Merkel, Uwe H.; Riehme, Jan; Naumann, Uwe

Reverse Engineering of Initial & Boundary Conditions with TELEMAC and Algorithmic Differentiation

Reverse engineering of initial & boundary conditions with TELEMAC and algorithmic differentiation

U. Merkel
Ingenieurbüro Merkel
Karlsruhe, Germany
info@uwe-merkel.com

J. Riehme, U. Naumann
STCE at RWTH Aachen University
Aachen, Germany
riehme@stce.rwth-aachen.de,
naumann@stce.rwth-aachen.de

***Abstract*—The TELEMAC-SUITE with Algorithmic Differentiation (TELEMAC-AD) calculates the impact of a high number of spatially distributed parameters on flow conditions. This technique is a revolutionary step forward in open channel flow simulations, as it solves previously unsolvable or computationally very expensive problems. Classic simulation methods return the combined impact of many parameters, whereas the adjoint version of TELEMAC-2D returns the individual influence of every single parameter in one run. A wide range of new applications is now possible with the TELEMAC-SUITE: Automatic optimization and calibration of flow relevant shapes, data assimilation, high resolution sensitivity analysis or inverse modelling.**

I. Introduction

This article describes a new method for open channel flow software which solves inverse flow problems. Instead of traditional forward questions like

„Put water in here, how will it be distributed within our project area?“

we solve inverse questions of the type

„We want specific flow conditions here, where and how do we have to modify our project area?“.

See Fig. 1 for a simple example comparison.

- Left: The traditional forward projects few input parameters have influence on many target parameters.
- Middle: The inverse problematic projects many input parameters have influence on a few target parameters.

Many input parameters means, specific values in many geometric points (e.g. bed roughness, bathymetry), or global parameters (e.g. discharge). Each has an individual influence. We want to quantify for all parameters their relevance for the resulting discharge Q. The relevance is described by the gradient of the output Q with respect to the inputs, which is the vector of partial derivatives ∂ Q / ∂ input(i).

If the hydraulic problem is based on a large number of influence factors, then the adjoint model of the TELEMAC-SUITE (TELEMAC-AD) is the most efficient solver: The complete gradient can be computed by one run of the adjoint model. TELEMAC-AD was generated semi-automatically by the differentiation-enabled NAG FORTRAN compiler, a joint development of the institute STCE at RWTH Aachen University, Germany, NAG, UK, and the University of Hertfordshire, UK.

After a short introduction (Chapter 1) two examples illustrate the potential of TELEMAC-AD (Chapter 2). A basic introduction of the AD concept follows in (Chapter 3). More detailed Information can be found in [1] and [2].

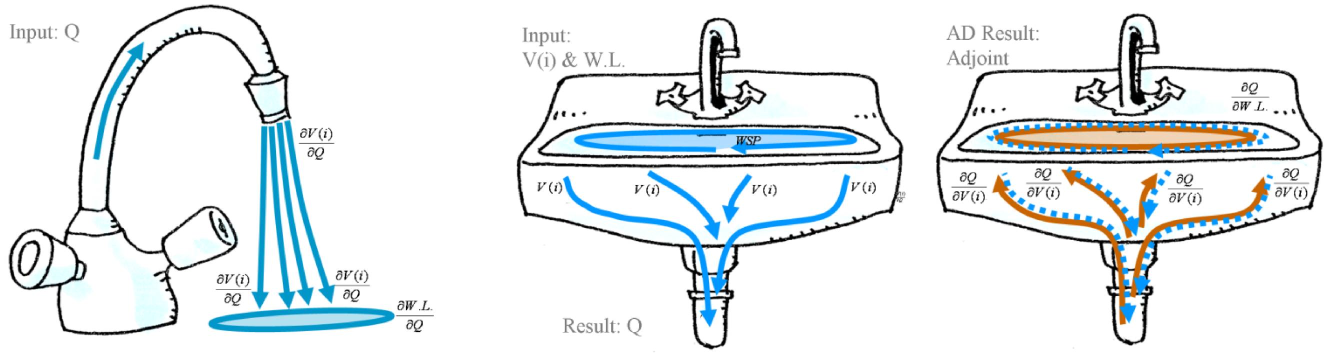

Figure 1. Left: common forward problem: Few Parameters influence many, dependency easy to calculate. Middle: inverse problem. Few parameters depend on many, sometimes millions of geometry points. Both, physical and numerical models overlay the single dependencies on the results. Right: The adjoint method solves until now unsolvable inverse problems by interpreting the forward problem backward.

