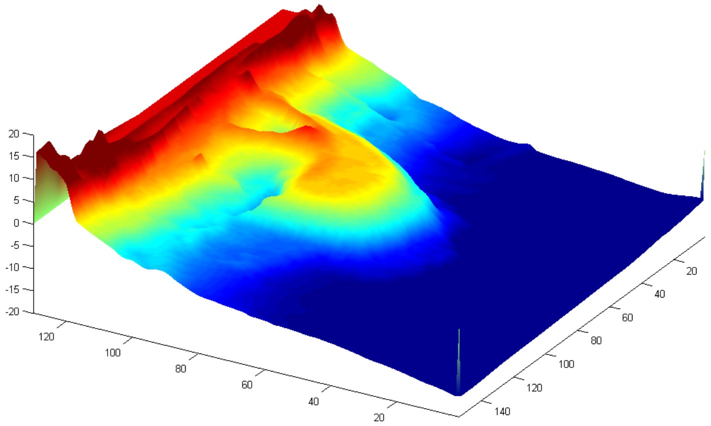
DEVELOPMENT OF A GENERIC AUTOMATED INSTRUMENT FOR THE CALIBRATION OF MORPHODYNAMIC DELFT3D MODEL APPLICATIONS



MSc. Thesis
R.W. Hasselaar
6 July 2012

DELFT, July 6, 2012

DELFT UNIVERSITY OF TECHNOLOGY (TUD)
FACULTY OF CIVIL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE (NUS)
DEPARTMENT OF ENGINEERING, FACULTY OF CIVIL ENGINEERING

DELTARES
ENABLING DELTA LIFE

<table>
<tr><td><b>SUPERVISORS</b></td><td>MARCEL STIVE</td><td>TU|DELFT</td></tr>
<tr><td></td><td>ARJEN LUIJENDIJK</td><td>DELTARES / TU|DELFT</td></tr>
<tr><td></td><td>VLADAN BABOVIC</td><td>NUS</td></tr>
<tr><td></td><td>SENG KEAT OOI</td><td>NUS</td></tr>
<tr><td></td><td>MARTIN VERLAAN</td><td>DELTARES / TU|DELFT</td></tr>
<tr><td></td><td>STEF HUMMEL</td><td>DELTARES</td></tr>
</table>

CONTACT
ROBERT HASSELAAR
RWHASSELAAR@GMAIL.COM
+316 15 38 63 97

PREFACE

This Masters thesis has been undertaken to wrap up the double degree program of hydraulic engineering and water resources management. This program is taught at the Technical University of Delft as well as at the National University of Singapore.

At the end of my stay in Singapore, I applied for a graduate internship at Deltares. Although this original plan failed as my assignment suddenly disappeared, I did end up at Deltares to produce my Master thesis. Together with Arjen we developed a new assignment, which had everything in it that I was looking for at Deltares; computers, modeling and some more computing, which I very much enjoyed completing.

This report has been written for an audience that is familiar with basic coastal morphological concepts. Furthermore, this report assumes basic knowledge on numerical models and its applications.

I would like to thank Deltares for providing me with a place to work among my fellow students in the feestruimte. Many thanks go out to my supervisors at Deltares, who always made time to answer my questions. A special thanks goes out to Stef, without whom I would have never discovered the realm of java. Finally, I would like to thank my fellow students at Deltares for all the good times; Talent team for life!

Delft, July 2012

SUMMARY

The numerical flow model Delft3D simulates the flow of water and movement of submerged sediment by means of solving simplified equations that describe these processes. Because of these simplifications, parameterizations are introduced. These parameters represent amongst others the processes that take place on scales smaller than the spacing of the numerical grid. Furthermore, parameters exist that represent measurable attributes of the system, such as the grain diameter. Lastly, scaling parameters are found in the model, with which the relative importance of certain processes in the model can be indicated. All these parameters have to be calibrated using measured data; a model application has to be tuned to produce the best fit with a set of measurements.

Although calibration is an essential part in the development of any numerical flow model, no automated or standardized calibration approaches exist within Deltares. As many parameters are available within Delft3D, calibration can be a complex, time consuming process. Automated calibration is objective and can save time, whilst in the process improving the model performance; i.e. minimizing the error between the model results and the measurements. Objective of this Thesis therefore was to develop an efficient method to improve performance of and insights in the Delft3D model throughout complex morphodynamic applications.

A careful reading showed that a generic open source platform for calibration of hydrodynamic Delft3D model applications already existed in the form of the software package OpenDA. This software systematically alters parameter values and compares the corresponding model results with the measurements provided. The results are judged by means of a cost function; a performance indicator which represents the goodness of fit between the model results and the measurements. OpenDA was however not yet applicable on morphodynamic model applications. Therefore, the software was adjusted, resulting in an upgraded version of OpenDA; OpenDA MOR. Apart from adding morphological parameter and -result readers, two additional performance indicators were implemented in the code; the Brier skill score and Kirchhofer scores. OpenDA MOR is suited for both calibration and sensitivity analysis of morphodynamic model applications.

The OpenDA MOR calibration instrument has been tested by means of TWIN experiments. From these tests, rules of thumb have been deduced on how to apply the tool. It was shown that a calibration was most likely to succeed when no more than 2 parameters were calibrated at once, these parameters are non-inter-related, the parameter(s) are sensitive and that the initial values of the parameters are within a 75% range of their optimum values.

An extensive sensitivity analysis was performed on 17 parameters using four model applications. A large overlap in sensitivity throughout the model applications was found; i.e. the same parameters were sensitive in all model applications. Furthermore, from the parameter sensitivity, conclusions could be drawn on the relative importance of the different processes in the various model applications. Lastly, the sensitivity analysis has resulted in insights in the inter-relationships that exist between the various model parameters.

After successfully testing the calibration instrument, different calibration cases have been set up to investigate two questions; can OpenDA MOR pinpoint sensitive parameters automatically? Do optimum parameter values differ for different model applications and transport formulations? It was found that automatic pinpointing was not possible, implying that a sensitivity analysis has to be performed separately before starting a calibration. From the second calibration case it became clear that different optimum parameter values were found depending on the model application and transport formula used. This shows the importance of calibration and that there is no such thing as a universal best suited calibration strategy. What parameters should be calibrated depends completely on the goal of the model; perfectly calibrated hydrodynamic parameters do not necessarily produce the best morphodynamic results and vice versa.

The test results and calibration cases have shown that the calibration instrument OpenDA MOR has successfully been developed and is applicable on any morphodynamic model application. However, the applicability on complex models was shown to be difficult from a practical point of view, because of the very long runtimes of these models. Therefore, further research is needed on how to decrease the runtimes of complex model applications. Moreover, a new cost function should be developed which enables a more accurate judgment of the morphodynamic model results, as it was found that the Brier skill score and Kirchhofer scores are not suited to replace the standard cost function used during this Thesis.

TABLE OF CONTENTS

PREFACE ... I

SUMMARY ... II

TABLE OF CONTENTS ... IV

1 INTRODUCTION ... 6

1.1 PROBLEM DEFINITION ... 7

1.2 OBJECTIVE AND RESEARCH QUESTION ... 7

1.3 STUDY APPROACH ... 8

1.4 LAYOUT REPORT ... 8

2 DELFT3D MODEL, PARAMETERS, FORMULATIONS AND MODEL APPLICATIONS ... 10

2.1 DESCRIPTION OF THE NUMERICAL MODEL USED IN THE CALIBRATION PROCESS ... 10

2.2 DESCRIPTION OF FREE MODEL PARAMETERS ... 11

2.2.1 Parameters related to hydrodynamic processes ... 12

2.2.2 Parameters related to morphodynamic processes ... 14

2.3 SEDIMENT TRANSPORT FORMULATIONS ... 16

2.4 OVERVIEW OF UTILIZED MODEL APPLICATIONS ... 19

3 DEVELOPMENT AND TESTING OF AN AUTOMATED CALIBRATION TOOL ... 23

3.1 WORKINGS OF THE OPENDA (OPEN DATA ASSIMILATION) SOFTWARE PACKAGE ... 23

3.1.1 General overview main components OpenDA ... 23

3.1.2 Overview of workings optimization algorithm ... 26

3.1.3 Configuring the OpenDA components ... 28

3.2 ALTERATIONS MADE TO THE OPENDA SOFTWARE PACKAGE ... 29

3.2.1 Development and application of additional java readers ... 29

3.2.2 Implementation of the decomposed Brier skill score ... 30

3.2.3 Implementation of the Kirchhofer method ... 39

3.3 PREVIEW OF OUTPUT FROM ADJUSTED SOFTWARE ... 41

3.4 VALIDATION OF THE CALIBRATION TOOL: TWIN EXPERIMENTS ... 42

3.4.1 Setup of TWIN experiment ... 42

3.4.2 Additional output from TWIN experiments ... 43

3.4.3 Results of single and multiple parameter TWIN experiments ... 45

4 APPLICATION OF AN AUTOMATED CALIBRATION TOOL 49
4.1 SENSITIVITY ANALYSIS OF MODEL PARAMETERS & TRANSPORT FORMULATIONS 49
4.1.1 Setup of sensitivity analysis 49
4.1.2 Results of sensitivity analysis 51
4.2 CALIBRATION CASES 55
4.2.1 Can OpenDA Pinpoint sensitive parameters automatically? 55
4.2.2 Calibrating two parameters using different models applications 58
4.2.3 Calibrating the Sand engine model 64
5 DISCUSSION OF RESULTS TESTS & CALIBRATION RUNS 67
5.1 LESSONS LEARNED FROM TWIN EXPERIMENTS 67
5.1.1 Factors determining success of TWIN experiment 67
5.2 ABSOLUTE AND RELATIVE PARAMETER SENSITIVITY 68
5.2.1 Why are parameters (non) sensitive? 68
5.2.2 Why does the parameter (non)-sensitivy differ throughout model applications? 69
5.2.3 Implications of parameter sensitivity 72
5.2.4 Parameter inter-relationships 72
5.3 REVIEW OF CALIBRATION CASES 73
5.4 REVIEW OF PERFORMANCE PARAMETERS 74
5.4.1 Review of decomposed Brier skill score 74
5.4.2 Review of Kirchhofer method 75
6 CONCLUSIONS 76
7 RECOMMENDATIONS 78
8 BIBLIOGRAPHY 79
APPENDIX A RESULTS SENSITIVITY ANALYSIS 80
A.1 EGMOND MODEL 80
A.2 TRENCH MODEL 83
A.3 BASIN MODEL 86

1 INTRODUCTION

Numerical models simulate parts of reality by means of solving simplified equations that describe the physical processes in reality. Because of these simplifications and the lack of knowledge about (parts of) these processes, parameters are introduced which contain the uncertainty of the system. These so called parameterizations are therefore a consequence of a lack of information on and knowledge of these complex processes that the model tries to reproduce. Even when processes are fully understood, a lack of computational power often still leads to simplifications and the introduction of parameters. These model parameters however do not necessarily represent measurable attributes of the system that is being simulated. A calibration procedure must therefore be executed to be able to determine suitable values for these parameterizations; in other words, models have to be tuned using measured data.

