Automatic Differentiation in Ordinary Differential Equations

Joseph Gergaud, Toulouse Univ., INP-ENSEEIHT-IRIT, UMR CNRS 5505, 2 rue Camichel, 31071 Toul

October 22, 2025

Abstract

1 Introduction

The objective of the article is to present the goods methods for computing the derivatives of the flow in the final time t_f with respect of the initial condition, a parameter or the initial time t_0. In the second section we present the different possibilities for computing these derivatives and in the third section we make some numerical comparaisons.

2 Mathematical results

Let f a continuous differentiable function from an open Ω of $\mathbf{R} \times \mathbf{R}^n \times \mathbf{R}^p$ into $\mathbf{R}^n$ We note $x(t, t_0, x_0, \lambda)$ the solution at time t of the following initial value problem

$$(IVP) \left\{ \begin{array}{l} \dot{x} = f(t, x, \lambda) \\ x(t_0) = x_0(\lambda) \end{array} \right.$$

which is defined on the intervalle $]\omega_-(t_0, x_0, \lambda), \omega_-(t_0, x_0, \lambda)[$. We denote

$$\mathcal{O} = \{(t, t_0, x_0, \lambda) \in \mathbf{R} \times \Omega, \omega_-(t_0, x_0, \lambda) < t < \omega_+(t_0, x_0, \lambda)\}.$$

which is an open set.

Théorème 2.1. *Let $f : \Omega \subset \mathbf{R} \times \mathbf{R}^n \times \mathbf{R}^p \to \mathbf{R}^n$, Ω open, f continue. We suppose that f have continuous partial derivatives with respect to x and λ*

$$\frac{\partial f}{\partial x} : \Omega \to \mathcal{M}_n(\mathbf{R}) \quad \frac{\partial f}{\partial \lambda} : \Omega \to \mathcal{M}_{(n,p)}(\mathbf{R}).$$

Then the function $x : \mathcal{O} \to \mathbf{R}^n$ is C^1 (it is C^{k+1}, $1 \leq k \leq \infty$, if f is C^k) and

(i) the function $\frac{\partial x}{\partial x_0}(., t_0, x_0, \lambda_0)$ $:]\omega_-(t_0, x_0, \lambda_0), \omega_+(t_0, x_0, \lambda_0)[\to \mathcal{M}_n(\mathbf{R})$ is the solution of the linear ordinary differential system

$$(VAR1)^1 \left\{ \begin{array}{l} \dot{\widehat{\delta x}}(t) = A(t)\delta x(t) \\ \delta x(t_0) = I_n, \end{array} \right.$$

where $A(t) = \frac{\partial f}{\partial x}(t, x(t, t_0, x_0, \lambda_0), \lambda_0) \in \mathcal{M}_n(\mathbf{R})$ and I_n is the identity matrix of order n.

(ii) the function $\frac{\partial x}{\partial t_0}(., t_0, x_0, \lambda_0)$ $:]\omega_-(t_0, x_0, \lambda_0), \omega_+(t_0, x_0, \lambda_0)[\to \mathbf{R}^n$ est is the solution of the linear ordinary diffferential equation

$$(VAR2) \left\{ \begin{array}{l} \dot{\widehat{\delta x}}(t) = A(t)\delta x(t) \\ \delta x(t_0) = -f(t_0, x_0, \lambda_0). \end{array} \right.$$

(iii) the function $\frac{\partial x}{\partial \lambda}(., t_0, x_0, \lambda_0)$ $:]\omega_-(t_0, x_0, \lambda_0), \omega_+(t_0, x_0, \lambda_0)[\to \mathcal{M}_{n,p}(\mathbf{R})$ is the solution of the linear ordinary differential equation

$$(VAR3) \left\{ \begin{array}{l} \dot{\widehat{\delta x}}(t) = A(t)\delta x(t) + B(t) \\ \delta x(t_0) = x_0'(\lambda), \end{array} \right.$$

with $B(t) = \frac{\partial f}{\partial \lambda}(t, x(t; t_0, x_0, \lambda_0), \lambda_0) \in \mathcal{M}_{n,p}(\mathbf{R})$ and $0_{n,p}$ is the 0 matrix of order (n, p).

[1] Variationnal equation.

3 Numerical results on the brusselator example

3.1 Introduction

In all this section we test the methods on the the Brusselator example of The Hairer and al. book [2], page 201.

$$(IVP)\begin{cases}\dot{x}_1 = 1 + x_1^2 x_2 - (\lambda+1)x_1\\ \dot{x}_2 = \lambda x_1 - x_1^2 x_2\\ x_1(0) = 1.3\\ x_2(0) = \lambda.\end{cases}$$

3.2 Finite differences

We approximate the dérivatives by finite differences. For example for the derivative with respect to the *jth* component of the parameter $\lambda \in \mathbf{R}^n$ is approximates by

$$\frac{\partial x}{\partial \lambda_j}(t, t_0, x_0, \lambda_0) \approx \frac{1}{\delta\lambda}(x(t, t_0, x_0, \lambda_0 + \delta\lambda e_j) - x(t, t_0, x_0, \lambda_0)),$$

where $(e_1, \ldots, e_p)$ denote the canonical basis of $\mathbf{R}^p$.

