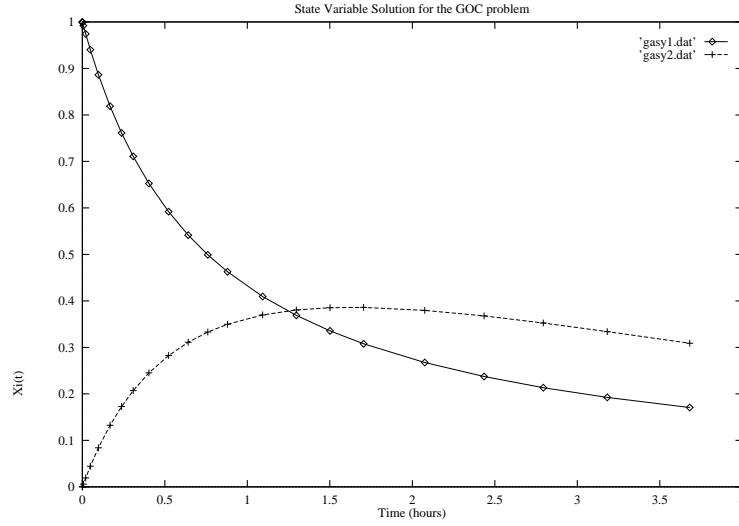


Fig. 1. DASSLSO Gas-Oil Cracking problem state variable solution.

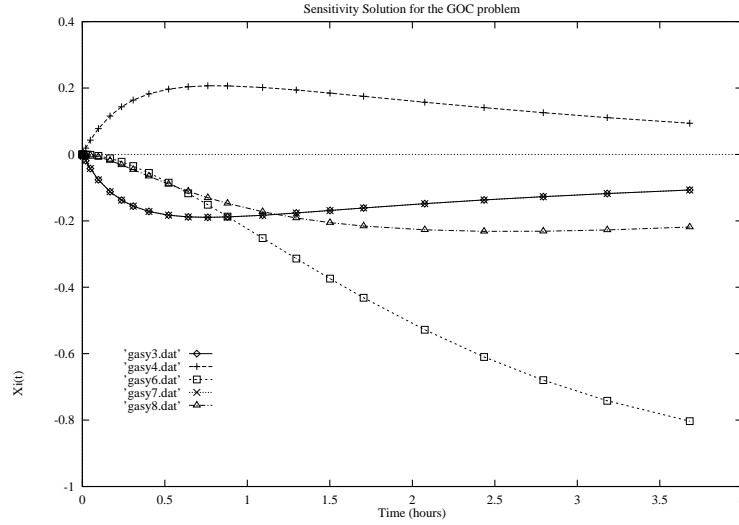


Fig. 2. DASSLSO Gas-Oil Cracking problem sensitivity variable solution. (Sensitivities s_1, s_2, s_4, s_5 and s_6 are plotted. Sensitivities s_i are labelled according to their position in the x vector, i.e. $s_1 \equiv x_3$.)

8 Appendix 2: Solution plots for the Single Pendulum problem

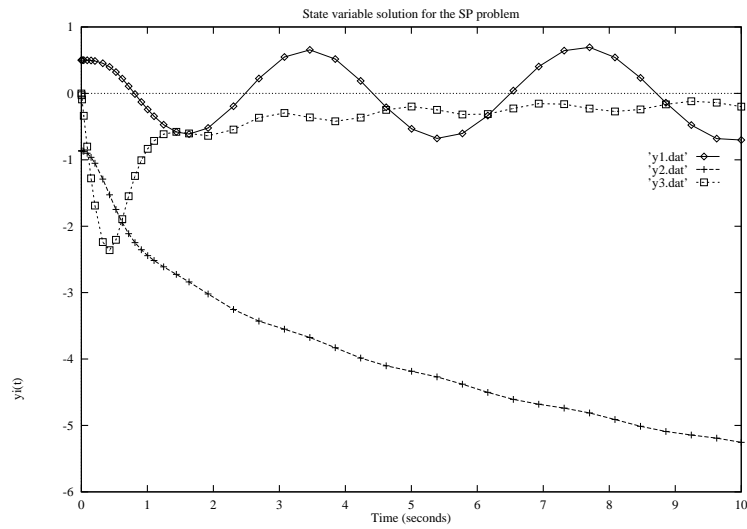


Fig. 3. DASSLSO Single Pendulum problem state variable (1-3) solution.