A typical calibration procedure consist out of three steps, which are repeated until the simulation results are deemed accurate enough; running the model, crosschecking the results against actual measured data and if necessary, adjusting the models parameters.

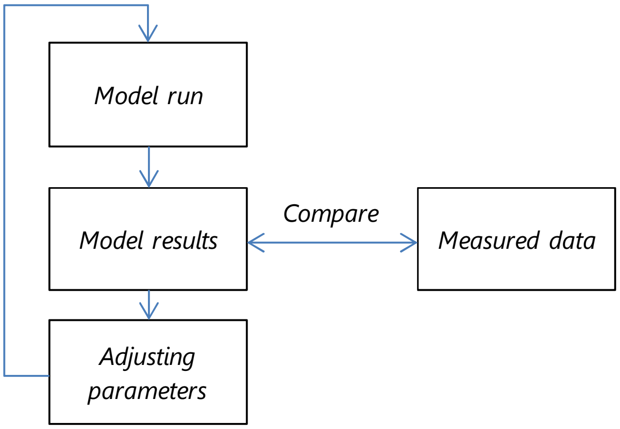


Figure 1-1 | Three steps in a typical calibration procedure

The simulation results can be compared against both measured data, as well as against data from other simulation results. In case simulation results are compared with results obtained with other parameter values, knowledge is gained about the sensitivity of the different parameters in the model; a sensitivity analysis.

Many projects that have been executed in the past by Deltares have required some kind of calibration of the numerical models that were used. Calibration is an essential part of the development of numerical flow model applications. Standardized approaches to calibration however do not exist. Deltares could therefore profit from the development of an automated calibration instrument. Standardized approaches for calibration can increase insights in the physical processes underlying the numerical model, whilst in the process improving the models results.

Careful readings in the field of calibration have shown that a very suitable software package for automated calibration exists; the Open Data Assimilation software package (OpenDA). OpenDA has been developed by Deltares and provides an open source environment for calibration and data assimilation. Although at first glance very much suited for the purpose of this thesis, OpenDA has not yet been applied in the field of morphodynamics.

Recent communications between the University of Reading and Deltares do show the need for a platform for data assimilation and calibration of morphological Delft3D model applications. Both parties can benefit from one another if these two institutes join forces; it would provide Deltares with a great opportunity to showcase the OpenDA software package. Furthermore, large datasets will be made available by the university, providing a unique chance of putting the software to the test. The university would be handed a great tool for calibration and data assimilation and could be guided by the experts of Deltares.

1.1 PROBLEM DEFINITION

Ideally, a calibration strategy for morphodynamic models consists of the following steps, executed in the order presented; calibration of the wave parameterizations, calibration of the flow module's parameterizations and finally, calibration of the parameters involved in the morphology module of Delft3D. The best modeling strategy largely depends on the processes that one is interested in. The choice on what data to calibrate is completely determined by the goal of the model. Therefore, there is no such thing as a universal best suited calibration strategy.

There is however a big difference in the calibration of hydrodynamic and morphologic parameters. Calibrating hydrodynamic parameters might improve morphological results, but it is doubtful whether this approach actually improves the physical processes underlying the model. Is a certain set of parameter values best representing reality or only representing the measured data set used for the calibration? An optimal set of hydrodynamic parameters does not necessarily imply optimal morphological results and vice versa!

Model calibration can be a very time consuming, complex process. Many parameters are available for calibration leading to even more possible combinations of these parameters. Furthermore, the processes involved in complex morphodynamic models display both linear and highly nonlinear behaviour. In addition, the large diversity in modelling cases leads to different scopes and goals of these models, all of which require different calibration strategies. Large complex models also require simplifications on the processes and forcings involved, increasing calibration complexity even further.

1.2 OBJECTIVE AND RESEARCH QUESTION

The complexity of a calibration procedure can be decreased by (partly) automating and standardizing the steps involved in the process. A standard framework can be setup for sensitivity analysis, decreasing the amount of parameters involved in the calibration process. An automated approach to calibration can save much time in the setup of numerical models, is objective, and can improve the models results.

The objective of this Masters thesis therefore is:

"Developing an efficient method to improve performance of and insights in the Delft3D model throughout complex morphodynamic applications"

This goal leads to the following research questions:

- *How can an efficient method for calibration and sensitivity analysis be developed?*
- *Can processes be identified that need to be improved in the Delft3D model?*
- *Is there an overlap in the sensitivity of parameters throughout model applications?*
- *Can the uncertainties of model applications be more accurately mapped?*
- *Does calibration of hydrodynamic parameters improve the results of morphodynamic simulations?*

1.3 STUDY APPROACH

Two distinct phases can be distinguished in the development of an automated calibration instrument for morphodynamic Delft3D applications; the development of the instrument (boxes 1, 2 and 3) and the application of the instrument (box 4).

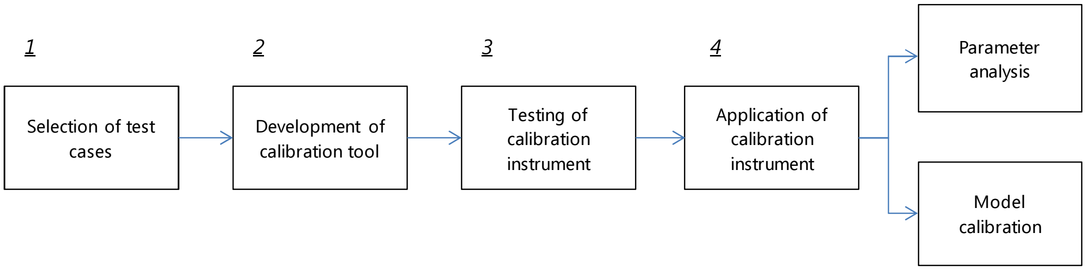

1
2
3
4
Selection of test cases
Development of calibration tool
Testing of calibration instrument
Application of calibration instrument
Parameter analysis
Model calibration

Figure 1-2 | Development and application of instrument

The development of the tool consists out of several steps; the search for test cases, the actual development of the tool and finally, the testing of the instrument. Ideally, the test cases will be increasing in complexity, improving the calibration instrument step by step. The testing of the tool will produce rules of thumb on how to implement the instrument.

Once testing is done, the instrument will be applied to various model applications, both simple and complex. A sensitivity analysis will provide insights in the sensitivity of the model parameters, as well as providing information on the relationships between these parameters. After gathering knowledge on the sensitivity of the parameters, suited parameters will be chosen with which real calibration runs will then be performed.

1.4 LAYOUT REPORT

Chapter 2 starts with a description of the numerical model that has been used during this Masters thesis; Delft3D. The numerical model, its formulations, a selection of its parameters and the processes related to these parameters are discussed. Separately, an overview is provided of the model applications that have been applied during the development and application of the calibration instrument.

In chapter 3, the development and testing of an automated calibration instrument is described, starting with an overview of the OpenDA software package. The software has been adjusted to make it applicable for morphodynamic Delft3D applications. The alterations that have been made to the software are discussed, after which the testing of the upgraded software package, by means of TWIN experiments, is described.

Following the development and testing of the calibration instrument, chapter 4 continues with the description of the application of the software. A sensitivity analysis has been performed on multiple model applications and parameters. Model applications for which measurements were available have been calibrated using the upgraded calibration software.

Chapter 5 wraps up all the lessons learned in the previous four chapters. It tries to summarize and explain the most important conclusions that were drawn after developing, testing and applying the automated calibration instrument.

2 DELFT3D MODEL, PARAMETERS, FORMULATIONS AND MODEL APPLICATIONS

In this chapter, the Delft3D model is described in section 2.1. Section 2.2 continues with a description of the parameters that have been used and the processes they are related to. In section 2.3 an overview is presented of the applied transport formulations. The chapter ends with section 2.4, which describes the different model applications that have been applied during the development and application of an automated calibration instrument.

2.1 DESCRIPTION OF THE NUMERICAL MODEL USED IN THE CALIBRATION PROCESS

Although in theory any numerical model is suitable for calibration, only one numerical model has been used in the development of an automated calibration tool; the software package Delft3D. This chapter starts with a description of the Delft3D model. An overview of the parameters that have been used from this model is provided in section 2.2, including a short description of each parameter and the processes they are related to.

The Delft3D software package has been developed by Deltares and provides a platform for computations of coastal, river and estuarine areas. The package can simulate flows, sediment transports, waves, water quality and morphological developments. It consists out of two different modules; the FLOW and WAVE module. In this thesis, only the FLOW module has been applied and will therefore be further elaborated on. *Delft3D-FLOW is a multi-dimensional (2D or 3D) hydrodynamic (and transport) simulation program which calculates non-steady flow and transport phenomena that result from tidal and meteorological forcing on a rectangular or a curvilinear, boundary fitted grid* (Deltares, 2010, p. 7).