A. A wide field of new applications

This new method opens a door to a new range of hydraulic models. Typically applications are the impact quantification for any point of a TELEMAC-2D -3D or SISYPHE model.

Any point *I* (and its connected parameters velocity V, water level W.L., manning's n ...)

- influences the flow conditions at a power plant intake and the maximum energy level which results in more or less electrical energy.
- influences robustness and sensitivity of hydraulic relevant structures and hydraulic driven processes of all kinds (like dams, bridges, flood protection measures, morphological aspects, a.o.) [3].

In combination with gradient based optimization methods the adjoint technology can be used for

- automatic calibration of thousands of roughness values at the same time, e.g. to fit water levels to measured values (until today an unsolved every day problem).
- semiautomatic modification of single point coordinates to modify flow conditions according to a target function. Airfoils and drag coefficients of cars have already been optimized with AD in mechanical engineering (shape optimization).

B. Using the classic forward calculation

The influence of boundary & initial conditions like W.L. and V(i), i=1...N, on a target parameter like Q for questions as in Fig. 1 (middle) is described with the gradient (left part of formula 1):

$$\nabla Q = \left[\frac{\partial Q}{\partial V(1)},...,\frac{\partial Q}{\partial V(N)},\frac{\partial Q}{\partial W.L.}\right]^T \approx \left[\frac{\Delta Q}{\Delta V(1)},...,\frac{\Delta Q}{\Delta V(N)},\frac{\Delta Q}{\Delta W.L.}\right]^T \quad (1)$$

Until today the differential analysis is the common way to analyze and optimize the flow field or morphology (right part of formula 1). Every element of the gradient approximation has to be computed separately with slightly different input parameters. For millions of points millions of calculations are necessary or the points have to be grouped, what will bias the results.

Since the early days of physical flow models the real target is to minimize the difference between a real and a desired flow field (maximum erosion, water levels, velocity field etc.). The main question in many cases is that one few parameter (the difference) depends on many.

This can be any mathematically relevant variable, but especially for open channel flow it is the spatial distribution of parameters in big FE meshes. But an analysis for millions of points (each one is a bathymetric feature) is too expensive normally.

C. Backward interpretation with adjoint models

The adjoint model of TELEMAC-AD computes the entire gradient from formula 1 in only 1 forward run and its following backward interpretation.

In the adjoint mode, TELEMAC-AD records every relevant instruction during a forward evaluation in the so called “tape”. At the end of the forward run all target values and the process flow is stored in the tape. The backward interpretation propagates the adjoints (derivatives) from the target parameters (at the end of the tape) to the input parameters (at the beginning of the tape). During this process every single instruction is interpreted by its adjoint version. The adjoints obtained for the input parameters can now be used as dependency information, for robustness or sensitivity analyses or in a next step for gradient based optimization methods.

II. EXAMPLES

Two examples with TELEMAC and SISYPHE illustrate the potential of the AD technology. Both are based on the open source version of TELEMAC v6p2.

- A morphodynamic 2D SISYPHE flume model with 92 roughness zones is automatically calibrated.
- A hydraulic 2D TELEMAC river model is examined to quantify the influence of 95000 spatially distributed parameters on the shear stress in 1 specific point.

Further example cases can be found at [3] and www.uwe-merkel.com/TELEMAC-ad.

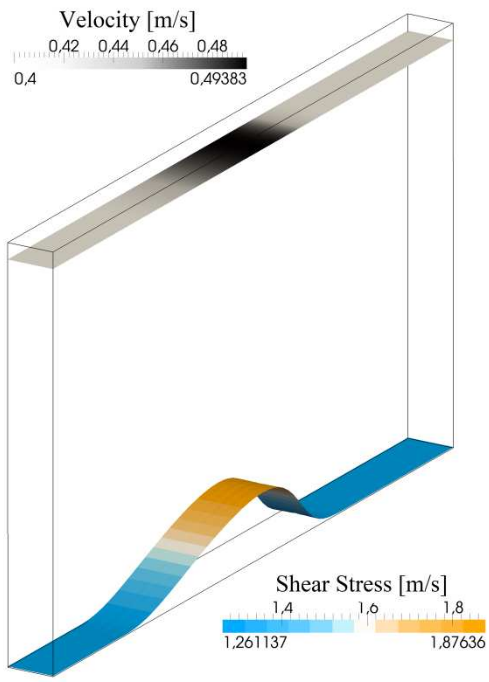

Figure 2. Setup of the flume: LxWxH: 16m x 1,1m x 0,6m; dune height: 0,1m; runtime: 14400 s

