WAVES TO WEATHER

Algorithmic Differentiation for Cloud Schemes

Manuel Baumgartner and André Brinkmann
Zentrum für Datenverarbeitung
Johannes Gutenberg University Mainz
manuel.baumgartner@uni-mainz.de

Max Sagebaum and Nicolas R. Gauger
Chair for Scientific Computing
Technische Universität Kaiserslautern

Peter Spichtinger
Institute for Atmospheric Physics
Jannes Gutenberg University Mainz

JG|U JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Introduction

Atmospheric modeling needs to include physical processes on each time and length scale. While a numerical model is able to represent processes acting on time and length scales that are longer or larger than the chosen numerical discretization, it nevertheless needs to include the effect of various non-resolved physical processes. A prominent example of such a subgrid process are clouds, which are not resolved in most models and even dedicated cloud models cannot resolve the microphysical processes of the cloud.
At the moment, no universal governing equation is available to describe the evolution of a cloud across all scales, thus several mathematical formulations of the cloud processes exist in the literature. In particular, these formulations typically contain uncertain parameters, which may be artificial parameters or physical parameters with limited observational evidence of their value. Such parameters introduce uncertainty into the (cloud) parameterization and, consequently, into the whole numerical model. We propose algorithmic differentiation (AD) as a way to identify parameters with largest sensitivities in numerical models and illustrate the technique at the example of a cloud scheme.

What is Algorithmic Differentiation (AD)?

- A (subgrid) parameterization may be thought of as a function f, taking the resolved flow characteristics y together with parameters η, and provides the feedback of the unresolved process $z = f(y, \eta)$.
- The sensitivity of z to a parameter equals the derivative $\frac{\mathrm{d}z}{\mathrm{d}\eta} = \frac{\partial f}{\partial \eta}$.
- AD allows to compute the derivative $\frac{\partial \bar{f}}{\partial \eta}$, where $\bar{f}$ is the implemented version of f, alongside the usual execution of the code.
- AD may be thought of as adding to each code statement an additional statement, evaluating the exact mathematical derivative of the statement at hand. Consequently, the computed derivatives are exact up to machine accuracy.
- For codes written in C++, the tool CoDiPack (Sagebaum et al., 2019) allows the easy inclusion of AD into an existing code.

The Box-Model including the IFS one-moment bulk cloud scheme

- The one-moment warm cloud scheme from the IFS model (ECMWF, 2017) was implemented into a box-model framework in C++, including explicit computation of depositional growth of cloud droplets (Rosemeier et al., 2018; Porz et al., 2018):

$$\frac{\mathrm{d}q_c}{\mathrm{d}t} = \underbrace{c\cdot(S-1)q_c^{\frac{1}{3}}}_{\text{Condensation}} - \underbrace{a_1 q_c^{\gamma}}_{\text{Autoconversion}} - \underbrace{a_2 q_c^{\beta_c} q_r^{\beta_r}}_{\text{Accretion}}, \tag{1a}$$

$$\frac{\mathrm{d}q_r}{\mathrm{d}t} = \underbrace{a_1 q_c^{\gamma}}_{\text{Autoconversion}} + \underbrace{a_2 q_c^{\beta_c} q_r^{\beta_r}}_{\text{Accretion}} + \underbrace{\left(e_1 q_r^{\delta_1} + e_2 q_r^{\delta_2}\right)\min\left(S-1,\, 0\right)}_{\text{Evaporation}} - \underbrace{d q_r^{\zeta}}_{\text{Sedimentation}}, \tag{1b}$$

$$\frac{\mathrm{d}q_v}{\mathrm{d}t} = -c\cdot(S-1)q_c^{\frac{1}{3}} - \left(e_1 q_r^{\delta_1} + e_2 q_r^{\delta_2}\right)\min\left(S-1,\, 0\right), \tag{1c}$$

where q_c, q_r, q_v denote the mixing-ratios of cloud droplets, rain droplets, water vapor, and S represents the saturation ratio.
- All coefficients except for c and all exponents are considered as parameters.
- As an example, we consider the formation of a cloud with an updraft velocity $w = 1\,\mathrm{m\,s^{-1}}$. Figure 1 documents the temporal evolution of the cloud variables q_c, q_r and the saturatiom ratio S.