The Delft3D-FLOW module solves the unsteady shallow water equations in either two (depth averaged) or three dimensions. This set of equations is derived from the Navier Stokes equations. Although these equations describe the flow of water very accurately, they are very difficult to solve without simplifications. When these equations are simplified to a set that can be solved using the present day computational resources, a new set of equations is found; the shallow water equations (refer equations 2-1, 2-2 and 2-3). The system of equations that together make up the shallow water equations consists out of the continuity equation and the equations of motion:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \qquad \text{2-1}$$

$$\begin{aligned}\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}&\\ =-g\frac{\partial \zeta}{\partial x}+fv+2\frac{\partial}{\partial x}\left(\nu_t^H\frac{\partial u}{\partial x}\right)&\\ +\frac{\partial}{\partial y}\left(\nu_t^H\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right)+\frac{\partial}{\partial z}\left(\nu_t^V\frac{\partial u}{\partial z}\right)&\end{aligned} \qquad \text{2-2}$$

$$\begin{aligned}\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\\ &= -g\frac{\partial \zeta}{\partial y}-fu+\frac{\partial}{\partial x}\left(\nu_t^H\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right)\\ &+2\frac{\partial}{\partial y}\left(\nu_t^H\frac{\partial v}{\partial y}\right)+\frac{\partial}{\partial z}\left(\nu_t^V\frac{\partial v}{\partial z}\right)\end{aligned} \qquad 2\text{-}3$$

Because of the simplifications that have to be applied to be able to solve the equations that describe the flow of water, parameterizations are introduced to these equations. When calibrating a model, these parameterizations are optimized to fit the problem at hand.

2.2 DESCRIPTION OF FREE MODEL PARAMETERS

An overview of the different parameters that have been utilized throughout this thesis is presented in Table 2-1. Following the Delft3D model layout, the parameters have been categorized in hydrodynamic, morphological and sediment parameters.

Table 2-1 | Overview of free model parameters used in thesis

Module	*Parameter*	*Symbol*	*Description*
Hydrodynamic	Vicoww	ν_{ww}	Vertical Eddy viscosity
	Dicoww	D_{ww}	Vertical Eddy diffusivity
	Vicouv	ν_{uv}	Horizontal Eddy diffusivity
	Dicouv	D_{uv}	Horizontal Eddy diffusivity
	Ccofu	C_u	Uniform value bottom roughness u-direction
	Ccofv	C_v	Uniform value bottom roughness v-direction
Morphological	SusW	Sus_w	Wave-related suspended sed. transport factor
	BedW	Bed_w	Wave-related bed-load sed. transport factor
	RDW	RDW	Wave related roughness height (only used if IopKCW <> 1)
	Rwave	R_{wave}	Wave related roughness = RWAVE * estimated ripple height
	Morfac	[-]	Morphological scale factor
	RDC	RDC	Current related roughness height (only used if IopKCW <> 1)
	AlfaBn	α_{Bn}	Transverse bed gradient factor for bed load transport
	AlfaBs	α_{Bs}	Streamwise bed gradient factor for bed load transport
	Sus	Sus	Multiplication factor for suspended sediment reference conc.
	Bed	Bed	Multiplication factor for bed-load transport vector magnitude
Sediment	SedDia	D_{50}	Median sediment diameter (D50)
	RhoSol	ρ_{sol}	Specific density

Parameters related to the hydrodynamics, the study of liquids in motion, control the hydrodynamic processes; i.e. waves, tides and wind induced currents. Parameters related to the morphodynamics control the processes that are involved in the transport of sediments on the bottom of the wet areas considered.

2.2.1 PARAMETERS RELATED TO HYDRODYNAMIC PROCESSES

Horizontal Eddy viscosity ν_{uv} & diffusivity D_{uv}

For depth averaged, two dimensional simulations, a horizontal Eddy viscosity and diffusivity have to be specified. In a fluid, energy is lost due to friction because of the molecular viscosity of the fluid. The molecular viscosity represents a measure of resistance of the fluid towards deformation by outside forces. In addition, fluid energy is lost due to small scale turbulent vortices, also known as Eddy's. Because of the very small scale of these Eddy's, ranging in the order of μm, they cannot be represented in most numerical grids which range in the order of meters at best. To include this loss of energy due to turbulent vortices on the smallest scales, an additional viscous term in additional to the physical molecular term is introduced; the Eddy viscosity.

Not only does turbulence account for losses of energy, it also induces mixing of the fluid. Mixing of the fluid can take place on large as well as small spatial scales. Because of the limitations on the spacing of the numerical grid, not all spatial scales can be accounted for. To be able to take into account this mixing of the fluid on the smaller scales, the Eddy diffusivity is introduced.

According to the FLOW manual, the values of the horizontal Eddy viscosity and diffusivity depend on the flow and the grid size used in the simulations. For relatively detailed models with grid sizes in the order of tens of meters, typical values are in the range of 1 - 10 $[m^2/s]$, whereas for larger grids of hundred meters of more the values range in between 10 - 100 $[m^2/s]$.

Vertical Eddy viscosity ν_{ww} & diffusivity D_{ww}

In three dimensional simulations, multiple vertical layers are introduced in the model, between which water and momentum are exchanged. Therefore, two additional terms have to be specified; the vertical Eddy viscosity and diffusivity. The vertical Eddy viscosity and diffusivity are computed using a turbulence closure model, which again solve for the lack of resolution on the smallest spatial scales.

Uniform values of the bed roughness Ccofu & Ccofv

In both 2D and 3D calculations, the bed roughness is represented by the Chézy coefficient. This coefficient is used to determine the bed shear stress induced by the flow. The Chézy coefficient is a smoothness coefficient. The higher its value, the smoother the bottom becomes. A decrease of the Chézy smoothness coefficient therefore implies a roughening of the bottom. The manner in which the bed shear stress and the Chézy coefficient are determined varies for both types of calculations. For 2D depth averaged flow, the bed shear stress $\overrightarrow{\tau_b}$ is calculated by a quadratic friction law:

$$\overrightarrow{\tau_b} = \frac{\rho_0 g \vec{U} \left| \vec{U} \right|}{C_{2D}^2} \qquad \text{2-4}$$

Where $\vec{U}$ represents the magnitude of the depth averaged horizontal velocity and C_{2D} the Chézy coefficient. In practice, higher values of the Chézy coefficient lead to higher sediment transport fluxes. Looking at the formula above this might seem strange, as a higher value of c_{2D} would imply a lower value of the shear stress, which in return would imply less transport. The Chézy coefficient however also influences the flow itself via the momentum equation; it is represented in this equation as a friction term. An increase in the value of c_{2d}, implying a smoother bottom, therefore also results in higher flow velocities. The effect of increasing flow velocities combined with an increasing value of c_{2D} can therefore still lead to an increase of the sediment transport.

The bottom roughness C_{2D} can be computed using different formulations. Throughout the three test models, both the Chézy and White-Colebrook formulations have been applied. In the Chézy formulation, the 2D Chézy coefficient is specified by a uniform value in both u- and v-direction:

$$C_{2D} = Ch\acute{e}zy\, coefficient\ \left[m^{1/2}/s\right] \tag{2-5}$$

The White Colebrook formulation requires the specification of a geometrical roughness of Nikuradse k_s [m]. Using this roughness, the Chézy coefficient is calculated as follows:

$$C = 18 \log \frac{12H}{k_s} \tag{2-6}$$

In which H represents the total water depth [m].

In the Egmond and trench migration, multiple layered, 3D model applications, the bed shear stress and Chézy coefficient are calculated differently. The bed shear stress is related to the velocity of the current in the first layer above the bed u_b:

$$\overrightarrow{\tau_{b_{3D}}} = \frac{g\rho_0 \overrightarrow{u_b}|\overrightarrow{u_b}|}{C_{3D}^2} \tag{2-7}$$

Using the distance to the computational grid point closest to the bed Δz_b, the roughness height z_0 of the bed and the magnitude of the bottom stress $|\overrightarrow{\tau_b}| = \rho_0 \overrightarrow{u_*}|\overrightarrow{u_*}|$, the Chezy coeffient is determined as follows:

$$C_{3D} = \frac{\sqrt{g}}{\kappa} \ln\left(1 + \frac{\Delta z_b}{2z_0}\right) \tag{2-8}$$

The roughness height can be determined by the actual geometric roughness of the bottom. It is represented by the RMS value of the sub grid bottom fluctuations. For rough bottoms, the roughness length z_0 can be calculated using the Nikuradse roughness:

$$z_0 = \frac{k_3}{30} \tag{2-9}$$

The Nikuradse roughness k_S typically ranges from 0.15 [m] for rivers beds down to 0.01 [m] for very smooth surfaces.

Wave related roughness height RDW, Current related roughness height RDC and the estimated ripple height R_{wave}

The bed roughness and flow resistance can be specified separately on a sub grid level by the use of trachytopes. At specified time intervals, the roughness will be updated using a prescribed trachytope formulation. These trachytopes, when applied to a model, therefore overrule the standard calculations of Chézy or White Colebrook to determine the Chézy coefficient $C_{2D/3D}$.

The effective sand roughness height k_s simulates the hydraulic roughness of the bottom. This effective bed roughness is not constant, but varies in time and space, depending on the flow conditions. In (Rijn, 2007), four separate contributing components of this effective roughness height are described; the grain roughness $k_{s,grain}$, the wave related bed from roughness $k_{s,w}$, the current related bed form roughness $k_{s,c}$ and k_a, the apparent bed roughness.

In the simulations of the Egmond model, trachytope formulation 105 is applied, which refers to a bed forms quadratic formulation:

$$k_{s,c} = \min\left(\sqrt{k_{s,c,r}^2 + k_{s,c,mr}^2 + k_{s,c,du}^2}, \frac{h}{2}\right) \qquad \text{2-10}$$

In which $k_{s,c}$ represents the effective current related sand roughness height and $k_{s,r}$, $k_{s,mr}$ & $k_{s,d}$ refer to the roughness heights of respectively ripples, mega ripples and dunes.