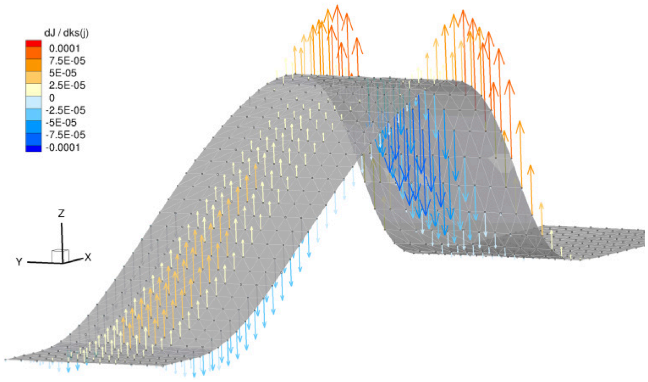

Figure 3. Arrows show direction and magnitude for the influence of a mesh points roughness zone on the cost function (sum of quadratic errors).

A. Automatic calibration of 92 morphological parameters

As introduction example serves a simple lab flume. Adjoints are calculated with the SISYPHE-AD version, and they are used to perform a simple Least Square Root (LSQR) optimization. 92 zones have been defined as relevant to the resulting evolution. Fig. 2 shows the setup, a straight flume with moving sediments in a dune shape. 92 local zones are defined for the roughness ks_j, $j=1,\ldots,92$, which itself is a combination of the boundary roughness and the grain roughness. The calculated morphological evolution $E_i = E_i(ks), i=1,..,891$ has to be optimized to match the observation values in all mesh points i.

A LSQR problem is formulated by introducing a cost functional J:

$$J = \sum_{i=1}^{891} (E_{OBS,i} - E_{IST,i})^2 \tag{2}$$

The adjoint model computed the gradient of J with respect to all roughness coefficients ksj, $j=1,\ldots,92$:

$$\nabla_{ks} J = \left[\frac{\partial J}{\partial ks_1}, \ldots, \frac{\partial J}{\partial ks_{92}} \right]^T \tag{3}$$

Therefore the adjoint version of SISYPHE calculates the evolution E(i) values after 14400 s as usual, and the cost functional J in addition. Backward interpretation calculates the gradient from (4) describing the dependency of J on ks_j.

Fig. 3 shows the 92 elements of the gradient as arrows along the 891 mesh points. The arrows show direction and magnitude of the dependency and must not be misunderstood as flow vectors.

In the next step this information is used to fit the calculation results to the observation results:

The observation result in this case is a SISYPHE simulation itself, whose final evolution is set as the ideal case. This is called a twin experiment, which allows evaluating the behavior of the optimization algorithm without the influence of measurement uncertainties or design wishes that might turn out as utopia. The twin experiment suits best for validating SISYPHE-AD and the optimization algorithm.

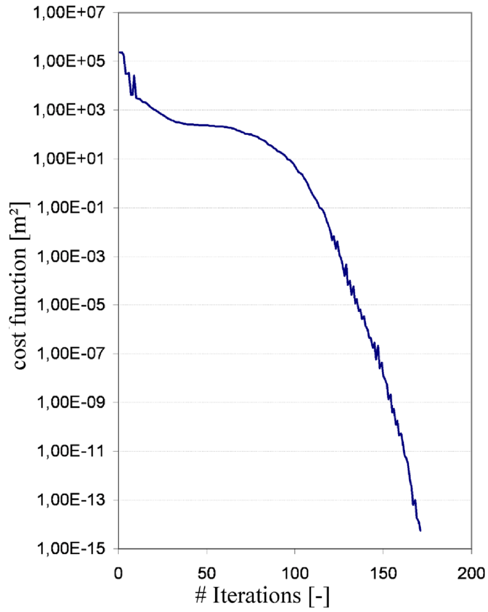

Figure 4. The adjoint based optimization of the roughness parameters, based on the grain diameter, converges until the cost function drops below J < 10-14 (Roughness: Strickler; Algorithm: BFGS).