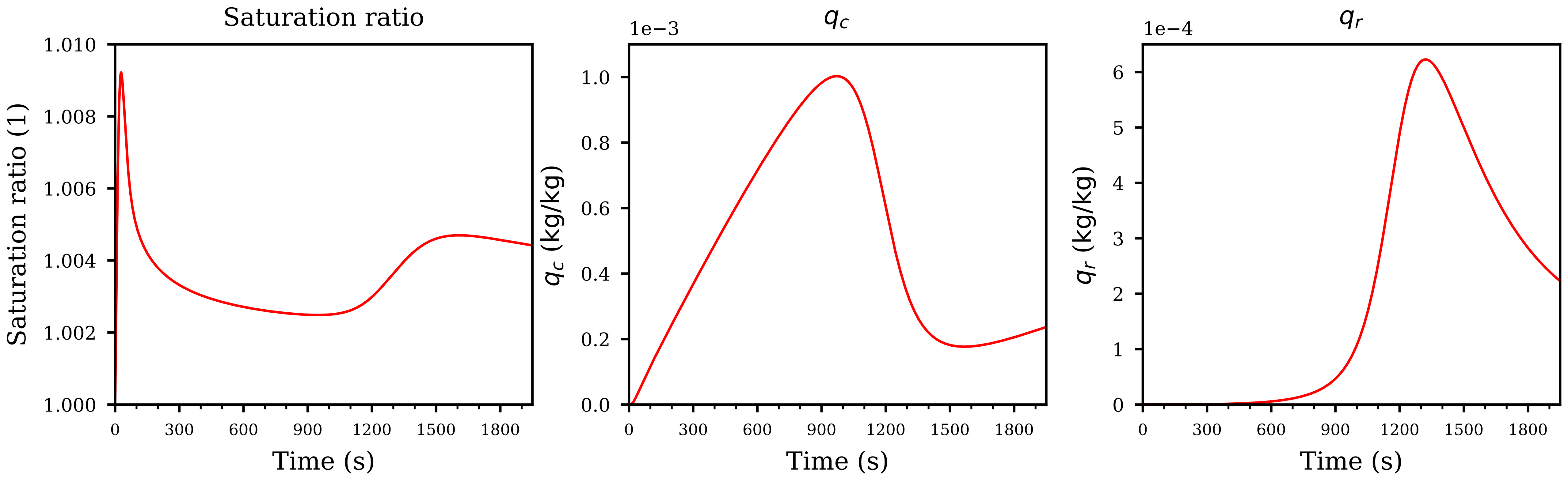

Figure 1: Temporal evolution of the saturation ratio S and the cloud variables q_c, q_r.

Temporal Evolution of the Sensitivities computed using AD

- AD computes the sensitivities of all coefficients and all exponents within each timestep.
- Figure 2 shows the sensitivities of the coefficients in the cloud model (1) for this scenario with respect to cloud droplet mass q_c (left column) and rain droplet mass q_r (right column).
- As an example, the coefficient for autoconversion is most sensitive during cloud formation, see the red curve in Figure 2.
- Although the sedimentation process only affects q_r according to the model equations (1), cloud droplet mass shows an indirect sensitivity to this coefficient, see the purple curve in 2, left column.
- Figure 3 shows the sensitivities of the exponents in the cloud model (1) for this scenario with respect to cloud droplet mass q_c (left column) and rain droplet mass q_r (right column).
- In this case, the most sensitive exponents for cloud droplet mass are the exponents of the accretion process.
- These results elucidate the sensitivity of the cloud variables to the parameter values in the model (1) with only a single model run instead of using an ensemble of runs, because AD computes the sensitivity along the usual program execution.
- AD-computed sensitivities depend on the timestep, but their ratio is independent of the timestep.