In Delft3D, two options are available to calculate ks (and kw):

1) k_s is derived from the current related effective roughness height as determined in the FLOW module (spatially varying) and $k_w = R_{wave}\Delta r$ (Δr represents the wave induced ripple height, set to a constant value of 0.025)
2) k_s and k_w are constant in space and specified by: $ks = RDC$ and $kw = RDW$

2.2.2 PARAMETERS RELATED TO MORPHODYNAMIC PROCESSES

Multiplication factors for sediment transport Sus, Bed, Sus_w & Bed_w

The mass balance, which describes the sediment transport rates $[m^3/m/s]$ in relation to bottom changes, reads as follows:

$$(1-p)\frac{\partial z_b}{\partial t} + \frac{\partial S_x}{\partial x} + \frac{\partial S_y}{\partial y} = 0 \qquad \text{2-11}$$

Where p is the porosity, z_b the bed level above a certain horizontal datum and $S_{x/y}$ the sediment transport rates in x and y direction per second and per meter width.

In the sediment transport regime, two distinct types of transport are distinguished; bed load and suspended load. Bed load transport is transport of sediment particles in a thin layer close to the bed, implying that the particles are always close to or in direct contact with the bed. Particles that are in suspension and therefore have no direct contact with the bed are considered to be part of the suspended load.

The computed total sediment transport consists out of four separate contributions: the current related bed load $\overrightarrow{S_{b,c}}$, the wave related bed load $\overrightarrow{S_{b,w}}$, the current related suspended load $\overrightarrow{S_{s,c}}$ and the wave related suspended load $\overrightarrow{S_{s,w}}$. The contribution of these separate contributions to the total sediment transport can be scaled using the multiplication factors Sus, Sus_w, Bed & Bed_w.

Transverse and streamwise bed gradient factor for bed load transport α_{bn} & α_{bs}

The bed load fraction of the total sediment transport is influenced by slopes in the bed; bed level gradients. In a two dimensional plane, two bed slope directions exist; a longitudinal slope in the direction of the flow and a transverse slope in the direction perpendicular to the (initial) flow.

The longitudinal sediment transport rate is influenced by the longitudinal slope and can be calculated using the method of Bagnold:

$$\overrightarrow{S_b'} = \alpha_s \overrightarrow{S_b^n} \qquad \text{2-12}$$

With $\overrightarrow{S_b^n}$ the bed load factor without influence of the bed slope. The coefficient α_s, the longitudinal bed slope correction factor, is calculated as follows:

$$\alpha_s = 1 + \alpha_{bs}\left[\frac{\tan\phi}{\cos\left[\tan^{-1}\left(\frac{\partial z_b}{\partial s}\right)\right]\left[\tan\phi - \frac{\partial z_b}{\partial s}\right]} - 1\right] \qquad \text{2-13}$$

With α_{bs} a tuning factor, ϕ the angle of repose and $\frac{\partial z_b}{\partial s}$ the bed slope in the direction of the unadjusted bed load vector.

Secondly, a bed load vector perpendicular to the longitudinal vector can be calculated:

$$\overrightarrow{S_{B,n}'} = \alpha_{bn}\left|\overrightarrow{S_b'}\right|\frac{u_{b,cr}}{\left|\overrightarrow{u_b}\right|}\frac{\partial z_b}{\partial n} \qquad \text{2-14}$$

Where α_{bn} is a tuning factor, $u_{b,cr}$ the critical near bed flow velocity, $\overrightarrow{u_b}$ the near bed velocity vector and $\frac{\partial z_b}{\partial n}$ the bed slope in the direction perpendicular to the uninfluenced bed load vector. The transverse and streamwise bed gradient factors are thus tuning parameters, with which the relative influence of the bed slopes on the bed load can be determined and adjusted.

Morphological scale factor MorFac

The morphological time scale factor can be used to speed up morphological effects in a model run. The erosion and deposition fluxes that are interchanged between the bed and the flow are simply multiplied with this factor at each numerical time step. Put simple, with a morphological time scaling factor of 10, in a one day run, the morphological changes of 10 days are simulated.

Median sediment diameter D_{50}

The median sediment diameter determines the diameter of the sand grains of the model.

Specific density ρ_{sol}

The specific density of the sediment determines the weight of the particle grains [kg/m^3].

2.3 SEDIMENT TRANSPORT FORMULATIONS

The Delft3D model offers several transport formulations which are used to calculate morphological developments. The transport formulations that have been applied on the model applications throughout the thesis are discussed in this section.

Engelund-Hansen (1967)

The transport formula as described by Engelund and Hansen reads as follows:

$$S = S_b + S_{s,eq} = \frac{0.05\alpha q^5}{\sqrt{g}C^3\Delta^2 D_{50}} \qquad 2\text{-}15$$

Where q represents the magnitude of the flow velocity, Δ the relative density $\frac{(\rho_s-\rho_w)}{\rho_w}$, C the Chézy coefficient and α a tuning parameter. In this formulation, waves have no direct impact on the sediment transport rates. The transport rate is therefore imposed as bed load transport due to currents S_{bc}.

Meyer-Peter-Muller (1948)

The Meyer Peter Muller sediment transport formulation includes a critical shear stress. Initiation of motion of particles will only take place at values of the shear stress above this critical value.

$$S = 8\alpha D_{50}\sqrt{\Delta g D_{50}}(\mu\theta - \xi\theta_{cr})^{3/2} \qquad 2\text{-}16$$

Where α respresents a calibration coefficient, Δ the relative density, μ the ripple factor, θ_{cr} the critical mobility factor determining initiation of motion (0.047), ξ a hiding and exposure factor for the sediment fraction considered and θ the Shields mobility parameter. The Shields mobility parameter and the ripple factor have to be determined separately:

$$\theta = \left(\frac{q}{C}\right)^2 \frac{1}{\Delta D_{50}} \qquad 2\text{-}17$$

$$\mu = \min\left(\left(\frac{C}{C_{g,90}}\right)^{1,5}, 1.0\right) \qquad 2\text{-}18$$

In which $C_{g,90}$ represents the Chézy coefficient related to grains:

$$C_{g,90} = 18\log\left(\frac{12h}{D_{90}}\right) \qquad \text{2-19}$$

In this formulation, waves have no direct impact on the sediment transport rates. The transport rate is therefore imposed as bed load transport due to currents S_{bc}.

Van Rijn 93 / 2000 / 2002 / 2004

The van Rijn (1993) transport formulation is the default transport formula applied by Delft3D when no transport formulation is specified. In (Rijn, van, L.,Walstra, D., 2003) the basic assumption in the transport formulations of van Rijn is formulated; the distinction between bed load and suspended load and the separate contributions to these transport types of currents and waves:

$$S_s = S_{s,c} + S_{s,w} \qquad \text{2-20}$$

$$S_b = S_{b,c} + S_{b,w} \qquad \text{2-21}$$

In which $S_{s,c}$ the suspended transport due to currents, $S_{s,w}$ the suspended transport due to waves and $S_{b,c}$ & $S_{b,w}$ the bed load transport due to respectively currents and waves. To determine the bed level changes due to these transport fluxes, the gradients of these fluxes are determined. Combined with the sediment continuity equation (refer equation 2-11), the following expression is obtained:

$$\frac{\partial z_b}{\partial t} + \frac{\partial\left(S_{b,x} + S_{s,x}\right)}{\partial x} + \frac{\partial\left(S_{b,y} + S_{s,y}\right)}{\partial y} = 0 \qquad \text{2-22}$$

Where the subscripts x and y distinguish between the sediment transport fluxes due to currents and waves in respectively u and v directions.

The current related suspended transport depends on the variation of the suspended sand concentration field which is determined by the currents and waves in the water. In a two dimensional model, the sand concentration field is expressed in terms of the depth averaged equilibrium sand concentration, which is derived from equilibrium transport formulations.

In a three dimensional approach, the transport of suspended sediment is calculated by solving the three dimensional advection diffusion equation:

$$\begin{aligned}\frac{\partial c^{(l)}}{\partial t} + \frac{\partial u c^{(l)}}{\partial x} + \frac{\partial v c^{(l)}}{\partial y} + \frac{\partial\left(w - w_s^{(l)}\right)c^{(l)}}{\partial z} - \frac{\partial}{\partial x}\left(\epsilon_{s,x}^{(l)}\frac{\partial c^{(l)}}{\partial x}\right)\\ - \frac{\partial}{\partial y}\left(\epsilon_{s,y}^{(l)}\frac{\partial c^{(l)}}{\partial y}\right) - \frac{\partial}{\partial z}\left(\epsilon_{s,z}^{(l)}\frac{\partial c^{(l)}}{\partial z}\right) = 0\end{aligned} \qquad \text{2-23}$$

With $c^{(l)}$ the mass concentration of sediment fraction (l) $[kg/m^3]$, u, v & w the flow velocity components $[m/s]$, $\epsilon_{s,xyz}^{(l)}$ the Eddy diffusivities of the sediment fraction and $w_s^{(l)}$ the settling velocity of the sediment fraction. Put simple, this equation represents a balance between sand particles falling down because of gravity and sand particles being stirred up again because of the turbulent Eddy's in the water column.

The settling velocity of a particle represents the velocity at which the drag forces acting on a grain of sand are in equilibrium with the force of gravity counteracting the drag. The settling velocity increases with increasing grain diameter and is calculated as follows:

$$w_{s,0}^{(l)} = \frac{10v}{d}\left[\left(1+\frac{0.01\left(s^{(l)}-1\right)gd^{(l)3}}{v^{2}}\right)^{0.5}-1\right] \tag{2-24}$$

The current related suspended transport rates in x- and y-direction are calculated based on the computed sand concentration fields:

$$S_{s,c,x} = \int_{a}^{h}\left(uc-\epsilon_{s,x}\frac{\partial c}{\partial x}\right)dz \tag{2-25}$$

$$S_{s,c,y} = \int_{a}^{h}\left(vc-\epsilon_{s,y}\frac{\partial c}{\partial y}\right)dz \tag{2-26}$$

Where a, as described in the FLOW manual (Deltares, 2010, pp. 354 - 356), represents a reference height, used to distinguish between the two main types of transport; sediment transport below this threshold value is considered bed load, whereas sediment transport above this value is treated as suspended load. The reference height a is determined as follows:

$$a = \min\left[\max\left\{AKSFAC\cdot k_{s},\frac{\Delta r}{2},0.01h\right\},0.20h\right] \tag{2-27}$$

Where a is the van Rijn reference height, $AKSFAC$ a user defined proportionality factor, k_s a user defined current related effective roughness height, Δr the wave induced ripple height, set to a constant value of 0.025 and h the water depth.