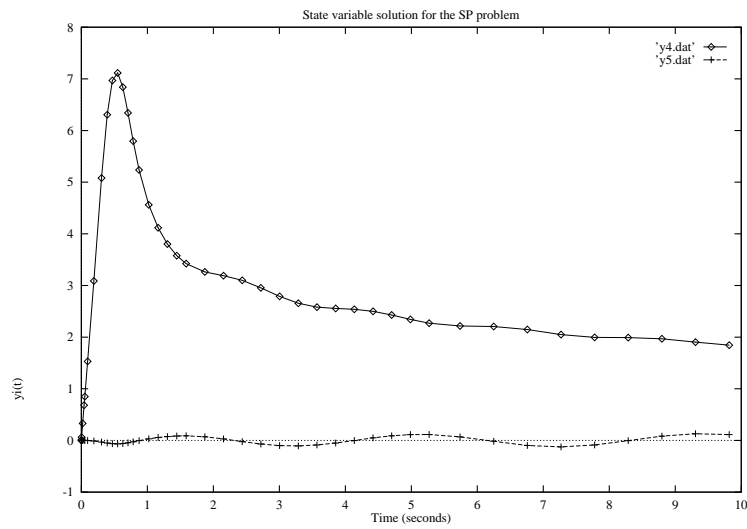


Fig. 4. DASSLSO Single Pendulum problem state variable (4-5) solution.

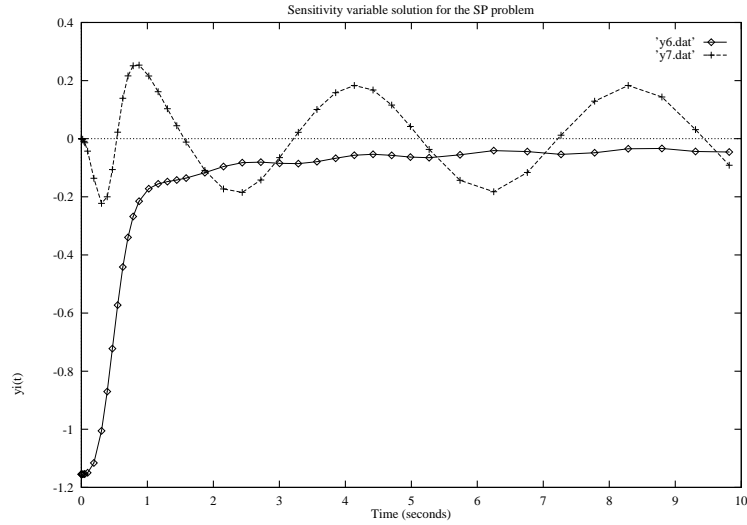


Fig. 5. DASSLSO Single Pendulum problem sensitivity variable solution. (Sensitivities s_1 and s_2 are plotted. Sensitivities are labelled according to their position in the y vector, i.e. $s_1 \equiv y_6$.)