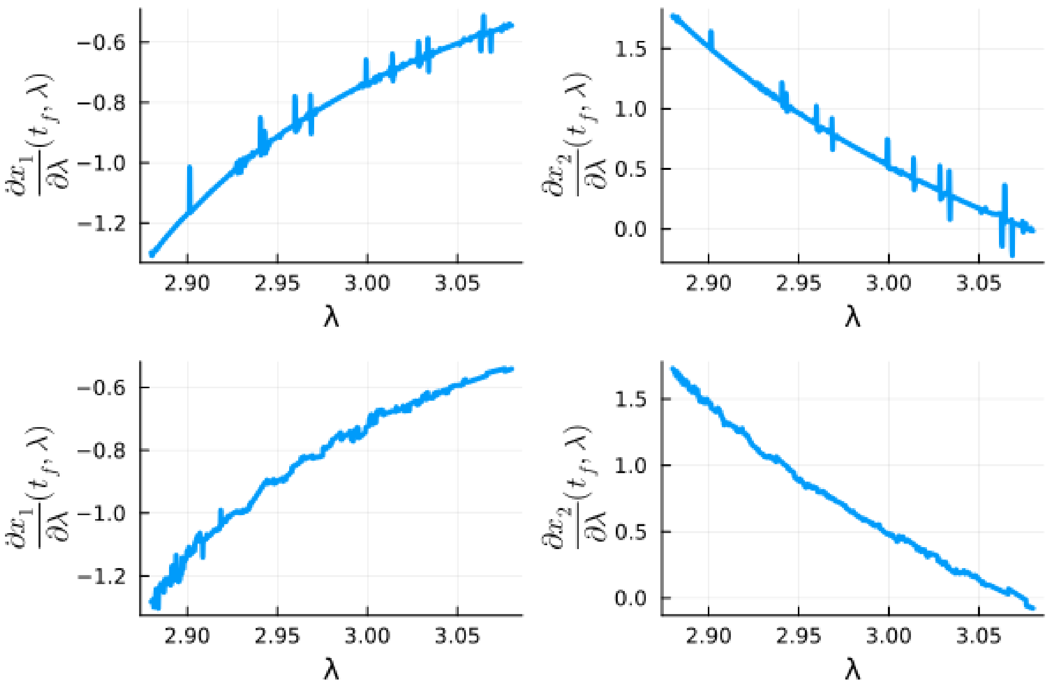

Figure 1: Derivative computing by finite differences. $t_f = 20, \lambda$ ranging from 2.88 to 3.08, $Tol = RelTol = AbsTol = 10^{-4}$. Top graphs is for $\delta\lambda = 4Tol$ and bottom graphs for $\delta\lambda = \sqrt{Tol}$. The numerical integrattion is done with Tsit5().

3.3 Variationnal equation

Here the derivative $\frac{\partial x}{\partial \lambda}(t, t_0, x_0, \lambda_0)$ is the solution of the variational equation

$$(VAR1)\begin{cases}\dot{\widehat{\delta x}}(t) = A(t)\delta x(t) + B(t)\\ \delta x(t_0) = x_0'(\lambda),\end{cases}$$

with $B(t) = \frac{\partial f}{\partial \lambda}(t, x(t; t_0, x_0, \lambda_0), \lambda_0) \in \mathcal{M}_{n,p}(\mathbf{R})$.

Remarque 3.1. Différentiation interne de Bock We can also approximate $A(t)$ and $B(t)$ by finite difference. But know, as we can use automatic differentiation for computing them, we don't test this possibility.