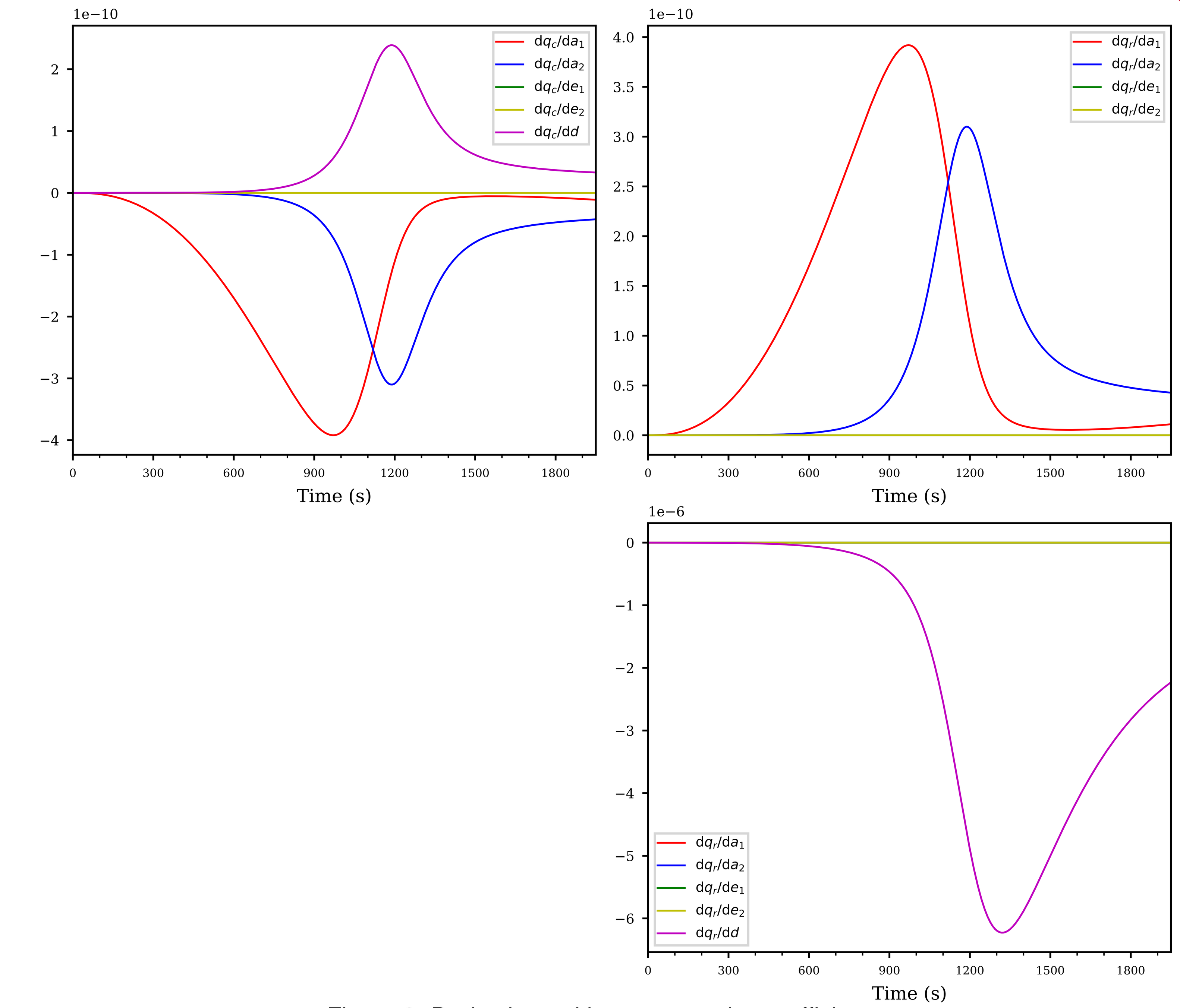

Figure 2: Derivatives with respect to the coefficients.