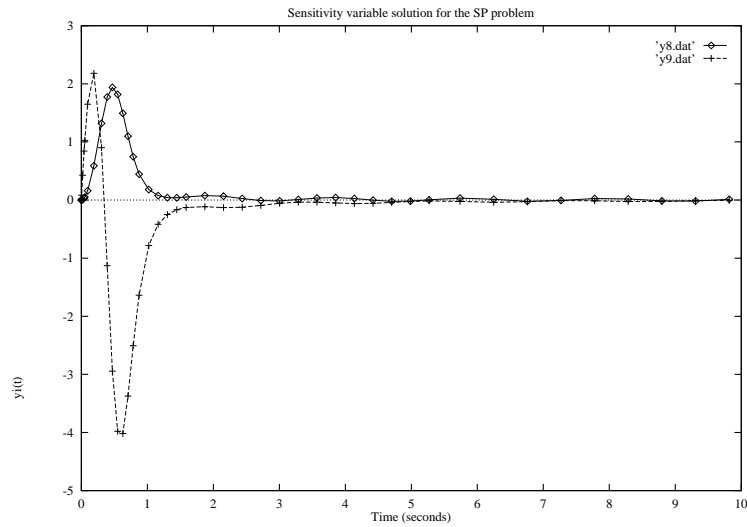


Fig. 6. DASSLSO Single Pendulum problem sensitivity variable solution. (Sensitivities s_3 and s_4 are plotted. Sensitivities are labelled according to their position in the y vector, i.e. $s_3 \equiv y_8$.)

9 Appendix 3: Solution plots for the Batch-Reactor problem

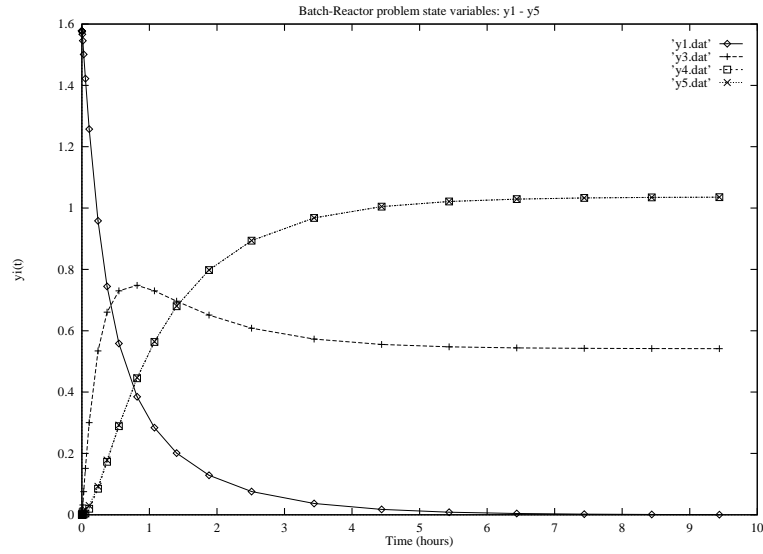


Fig. 7. DASSLSO state variable solution to the Batch-Reactor problem for y_1 , y_3 , y_4 and y_5 .

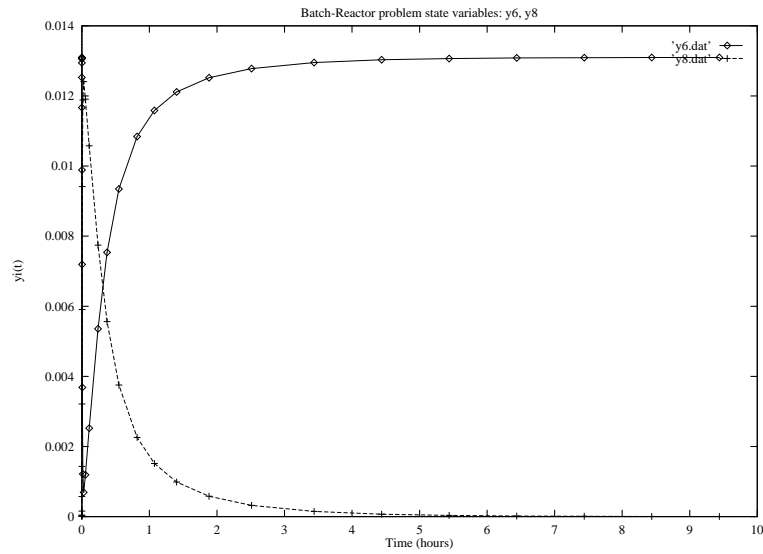


Fig. 8. DASSLSO state variable solution to the Batch-Reactor problem for y_6 , y_8 .