Sediment is entrained in the water column because of a reference concentration which is imposed at the reference height:

$$C_{a}^{(l)} = SUS\eta^{(l)}0.015\rho_{s}\frac{d_{50}^{(l)}\left(T_{a}^{(l)}\right)^{1.5}}{a\left(D_{*}^{(l)}\right)^{0.3}} \tag{2-28}$$

Where $c_a^{(l)}$ represents the mass concentration at the reference height a, and SUS a scaling factor. The reference concentration embodies the lower boundary condition for the calculation of the suspended sediment concentration of equation 2-23. The upper boundary is made up out of the water surface, at which the sediment concentration is zero. The following five formulas are only presented to show the influence of k_c & k_w on the reference concentration. It is beyond the scope of this thesis to elaborate further on the other terms in the equations.

$$T_a^{(l)} = \frac{\left(\mu_c^{(l)}\tau_{b,cw} + \mu_w^{(l)}\tau_{b,w}\right) - \tau_{cr}^{(l)}}{\tau_{cr}^{(l)}} \tag{2-29}$$

$$\mu_c^{(l)} = \frac{f_c'^{(l)}}{f_c^{(l)}} \tag{2-30}$$

$$f_c^{(l)} = 0.24\left[\log\left(\frac{12h}{k_c}\right)\right]^{-2} \tag{2-31}$$

$$\tau_{b,w} = 0.25\rho_w f_w \left(\widehat{U}_\delta\right)^2 \tag{2-32}$$

$$f_w = \exp\left[-6 + 5.2\left(\frac{\widehat{A}_\delta}{k_w}\right)^{-0.19}\right] \tag{2-33}$$

The magnitude of the bed load transport which takes place below the reference height a is calculated as follows:

$$|S_b| = 0.006\eta\rho_s w_s d_{50}^{(l)} M^{(0.5)} M_e^{(0.7)} \tag{2-34}$$

Bijker (1971)

The Bijker transport formula calculates bed load and suspended load separately. Furthermore, separate contributions to the bed and suspended load transport due to wave asymmetry and the bed slope are incorporated in the formulation for the bed load:

$$\overrightarrow{S_b} = \overrightarrow{S_{b0}} + \overrightarrow{S_{b,asymm}} + \overrightarrow{S_{s,asymm}} + \overrightarrow{S_{b,slope}} + \overrightarrow{S_{s,slope}} \tag{2-35}$$

$$\overrightarrow{S_0} = \overrightarrow{S_{s0}} \tag{2-36}$$

2.4 OVERVIEW OF UTILIZED MODEL APPLICATIONS

During the development of the automated calibration tool, three Delft3D model applications have been utilized; the Egmond, trench migration and basin model. After testing of the workings of the tool, it has also been applied to a real life test case; the Sand engine model. The utilized model applications will be discussed next.

Description of the Egmond model

The Egmond model is representative for the location of Egmond aan Zee, which is located at the Dutch coast in between Den Helder in the north and Ijmuiden in the south. The model is a Cross shore profile 2DV model with 12 layers, covering about 1500 meters of cross shore distance up to a depth of -15 [m]. The grid is made up of 1 cell in the N- and 173 cells in the M-direction. In the area of interest, the grid size ranges in between 40 [m] offshore to 20 [m] nearshore.

The initial bathymetry of the model, showing two sub tidal breaker bars and one intertidal swash bar, is presented in Figure 2-1. The figure also shows the areas of interest that have been investigated throughout the different calibration and test runs with this model. These areas are in accordance with the areas used in (Giardino, A., Briére, C. Werf van der, J., 2011).

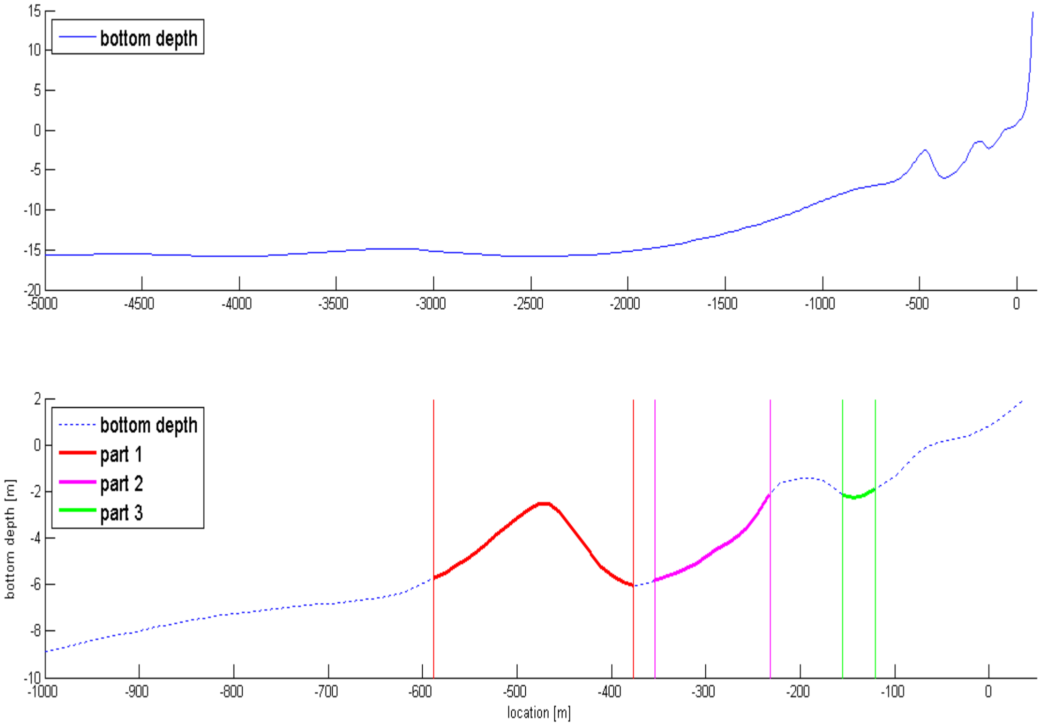


Figure 2-1 | Initial bottom Egmond model including areas of interest

At the offshore boundary, the model is forced by a water level type of boundary condition. This single semi diurnal tidal forcing represents the average long term situation; a so called morphological tide. At the northern and southern open boundaries, a Neumann type of boundary condition is applied. Apart from these tides, the model is also forced by waves.

Measured bathymetrical data of the Egmond model site is available of the following data; 16-10, 19-10, 04-11 and 07-11. The model is therefore run for a period of 16-10 to 07-11. Because of the availability of this measured bottom changes, this model is very well suited for calibration purposes.

Description trench migration model

The trench migration model represents a section of a river of about 30 meters long and 0.5 meters wide. Although the model is only 0.5 meters wide, it contains multiple cells in both M- (5) & N-directions (100) with 10 layers, which implies that this is a full 3D model.

The initial bathymetry of the model is presented in Figure 2-2. The figure shows the initial trench of about 20 [cm] of depth that is located about halfway in the model.

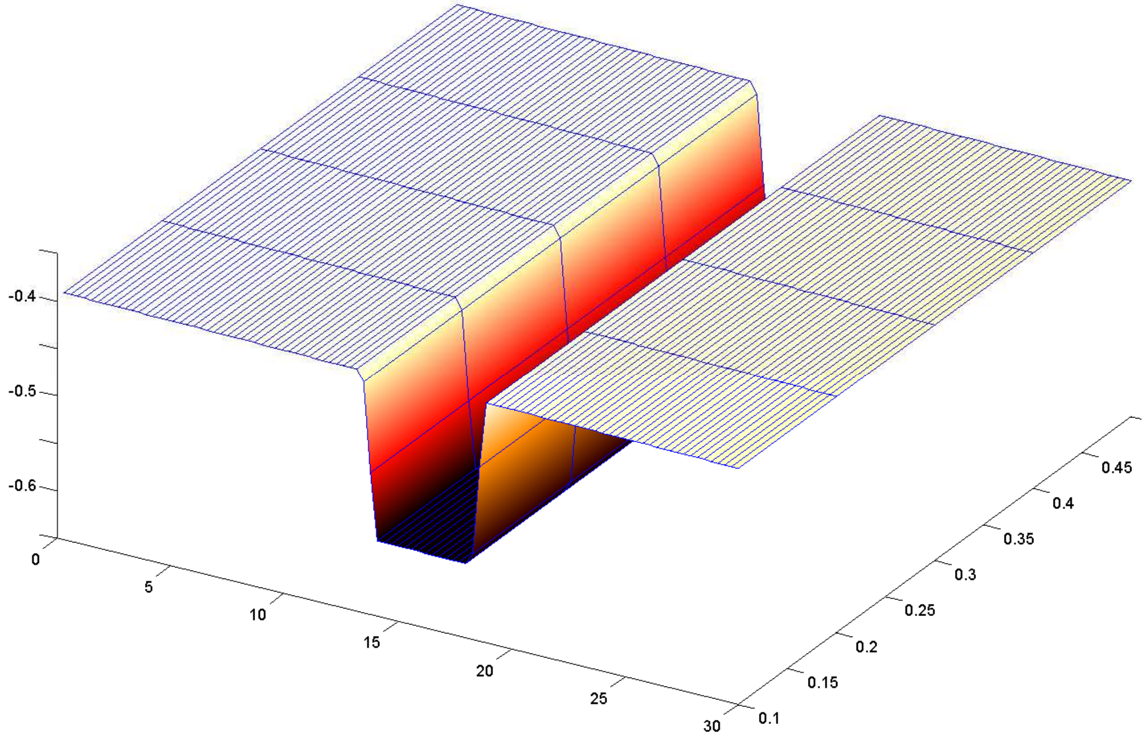


Figure 2-2 | Initial bathymetry trench migration model

At the upstream boundary of the model, a water level is prescribed, whilst at the downstream boundary, a constant current of 0.51 [m/s] is applied. Measured data of the migrating trench is available at 0 [h] and after 15 [h]. The model is therefore run for a period of 15 hours.