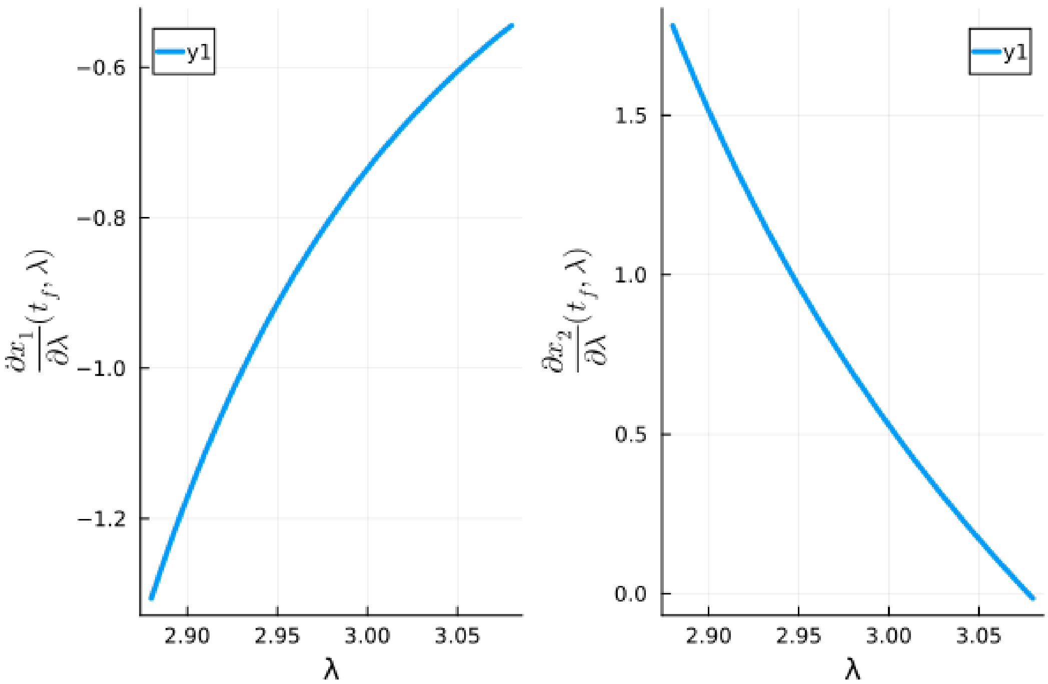

Figure 2: Derivative computing with the variational equation. $t_f = 20, \lambda$ ranging from 2.88 to 3.08, $Tol = RelTol = AbsTol = 10^{-4}$. The numerical integrattion is done with Tsit5().

3.4 Automatic differentiation of the flow

Here, we use automatic differentiation on the function $\varphi(\lambda) = x(t_f, t_0, x_0(\lambda), \lambda)$. This is also known as the Internal Numerical Differentiation of Bock [1]

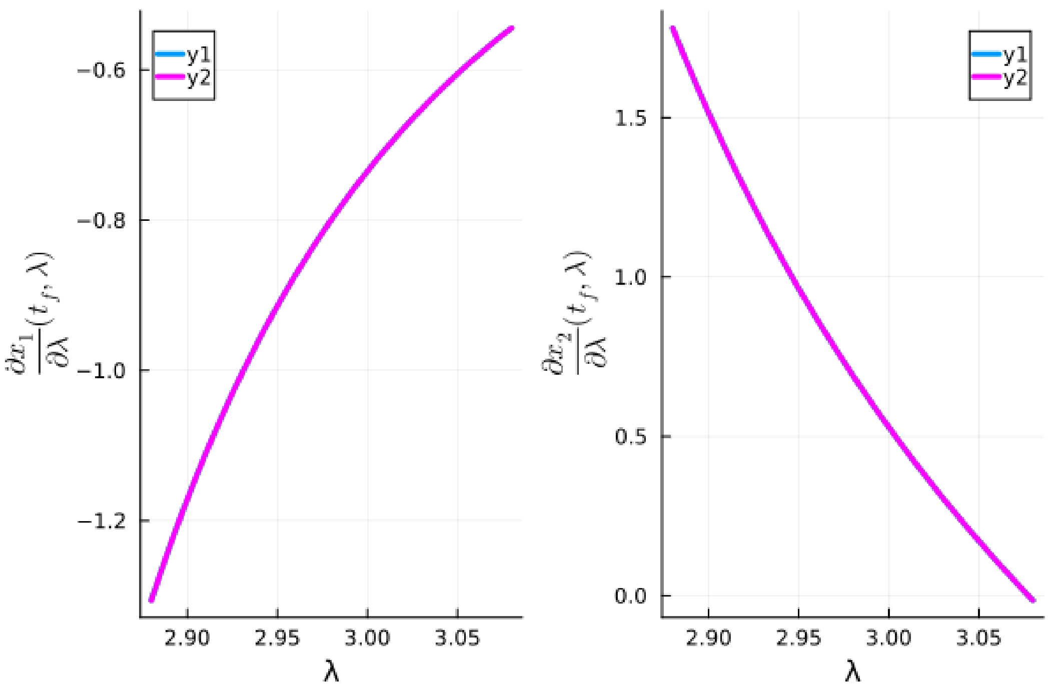

Figure 3: Derivative computing by automatic differentiation of the flow. $t_f = 20, \lambda$ ranging from 2.88 to 3.08, $Tol = RelTol = AbsTol = 10^{-4}$. The numerical integrattion is done with Tsit5(), the automatic differentiation is ForwardDiff.

4 Test on the time steps

5 Conclusion and perspectives

References

[1] H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In K.H. Hebert, P. Deuflhard, and W. Jäger, editors, *Modelling of chemical reaction systems*, volume 18 of *Springer series in Chem. Phys.*, pages 102–125, 1981.

[2] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I, Nonstiff Problems*, volume 8 of *Springer Serie in Computational Mathematics.* Springer-Verlag, second edition, 1993.