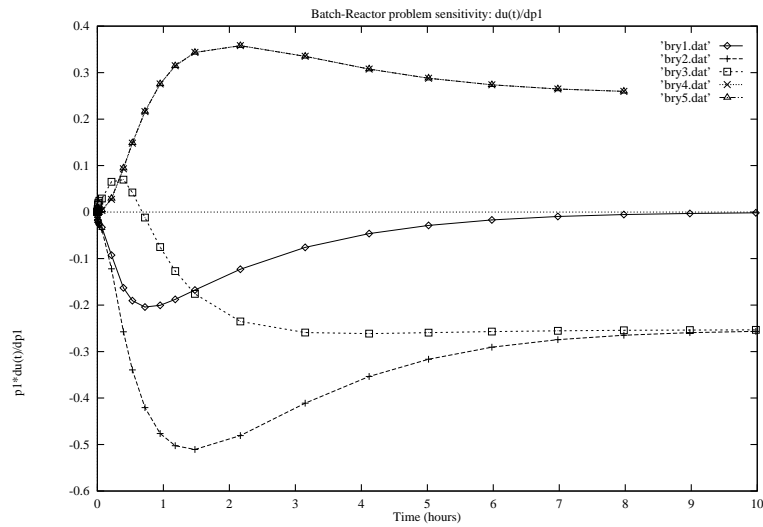


Fig. 9. DASSLSO sensitivity solution to the Batch-Reactor problem for p_1 . (Sensitivities of first 5 solution components with respect to parameter p_1 are plotted.)

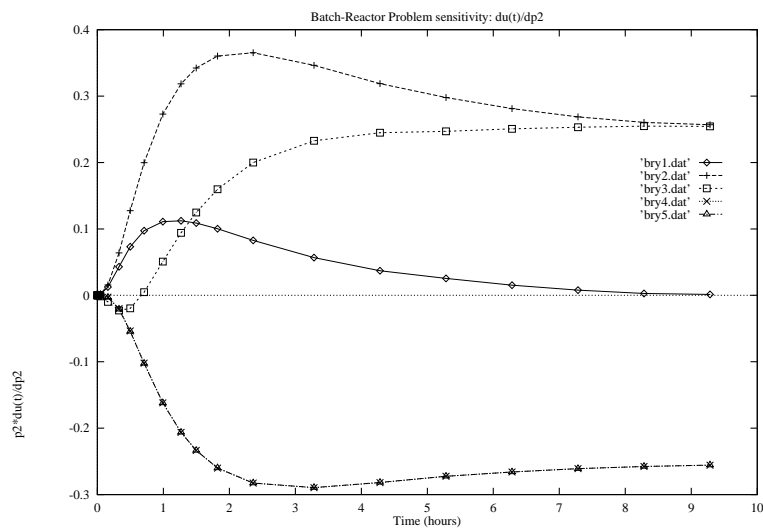


Fig. 10. DASSLSO sensitivity solution to the Batch-Reactor problem for p_2 .