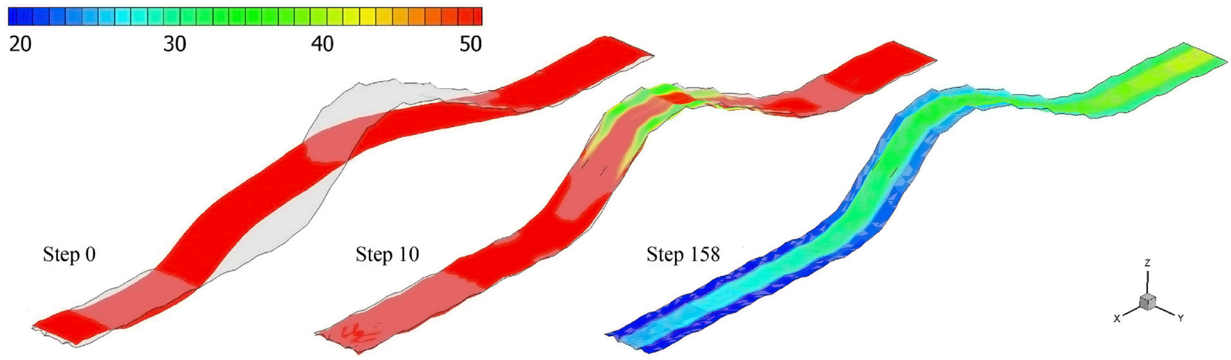

Figure 5. Development of the cost function during the optimization (Initial state, 10 steps, 158 steps): Grey shaded; optimization target. The color indicates the development of the roughness which respects grain size and boundary influence.

The target function J is reduced in each iteration step, which means that for each step a full SISYPHE-AD run is executed and its resulting gradient is used in a wrapping minimization program. This mantle program calls CG, BFGS or SLSQP algorithms, which are part of the MINPACK optimization library [10]. The optimization terminates if the cost function goes below a threshold or doesn't converge. The master thesis of Monika Schäfer focuses on the performance of these algorithms [4].

Fig. 4 shows the progress of the optimization, Fig. 5 the development of bathymetry and roughness during the optimization process. The observation bathymetry (transparent gray) is converging very fast, while the roughness (grain size roughness) is developing slower. But after 158 full calculations of SISYPHE-AD the cost function J is below the terminating threshold 10-14.

B. Dependency of shear stress in a groyne head scour

221_Donau is a public validation example for TELEMAC-2D and it is available at OPENTELEMAC.ORG. It is used here to analyze how the surrounding bathymetry and the neighborhoods roughness distribution influence flow conditions in a certain point of a real river. The shear stress τ (taken from 12 points in the middle of the groyne head scour, see marker in Fig. 6 is the target of this analysis.

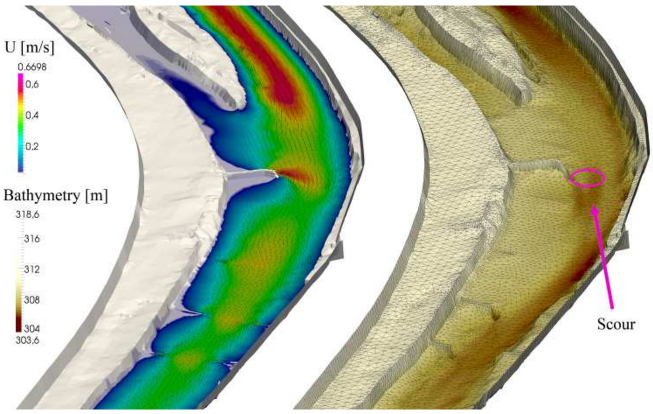

Figure 6. Perspective view of the 221_Donau model, with a dominant groyne and its head scour. Right: Geometry defined with approx. 47500 mesh points. Left: Scalar velocities.

In difference to the first example this real world example has highly complex multidimensional flow conditions with islands, groynes, tidal flats, and many other flow interacting features. The spatial interaction is broken down to local values for every point with TELEMAC-2D-AD, here with focus on the roughness and elevation values of the 47500 points in the 2D mesh. The dependency is individually quantified in total for 95000 parameters. This example is chosen to proof the usability for practical purposes.

1) Classic solution with finite differences (FD):

Solving per point independent dependency analysis requires a base calculation and 95000 full TELEMAC-2D calculations with a slight modification of one input parameter. Only this will proof spatial independent input-output dependencies.