Description of the basin model

The basin model is a depth averaged 2DH model containing 149 cells in both M- and N-directions and only one layer of depth. This model has been created for theoretical purposes only, implying that the model does not represent a specific part of the world like the Egmond model. Therefore no measurements of the modelled area are present.

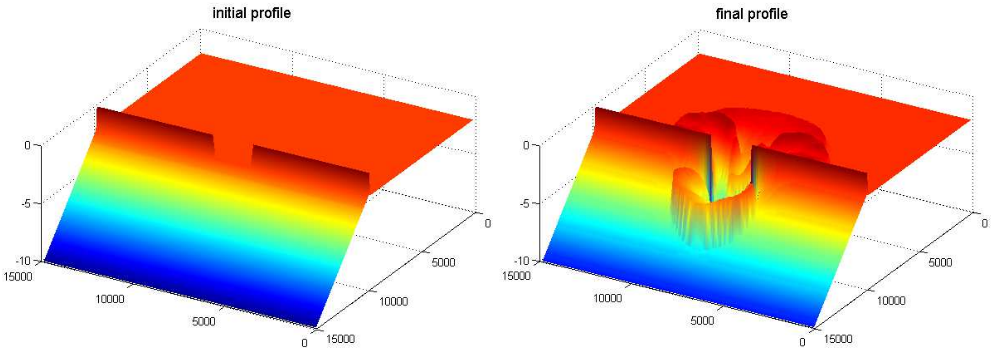


Figure 2-3 | Initial and final profile basin model

The seaward deeper end of the model located in the south in Figure 2-3, is specified by an open boundary, on which a harmonic tide is forced. The figures provide an overview of both the initial and final profile of the basin model. The northern, eastern and western boundaries are all closed. The shallow red part can thus be seen as a basin, which is filled and emptied by the harmonic tide through the opening located in the middle of the model.

Description of the Sand engine model

The Sand engine model is a 2DH, depth averaged, single layered model. The model is a numerical replica of the Sand engine, which is an unique large scale nourishment located at the Dutch coast in front of Monster above the Hoek van Holland, just south of Scheveningen beach. The Sand engine model is forced by both tides and waves, coupling runs from the Delft3D FLOW and WAVE modules.

The measured bottom profile on 03-08-2011 is presented in Figure 2-4. The number of cells in respectively M and N direction are shown on the eastern and southern axis.

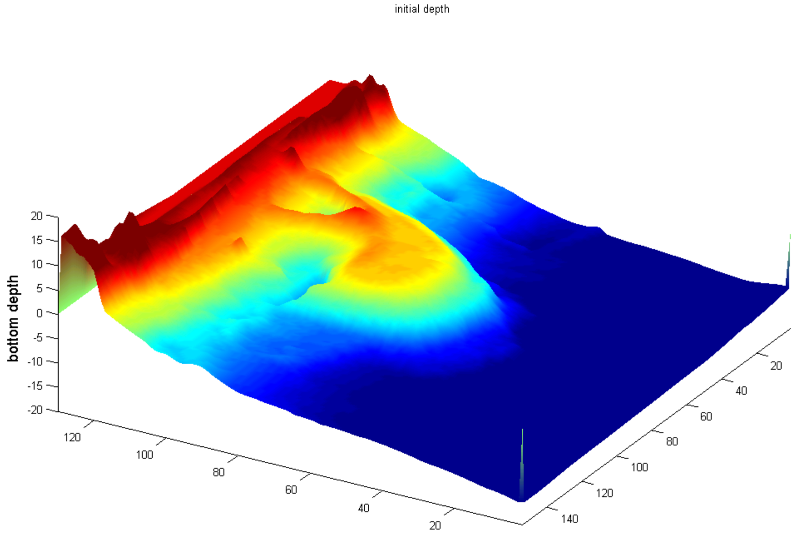


Figure 2-4 | Bottom profile Sand engine on 03-08-2011

The offshore located, western boundary of the model is forced by a harmonic tide. This western boundary is cut into pieces, all of which are forced by a slightly different tidal signal, to be able to reproduce a very realistic tidal forcing. On the northern and southern boundaries, Neumann, or water level gradient boundaries are prescribed.

The stretch of coast containing the Sand engine is monitored every month, to keep a detailed track of the morphological developments of this large scale nourishment. This abundance of detailed bottom data makes this model perfect for calibration purposes.

3 DEVELOPMENT AND TESTING OF AN AUTOMATED CALIBRATION TOOL

A software package called OpenDA has been developed for calibration and data assimilation purposes by Deltares. Section 3.1 starts with an introduction of the existing software package. In section 3.2, an overview is provided of the alterations that have been made to the software to make it suitable for the objectives of this thesis.

3.1 WORKINGS OF THE OPENDA (OPEN DATA ASSIMILATION) SOFTWARE PACKAGE

3.1.1 GENERAL OVERVIEW MAIN COMPONENTS OPENDA

An automated calibration tool named OpenDA has been developed at Deltares. This open data assimilation software package is an open source, generic toolbox for calibration and real time data assimilation. The software has been set up in a modular fashion, making it applicable to virtually any numerical model. The calibration functionality of the software package will be described next, as the data assimilation extension is out of scope of this thesis.

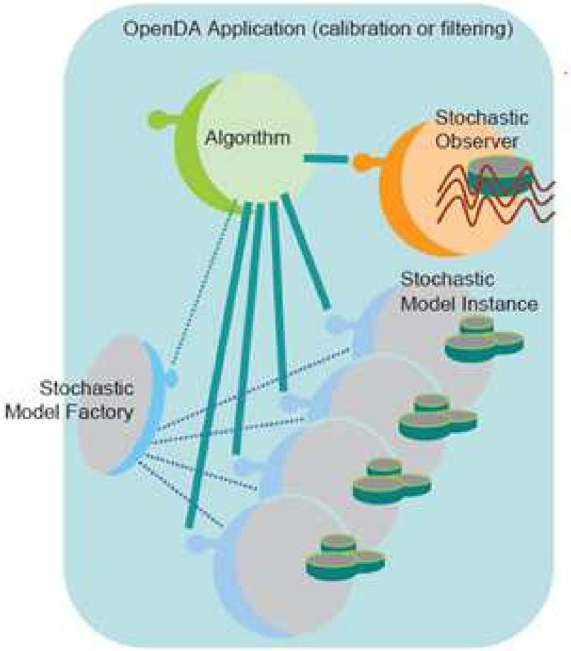

OpenDA Application (calibration or filtering)
Algorithm
Stochastic Observer
Stochastic Model Instance
Stochastic Model Factory

Figure 3-1 | overview general workings OpenDA toolbox (Hummel, S., Verlaan, M., Sarafy, el, G., Velzen, van, N., 2011)

The package consists out of several interacting building blocks which are illustrated in Figure 3-1. These blocks offer a platform on which algorithms and models can interchange information. The three main building blocks are the algorithm, the stochastic observer and the stochastic model factory. The algorithm building block is the main component of the OpenDA software package. In the algorithm block, the algorithm used in the calibration process is specified. It uses information from the stochastic observer and the different stochastic model instances and compares the values of both during the calibration process. More information on the workings of the different algorithms is provided later. The stochastic model factory creates the different stochastic model instances that are used by the algorithm; each model instance corresponds to one model run. In the stochastic observer all information is found with regards to the measurements against which the models predictions are compared.

OpenDA defines standardized interfaces which are used in the communication between the three main building blocks. The interfaces that are implemented by the model building block are subdivided in the model instance interface and the stochastic model interface. "*The model instance interface defines functionalities that a model should implement. The stochastic model instance interface defines the stochastic extension of the deterministic model.*" (OpenDA). When implementing OpenDA to a certain numerical model, the different interfaces have to be coupled to the model; this is also known as wrapping of the model. In case of Delft3D, this has been done by means of a so called black box configuration. This implies that the user only has to develop modules, java classes, which can read and write in- and output files of the model. The black box model utilities then create a stochastic model extension which is used in the calibration process.

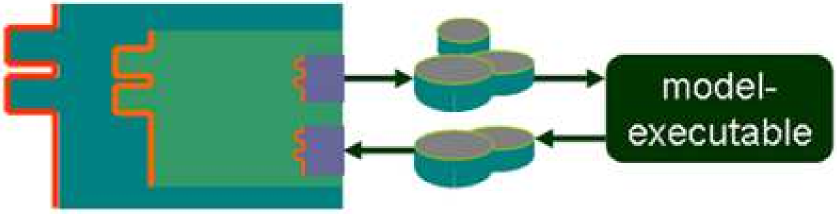


Figure 3-2 | workings black box; outer most box (dark green): black box ***stochastic model utilities****; inner box (light green): black box* ***model utilitie****s; inner most boxes (purple): black box* ***data objects*** *(OpenDA)*

Black box data objects convert the model instance's in- and output files into exchange items. These exchange items are the items that can be changed and exchanged during the calibration process by the algorithm and the numerical model. The black box model utilities task is the selection of exchange items that are to be used in the calibration process from the model that is being calibrated. Both the parameters that will be calibrated as well as the results on which the calibration is dependent are chosen from the Delft3D model. The black box stochmodel instance groups all the exchange items into vectors. Three distinct vectors can be identified; a parameter vector, a state vector and a predictions vector. Figure 3-3 provides an overview of the black box components.