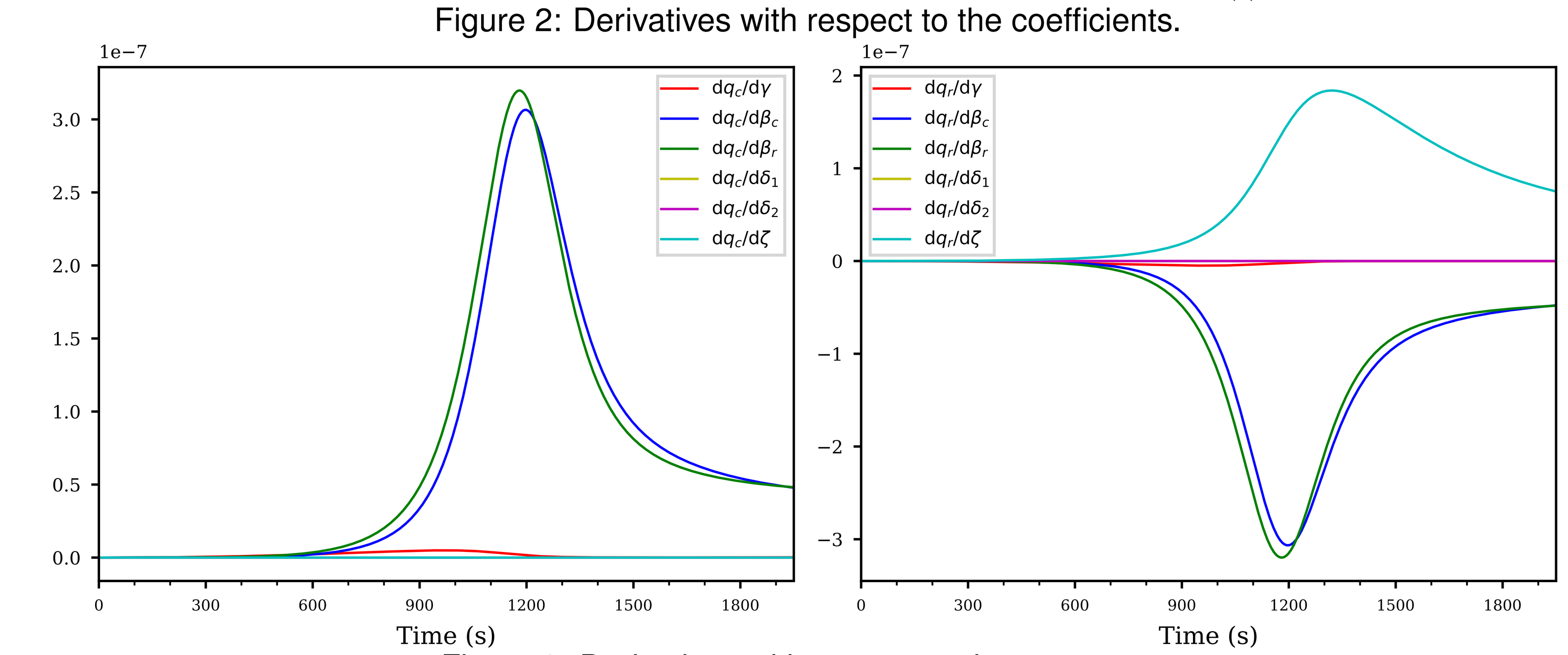

Figure 3: Derivatives with respect to the exponents.

Conclusion

- AD is well suited to compute the sensitivities of parameters within a model, i.e. it computes the sensitivity of the *implemented* code to the parameter.
- AD computes the sensitivities up to machine accuracy and introduces only a constant overhead compared to the program execution without AD. This may be much cheaper than conventional ensemble runs.
- This technique is not restricted to cloud schemes nor to C++, but may be applied to any (subgrid) parameterization although Fortran codes usually need more code modifications to implement AD.

References

Baumgartner, M., M. Sagebaum, N. R. Gauger, P. Spichtinger, and A. Brinkmann (2019). "Algorithmic differentiation for cloud schemes (IFS Cy43r3) using CoDiPack (v1.8.1)". *Geoscientific Model Development* **12**, 12.
ECMWF (2017). *IFS DOCUMENTATION – Cy43r3. Part IV: Physical Processes.*
Porz, N., M. Hanke, M. Baumgartner, and P. Spichtinger (2018). "A model for warm clouds with implicit droplet activation, avoiding saturation adjustment". *Mathematics of Climate and Weather Forecasting* **4**, 1.
Rosemeier, J., M. Baumgartner, and P. Spichtinger (2018). "Intercomparison of Warm-Rain Bulk Microphysics Schemes using Asymptotics". *Mathematics of Climate and Weather Forecasting* **4**, (1).
Sagebaum, M., T. Albring, and N. R. Gauger (2019). "High-Performance Derivative Computations using CoDiPack". *ACM Transactions on Mathematical Software (TOMS)* **45**, 4.