See formula 4 and 5:

$$\frac{\delta\tau}{\delta z(i)} \approx \frac{\tau(z(i)+h)-\tau(z(i))}{(z(i)+h)-z(i)} \tag{4}$$

$$\frac{\delta\tau}{\delta ks(i)} \approx \frac{\tau(ks(i)+h)-\tau(ks(i))}{(ks(i)+h)-ks(i)} \tag{5}$$

One calculation runs 5min on a up to date desktop computer, this means 330 days for all calculations or a very expensive outsourcing on a bigger cluster.

And the result is only valid for:

- one parameter set (discharge, bathymetry, roughness, turbulence setting ...)
- the simulated time span
- very limited extrapolation, due to the nonlinearity in many sub models (see Fig. 8!).

Practical projects usually observe many variants, optimization projects even more, which leads to very expensive computational costs, making this technology economically unusable for most small and medium size projects.

2) Adjoint solution (AD):

TELEMAC-AD calculates the full forward run and backward interpretation in 678 min on the same desktop computer, and returns all 95000 adjoints for the shear stress τ. The gain of computational speed equals the usage of an approx. 1000-core cluster when using the classic FD method. Some results are displayed in Fig. 7. At the time of writing TELEMAC-2D-AD is still not parallelized, and not optimized for speed, which means that a further speedup is expected after completion of these ongoing developments.

3) Interpretation of the resulting adjoints:

Adjoints computed by TELEMAC-2D-AD describe the change of the output τ as a linear relation of its specific input. Therefore extrapolations for other input parameter sets can only be done with great care.

The dependency of the bathymetry on τ at the simulated flow conditions is dominated by the obvious separation of the 2 arms around the upstream island. If the southern arm, which has low flow, is lifted, then more water is pushed to the main channel (1). The same happens if the surrounding of the groyne and the opposite site (2) of the cross section are elevated. A kind of funnel effect increases the shear stress. Decreasing the scour itself increases the shear stress as well (3).

The perspective view from downstream (4) reveals that some other groynes have a high impact, as they influence with their back draft the water level in the examined area. The lower groynes, which are smaller and shaded by the bigger ones therefore don't influence the examination zone anymore. Again the reader shall be warned that only a change of few decimeters in any topographic feature might change the result totally. The general noise in the adjoints origins from bumps and holes in the bathymetry. Following the direction of the adjoints will smoothen the main channel and therefore accelerate flow and increase the shear stress.

On the contrary for bed roughness a clear tendency is visible: the smoother the bed along the main channel, the higher the shear stress at the groyne head scour. If the roughness gets higher on the opposite shore (5), more water is pressed to the scour. Classic methods (numerical and physical) would have given a rough idea about this dependency, but for the first time a hydraulic model can exactly define the spatial limits of the relevant area. The scour itself has a different tendency (6): Increasing the roughness in the target point obviously increases the local shear stress.

III. ALGORITHMIC DIFFERENTIATION

Algorithmic differentiation (AD) is a mathematical method that extends existing computer programs in a way that for a priori chosen results their dependency to a priori chosen input variables is additionally calculated. AD tools work on the original source code, and not as a new implementation of the mathematic model. Differentiated models obtained by AD compute derivatives in machine precision for a given input parameter set.

Since some time aerospace, mechanical engineering and meteorology are working successfully with adjoint models. They use AD for uncertainty quantification, data assimilation, optimization strategies and inverse problems. Inverse problems in hydraulic engineering are for example the quantification of boundary and initial conditions for given results.

The principles of AD are based on the fact, that every calculation is finally a combination of basic operations (+, -, *, /, exp, sin,...) with well known differentiation rules. All more complex formulas are a sequence of these; derivatives of basic operations are combined by the chain rule to derivatives of sequences.

Two basic techniques are used for first order derivatives:

- Tangent linear models (forward models) work the same direction as finite difference (FD) approximations of derivatives, but at machine precision. Fig. 8 shows this advantage against FD, which has to use 2 calculation results for each derivative.
- Adjoint models (inverse interpretation of the forward model) propagate the adjoint (derivative) from the final results back to initial and boundary conditions. For the backward interpretation the full path of the forward calculation has to be recorded. If the forward model has just a single target parameter value, the vector of derivatives (the gradient) can be calculated in only one backward interpretation.