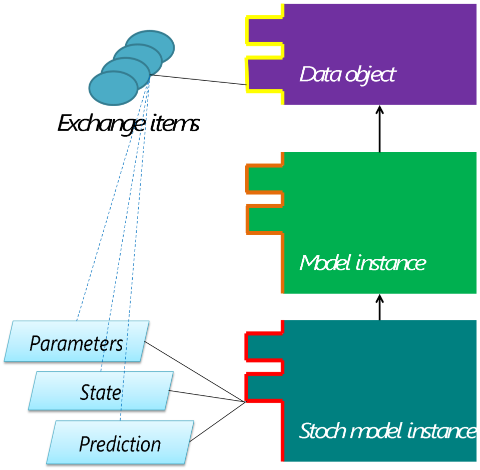


Figure 3-3 | black box components and their roles

The parameter vector contains all the information related to the parameters that are being calibrated. The information of the parameters is fed into the model, which is then run using the parameters settings from the parameters vector. The outcome generated by the model in return is stored in the predictions vector. The models predictions are than compared to the measurements. If necessary, the parameters vector is adjusted and the model is re-run, until one of the stop criteria of the search algorithm, which is used in the optimization process, is met. The state vector contains the initial state of the model. This state can be used to improve the models predictions. When a model is run continuously, like whether models, the model state can be updated using measurements, thereby improving the models results. This so called data assimilation function of OpenDA is out of scope of this thesis.

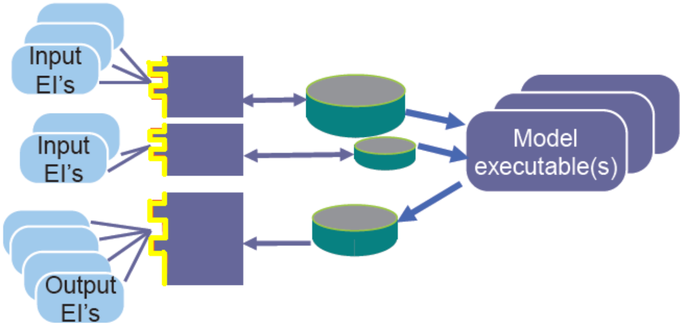


Figure 3-4 | role of data objects in the exchange of information between OpenDA & a numerical model

(Hummel, S., Verlaan, M., Sarafy, el, G., Velzen, van, N., 2011)

Figure 3-4 repeats the role of the data objects in the calibration process. The data object receives input exchange items, containing information about the values of the parameters that are being calibrated. These are fed into the numerical model, which performs a run with these parameter settings. The models results are than transformed into output exchange items by the data object, which in return is stored in the predictions vector. These results are compared to the measurements. This process is repeated until one of the stop criteria of the optimization algorithm is met.

3.1.2 OVERVIEW OF WORKINGS OPTIMIZATION ALGORITHM

A calibration procedure starts with the selection of the model parameters that have to be calibrated. Secondly, an optimization algorithm is chosen, which assigns values to these parameters. Depending on the calibration algorithm, these values are improved following specific rules.

For calibration, the following list of algorithms is present in the OpenDA toolbox (Verlaan): Dud, Sparse Dud, Simplex, Powell, Gridded full search, GLUE, Conjugate gradient, LBFGS. Although many optimization methods are available, only Dud (Does not Use Derivatives) has been applied throughout this thesis. Both (Giardino, A., Briére, C. Werf van der, J., 2011) and (Gautier, 2010) recommend the use of Dud. The method has proven to be highly efficient and produce robust results.

The selected algorithm optimizes a goodness of fit criterion (GoF) in the process of improving the parameter values. This criterion is a direct measure of the calibration quality of each iteration in the calibration process. The GoF criterion can be expressed as a cost function, which expresses the accuracy of every iteration in terms of a certain cost.

The cost function that has been used in this thesis is written in the form of a least squares function, which determines the aggregated difference between the model runs and the measurements provided (refer equation 3-1). The parameter values that are found at the minimum value of the cost function represent the optimal parameter settings.

$$J(\vec{\Theta}) = \sum_{n=1}^{N} \frac{\left(\xi_n(\vec{\Theta}) - \widehat{\xi_n}\right)^2}{\sigma^2} \qquad 3\text{-}1$$

$\Theta_{1,P}$ = estimates of the model parameters
$\vec{\Theta}$ = $(\Theta_1, \Theta_2, \dots, \Theta_P)$
$\xi_n(\vec{\Theta})$ = model predictions
$\widehat{\xi_n}$ = measurements
σ^2 = uncertainty of measurements
N = active grid cells

The uncertainties of the observations σ^2 are included in the cost function. If the uncertainty of a certain set of measurements is relatively high, a relatively small weight will be assigned to these measurements in the cost function, as the difference between model results and measurements is divided by the uncertainty. Therefore, the influence of these measurements on the cost functions value will be smaller than that of measurements that are less uncertain.

Doesn't use derivatives (Dud) optimization algorithm

Doesn't Use Derivatives (DUD) is a derivative free minimization algorithm for nonlinear least squares problems (Ralston, M.L., Jennrich, R.I., 1978). The search direction is determined by the method on the basis of a repeated linearization of the models predictions $\xi_n(\vec{\Theta})$.

For two parameters, α and β, the Dud minimization algorithm operates as follows:

- Three model runs are performed with parameter settings $[\alpha_0, \beta_0]$, $[\alpha_0 + \Delta\alpha, \beta_0]$ and $[\alpha_0, \beta_0 + \Delta\beta]$.
- Secondly, the model is linearized around these values, resulting in a search direction and parameter values $[\alpha_1, \beta_1]$to be tested.
- If this is an improvement, the linearization is updated with the new point (if the problem is linear, the algorithm is done after this step).
- The procedure is repeated until a minimum is found. The first iteration step might require more model evaluations to find a better estimate when multiple parameters are specified.
- If no minimum is found, a line-search is performed in the direction with the lowest values of the cost function until an improvement is found.

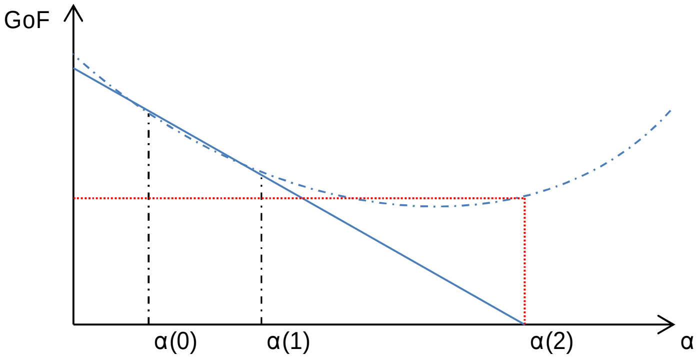


Figure 3-5 | Visualization working Dud algorithm for one parameter

Figure 3-5 visualizes the working of Dud on a linear problem where only one parameter (α) has to be calibrated. In the first step, two runs are performed with parameter settings $[\alpha_0]$ and $[\alpha_0 + \Delta\alpha = \alpha_1]$. Using linearization along the solid blue line, the first guess to solve the problem and minimize the cost function is a run using parameter setting $[\alpha_2]$. Because of the linearity of the problem, the algorithm detected the minimum after one guess, in the third run. The dashed blue line represents a nonlinear problem. Although the first guess of the algorithm will remain the same, parameter setting $[\alpha_2]$ does not represent the minimum in this nonlinear case as indicated by the dashed red line.

There are various options available to configure the Dud algorithm. *"In general, however, the user should not adjust the internal control settings of the algorithms"* (Wenneker, I.,Gautier, C.,Gerritsen, H., 2009, p. 30). Although it is recommended not to alter the internal configuration of the Dud algorithm, discussing the various options to do so provides more insights in the workings of the algorithm. The algorithm can be configured using the following lines:

- outerLoop maxIterations="9"
- absTolerance="0.01"
- relTolerance="0.01"
- innerLoop maxIterations="6"
- maxRelStepSize="10.0"
- backTracking shorteningFactor="0.5"
- startIterationNegativeLook="3"

The absolute and relative tolerance criteria determine when the algorithm is done with its minimization process (refer equations 3-2 and 3-3). When either of these criteria is met, the algorithm stops.

$$absTolerance = GoF_t - GoF_{(t-1)} \tag{3-2}$$

$$relTolerance = \frac{GoF_t - GoF_{(t-1)}}{GoF_{(t-1)}} \tag{3-3}$$

If the initial guess is not correct because of the non-linearity of a certain problem and the algorithm overshoots, it will start backtracking with the given shortening factor, decreasing the step size with which the parameter values are altered along the way. If this still does not improve the cost function, the method will also start looking in a negative direction after a certain amount of runs (startiterationnegativelook). If no progress is found after *'LineSearch maxIterations'*, the program will stop.

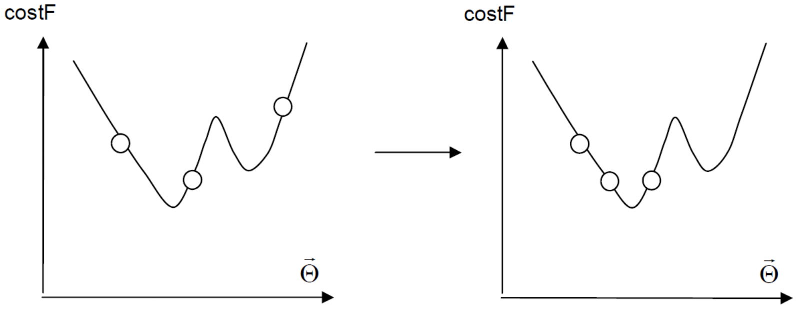


Figure 3-6 | Evaluation of the minima with inner and outer loop (Giardino, A., Briére, C. Werf van der, J., 2011)

Figure 3-6 shows the difference between the inner- and the outerloop. In the innerloop, Dud refines the values of the parameters locally until one of the stop criteria is met. The outerloop is used to 'jump' out of possible local minima.

Gridded full search algorithm

The second algorithm that has been applied is the gridded full search algorithm. This algorithm can be used to systematically alter parameter values with which the numerical model is then run. This algorithm is very well suited to perform a sensitivity analysis on a pair of parameters.