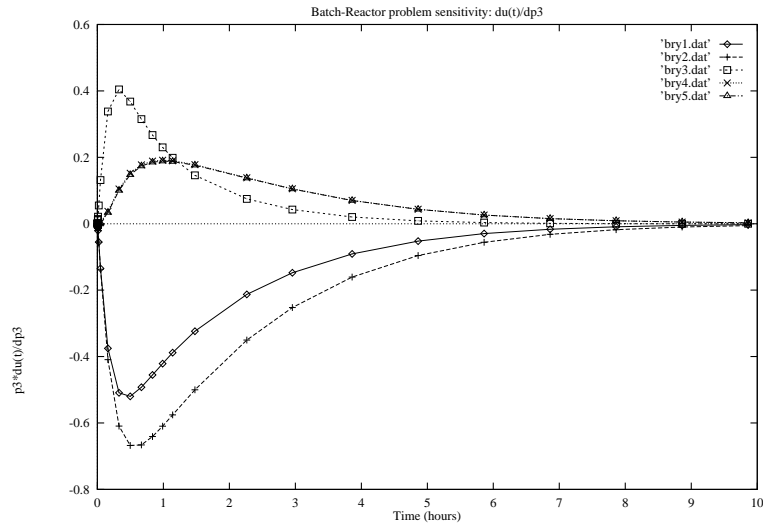


Fig. 11. DASSLSO sensitivity solution to the Batch-Reactor problem for p_3 .

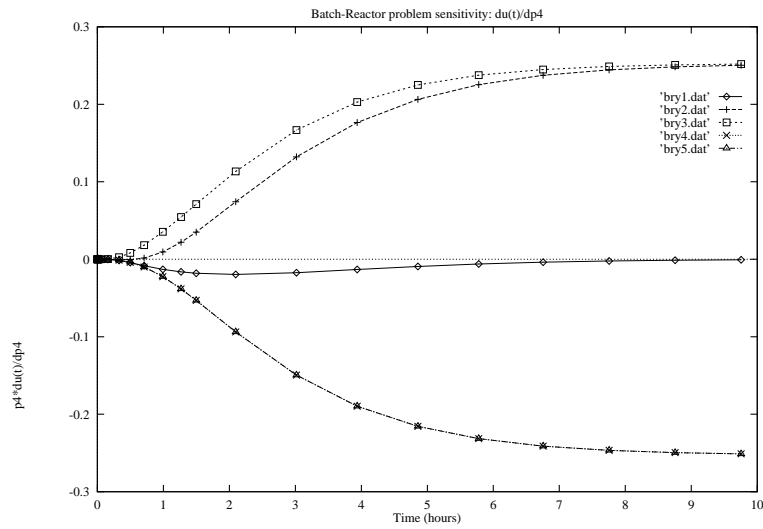


Fig. 12. DASSLSO sensitivity solution to the Batch-Reactor problem for p_4 .

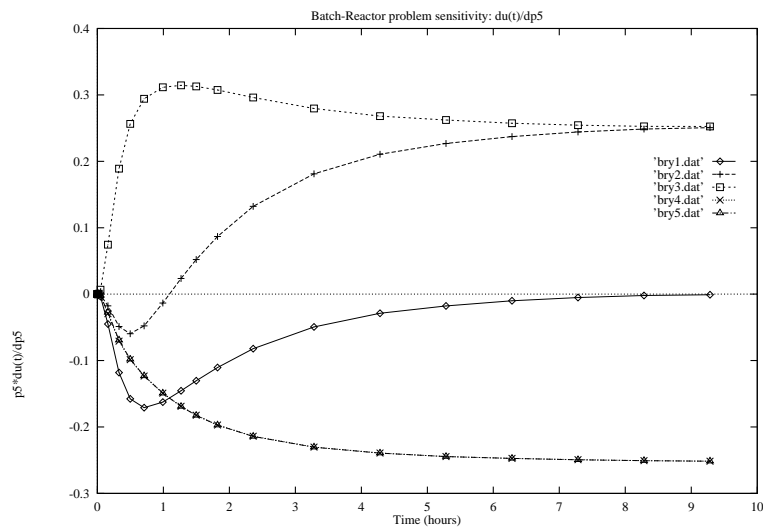


Fig. 13. DASSLSO sensitivity solution to the Batch-Reactor problem for p_5 .