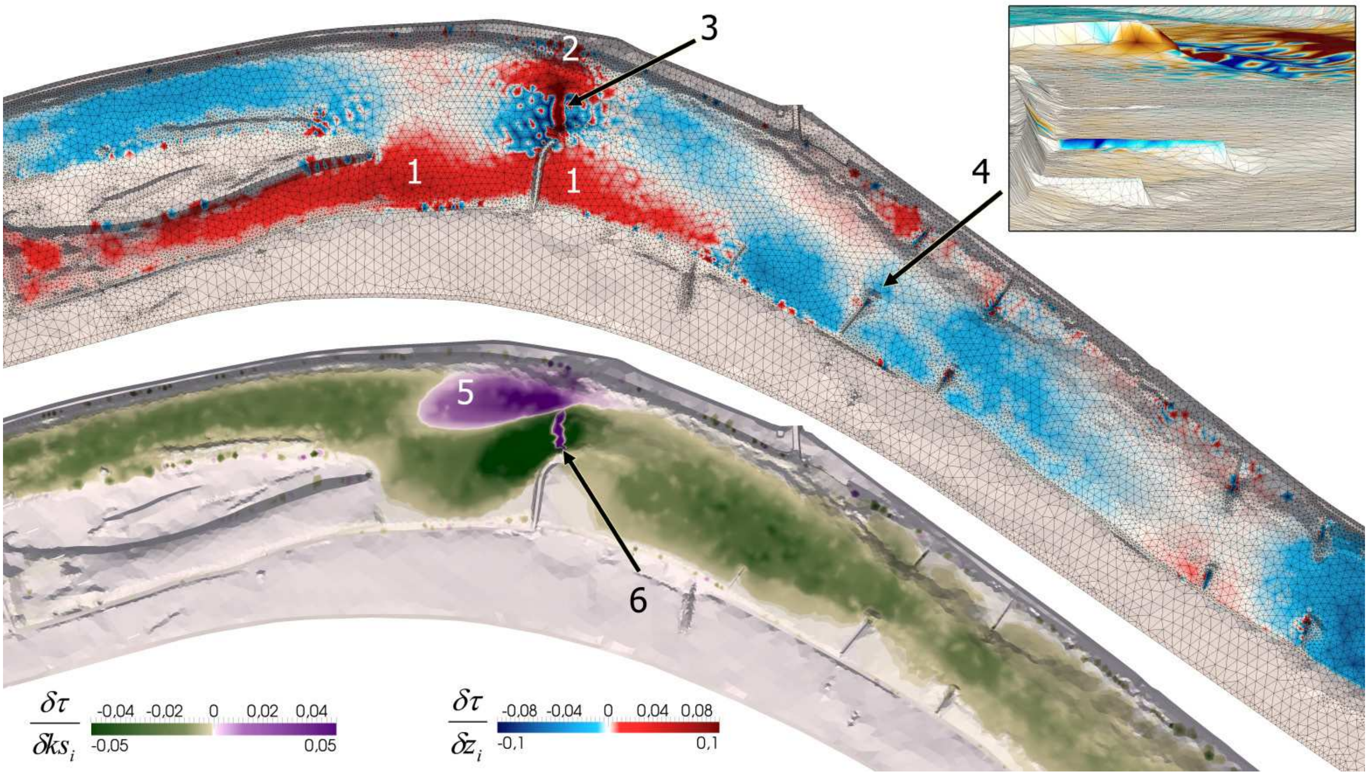

Figure 7. Result of a single TELEMAC-2D-AD calculation. Top: Dependency of the shear stress in the scour (marked) to neighbouring geometry information. Bottom: Influence of the roughness on the shear stress [values per m²]. Values are only valid for the current flow conditions and setup!

Second or higher order derivatives can be calculated by combining the basic techniques. Higher order derivatives might speed up optimization processes significantly. For further information see [5].

The AD-enabled NAG Fortran Compiler [7], developed at the Institute"Software and Tools for Computational Engineering" (STCE, University RWTH Aachen) is a commercial extension of the NAG FORTRAN compiler (Numerical Algorithm Group, Oxford, UK). It uses a hybrid technique of source code transformation and an efficient overloading based run time library. [6] discusses this methods in detail. The hybrid approach allows an efficient differentiation of large projects like the TELEMAC-SUITE.

For practical usage the recording of all iterations and all operations within these iterations is a memory consumption problem. For the 221_Donau example 10TB of RAM would be necessary. Therefore TELEMAC had to be extended by a so called checkpointing technology. What means that every calculation can be saved to RAM and restarted from RAM with binary identically results at any point in time, with only a minimum recording of variables.

For the backward propagation of the adjoints (backward time step wise) the necessary detail information about subprogram internal operations is recalculated from the checkpoints in reverse order. This increases calculation effort by 200%, but reduces the RAM usage, as the minimum checkpointing system only needs 300MB for 1000 time steps in the 221_Donau example, plus 10GB for the current time step.

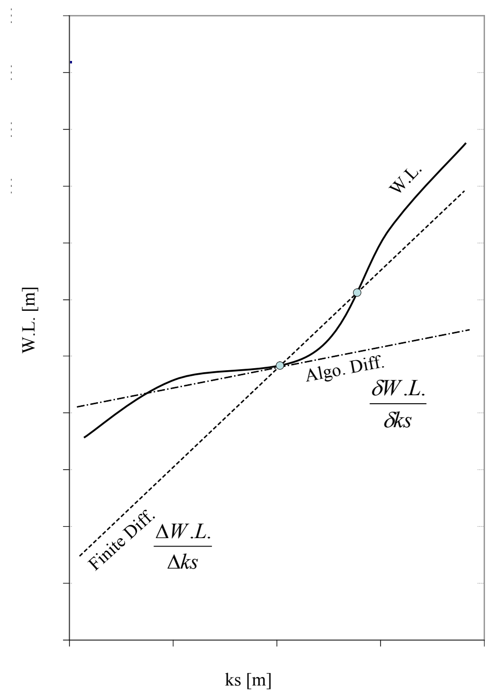

Figure 8. Water level as a function of the roughness ks. Normally the dependency is not linear. Dependency is described as first derivation, this means the ascent of the tangent is defined in only one point for algorithmic (~analytic) differentiation and in two for finite differences.

IV. CONCLUSIONS

The methods of reverse interpretation and algorithmic differentiation enable a very fast quantification of dependency gradients with millions of influence parameters! The dependency of any numerical result (energy, evolution, transport rates, risk ...) on any numerical input value (spatial or global) can be quantified independently within only one run of the adjoint model. For hydraulic modelling especially the ability to dissolve spatial interactions opens a gate to a new generation of models, which solve until now unsolved problems. Spatial independent sensitivity and robustness information helps to understand complex flow situations and can be used for gradient based optimization processes [3]. This method is currently unique for open channel flow software, and the growing number of new examples will be continuously updated on www.uwe-merkel.com/TELEMAC-ad.

This first project was funded by the BAW with participation of the EDF. Their aim is to have a tool that dissolves the interaction between morphology, flow and constructions.

REFERENCES

[1] Villaret, C. 2013. Hervouet, J.M. Kopmann, R. Merkel, U. Morphodynamics modelling using the TELEMAC finite-element System. Computers and Geosciences. Heft 53, S.105-113. (2013)

[2] Kopmann, R. Merkel, U. Drei Jahre TELEMAC Open Source. In: Wasserwirtschaft (2013), Heft 12

[3] Kopmann, R. Merkel, U. Riehme, J. Using Reliability Analysis in Morphodynamic Simulation with TELEMAC2D / SISYPHE. Proceedings: TELEMAC Users Club 2012, Oxford, (2012)

[4] Schäfer, M. Optimierungsalgorithmen für LSQR Optimierungsprobleme mit dem algorithmisch differenzierten TELEMAC-2D AD. Diplomarbeit (2013)

[5] Griewank, A., Walther, A. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. 2d ed. SIAM. Philadelphia. . (2008).

[6] Rauser, F. Riehme, J. Leppkes, K. Korn, P., Naumann, U. On the use of discrete adjoints in goal error estimation for shallow water equations. Procedia Computer Science, 1, 107–115.(2010).

[7] AD-enabled NAG FORTRAN compiler: Users manual. http://www.stce.rwth-aachen.de/files/nagfor_ad/userguide_v0.2.pdf (03/2013).

[8] Naumann, U. The Art of Differentiating Computer Programs. An Introduction to Algorithmic Differentiation. Software, Environments, and Tools, SIAM, (2012).

[9] Naumann, U., Riehme, J. A differentiation-enabled Fortran 95 compiler. ACM Transactions on Mathematical Software, 31(4):458–474. (2005).

[10] Nocedal, J. Wright, S.Numerical Optimization. 2nd ed. Springer. (2006).