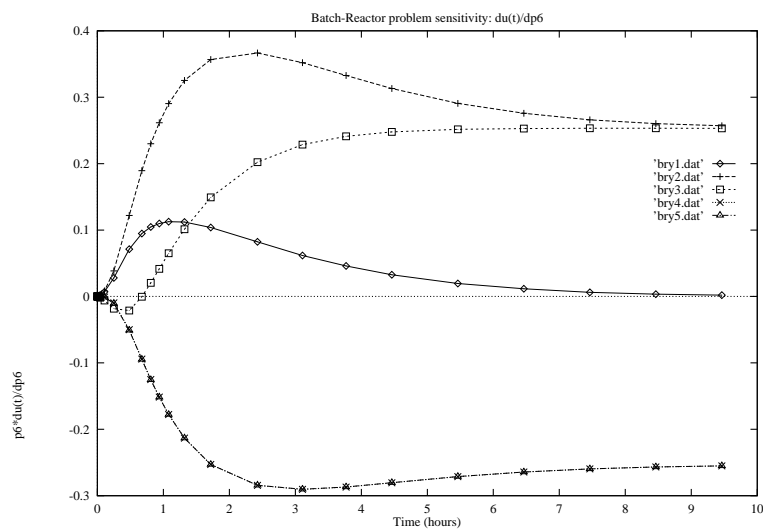


Fig. 14. DASSLSO sensitivity solution to the Batch-Reactor problem for p_6 .

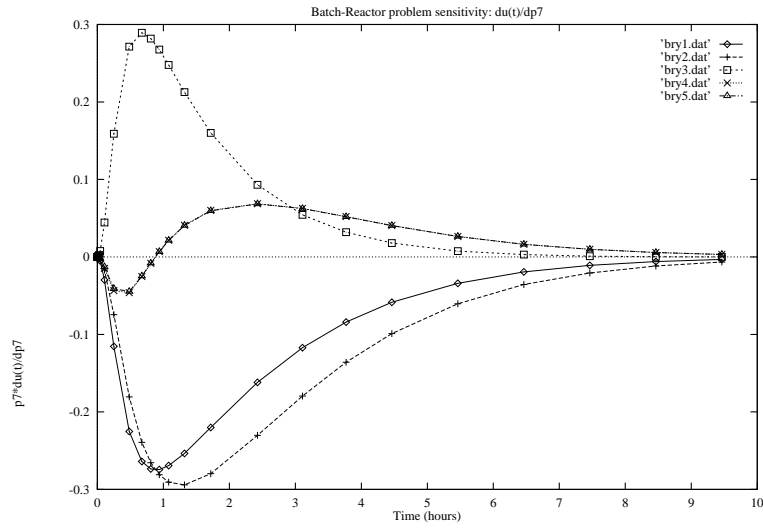


Fig. 15. DASSLSO sensitivity solution to the Batch-Reactor problem for p_7 .

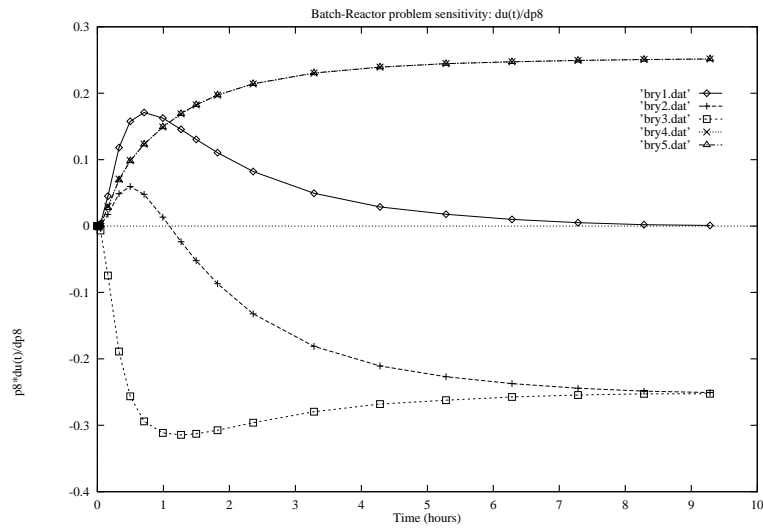


Fig. 16. DASSLSO sensitivity solution to the Batch-Reactor problem for p_8 .