3.1.3 CONFIGURING THE OPENDA COMPONENTS

The communication interfaces used by OpenDA are configured by means of XML files. In these files, all the required information on the building blocks is provided. There are four configuration files that have to be set before any calibration: the main configuration file (1), the stochastic model file (2), the stochastic observer file (3) and the algorithm file (4).

1) Main configuration file (filename.oda); OpenDA java class names, working directories and configuration file names are specified for the three main building blocks.

Furthermore, it is specified how the results of the calibration are outputted; either in the form of an m-file (Matlab file), as comma separated values or both.

2) Stochastic model files; all information with regards to the Delft3D model is provided. Delft3D is implemented as a black box model, which requires three configuration files: a wrapper file, a model file and a stochmodel file.
    a. Wrapper file (D3DWrapper.xml); specifies which java classes have to be used to read and write the in- and output files of the model. OpenDA uses these java classes to communicate with the model in the black box configuration. Also, generic information about the model and execution steps of the models executable are provided.
    b. Model file (D3DModel.xml); provides more specific information about the model, such as which parameters and what information from the models runs results has to be used in the calibration.
    c. Stochastic file (D3DStochModel.xml); describes the stochastic model configuration. This file contains information on the vector specification implying that the initial values and step sizes of the calibration parameters are specified. Also, the models results are linked to the measurements.
3) Stochastic observer files; provide all information, including uncertainties, that is related to the measurements.
    a. Stochastic file (stochObsConfig.xml); specifies which measurements are to be used in the calibration process.
    b. Uncertainties file (obsUncertainties.xml); specifies the uncertainties corresponding to each of the measurements used.
4) Algorithm file; all input for the selected algorithm is specified.

3.2 ALTERATIONS MADE TO THE OPENDA SOFTWARE PACKAGE

Although OpenDA suits the specifications of an automated calibration tool very well, it does not support the calibration of morphological parameters of Delft3D using morphological measurements and model results. In the past, Delft3D model applications have been calibrated using OpenDA, but only using hydrodynamic results to calibrate hydrodynamic parameters.

3.2.1 DEVELOPMENT AND APPLICATION OF ADDITIONAL JAVA READERS

As stated earlier, OpenDA has to be coupled to the numerical model that needs to be calibrated. In case of Delft3D, a black box configuration has been implemented to wrap the model. To prepare OpenDA for calibration of morphodynamic parameters, java code had to be written which extracts the required information from the Delft3D model applications.

Java code has been written which allows the required morphological parameters to be adjusted during the calibration process. Readers have been developed that allow calibration on all parameters from the sediment file, the morphological file, as well as all numerical parameters from the model definition file. In addition to the extraction of the desired parameters, morphological results had to be obtained from the Delft3D model applications. The required bottom depth from the model is stored in a so called trim file. A reader has been developed which enables the extraction of the bottom depth. The bottom depth can be extracted at one or more time steps, as well as for any single, or multiple parts of the grid.

Lastly, a reader has been developed which reads data from depth files (.dep files). Files of this format are used to implement the bathymetry of any Delft3D model application. Using either the depth file, or the trim file reader, morphological measurements can be loaded into the OpenDA application.

The bed level results from the numerical model have been implemented in the least squares cost function format (refer 3-1). The input files allow for a choice of both active time steps and grid parts to be calibrated on. The cost function sums the separate values of these active grid parts for every time step.

$$J(\vec{\Theta}) = \sum_{t=1}^{T}\sum_{n=1}^{N} \frac{\left(\xi_n(\vec{\Theta}) - \widehat{\xi_n}\right)^2}{\sigma^2} \qquad 3\text{-}4$$

$\Theta_{1,P}$	=	estimates of the model parameters
$\vec{\Theta}$	=	$(\Theta_1, \Theta_2, \ldots, \Theta_P)$
$\xi_n(\vec{\Theta})$	=	predicted bottom depth
$\widehat{\xi_n}$	=	measured bottom depth
σ^2	=	uncertainty of separate measurements
T	=	time steps selected
N	=	active grid cells (depending on grid part selection)

3.2.2 IMPLEMENTATION OF THE DECOMPOSED BRIER SKILL SCORE

Numerical models have to be judged on their performance. The performance of a numerical model is an indication of how well reality is reproduced by the model. In case of a calibration procedure, the numerical model has to reproduce a certain set of measurements, representing reality. This performance can be assessed by three parameters; the bias, accuracy and skill (Sutherland, J., Peet, A., Soulsby, R., 2004). The difference between the central tendencies, the mean of the model, and the predictions, is described by the bias. The accuracy is a measure of the average size of the difference between the models results and the corresponding measurements. Skill finally is based on the accuracy. Skill is a measure of the accuracy of a model run relative to the accuracy of a baseline prediction.

The implementation of the decomposed Brier skill score (BSS) is a means to evaluate the results of each calibration iteration. The BSS is a non-dimensional value based on a least squares principle as is the least squares cost function.

$$BSS = 1 - \frac{MSE(Y,X)}{MSE(B,X)} = 1 - \frac{\langle (Y-X)^2 \rangle}{\langle (B-X)^2 \rangle} \qquad 3\text{-}5$$

Where Y are the models predictions, X the measurements, B the baseline prediction and MSE the abbreviation of the mean squared error. The baseline prediction is represented by the initial bathymetry of the model. The BSS can take values in between 1 and $-\infty$, with 1 corresponding to perfect modeling and negative scores implying that the model performed worse than the no change baseline scenario.

Implementing the BSS as a cost function will not produce different calibration results as it is based on a least squares principle as is the least squares cost function. Whether the algorithm is minimizing the least squares cost function or maximizing the BSS does not make a difference in a mathematical sense. The decomposed scores however do provide insight in why the cost function is decreasing or increasing in value. If a skill score is less than 1, it is not clear what the reason for the imperfect score is from the score itself. Take a modeled breaker bar as an example; the shape of the bar could be to flat and simultaneously, it could have been moved in the wrong direction, both factors contributing to the error. From the BSS alone it is not clear which factor is contributing more to the error. Equation 3-6 shows the BSS, decomposed into four separate parameters α, β, γ & ϵ, as proposed in (Murphy, A., Epstein, E., 1988).

$$BSS = \frac{\alpha - \beta - \gamma + \epsilon}{1 + \epsilon} \qquad 3\text{-}6$$

The values of α, β, γ & δ provide insights in the type of error that is found in the predictions of the model; a phase, amplitude or map mean error. The parameter α is a measure of the phase error found in the predictions. The occurrence of a phase error implies that the sand is moved to the wrong position. If the shape of a certain bottom profile is predicted perfectly, but it is shifted somewhat in the wrong direction, this is reflected in the phase error. It ranges in between 0 to 1, where 1 represents perfect modeling of the phase.

$$\alpha = r^2_{Y'X'} \qquad 3\text{-}7$$

$$r^2_{Y'X'} = (r_{Y'X'})^2 = \left(\frac{\langle Y'X' \rangle}{\sigma_{Y'}\sigma_{X'}}\right)^2 \qquad 3\text{-}8$$

$$Y' = Y - B \qquad 3\text{-}9$$

$$X' = X - B \qquad 3\text{-}10$$

The parameter β is a measure of the amplitude error. The amplitude error expresses differences in the volumes of sand that are moved. It ranges in between 0 to ∞, where 0 represents perfect modeling of phase and amplitude.

$$\beta = \left(r_{Y'X'} - \frac{\sigma_{Y'}}{\sigma_{X'}}\right)^2 \qquad 3\text{-}11$$

The map mean error, or the average error, is expressed by γ. It expresses the deviation of the predicted average bed level from the average of the measurements. The parameter ranges in between 0 and ∞, with 0 representing perfect modeling of the map mean.

$$\gamma = \left(\frac{\langle Y' \rangle - \langle X' \rangle}{\sigma_{X'}}\right)^2 \qquad 3\text{-}12$$

The final parameter ϵ is a normalization term. It is described as the mean difference between measurements and the baseline prediction divided by the standard deviation of these differences. Therefore, it is the only term that is unaffected by the model predictions.

$$\epsilon = \left(\frac{\langle X' \rangle}{\sigma_{X'}}\right)^2 \qquad \text{3-13}$$

Testing of decomposed Brier skill score

Several tests have been performed to check the suitability of the decomposed Brier skill score in determining whether a certain simulated bottom profile matches another measured or simulated bottom profile. First, two test cases are discussed, after which a real test is performed on the basin model. The first test case simulates a simple 2D bottom profile and the second test case simulates a simple 3D bottom profile. The first two test cases do not represent real model results, but have been constructed using Matlab.

Figure 3-7 and Figure 3-8 display the results of the 2D test cases. The first graph in these figures shows the start position of a sand bar, represented by the dashed black line. Its position is plotted on the x-axis against the bottom height on the y-axis. The dashed red line shows the measured position of the same sand bar after a certain amount of time. The five colored bars plotted around the start and measured sand bar positions represent five different hypothetical simulation results. In the first test, the shape of the sand bars has been kept constant, but the position of the simulated sand bars differs throughout the different runs. During the second test, both the shape and the position of the simulated sand bars vary.

The second graph of the figure shows the decomposed values of the Brier skill score, corresponding to the five different hypothetical simulation results. The values of the Brier skill score, alpha, beta and epsilon are represented by colored lines. The values indicated by the lines correspond to values on the left axis. The values of gamma are printed in the graph in a textbox. The values of the dashed lines in the graph correspond to the values on the right axis. This axis represents both the normalized distance between the top of the simulated sand bars and the top of the measured sand bar, as well as the normalized size of the bars. The distance is normalized by the distance between the start position and the measured position of the sand bar. The size of the sand bars is normalized by the volume of the measured sand bar. Both distance and size of the sand bars are normalized to be able to compare the results between the 2D and 3D tests. The results of the two 2D tests are discussed next; the separate scores that together make up the decomposed Brier skill score are discussed one by one.

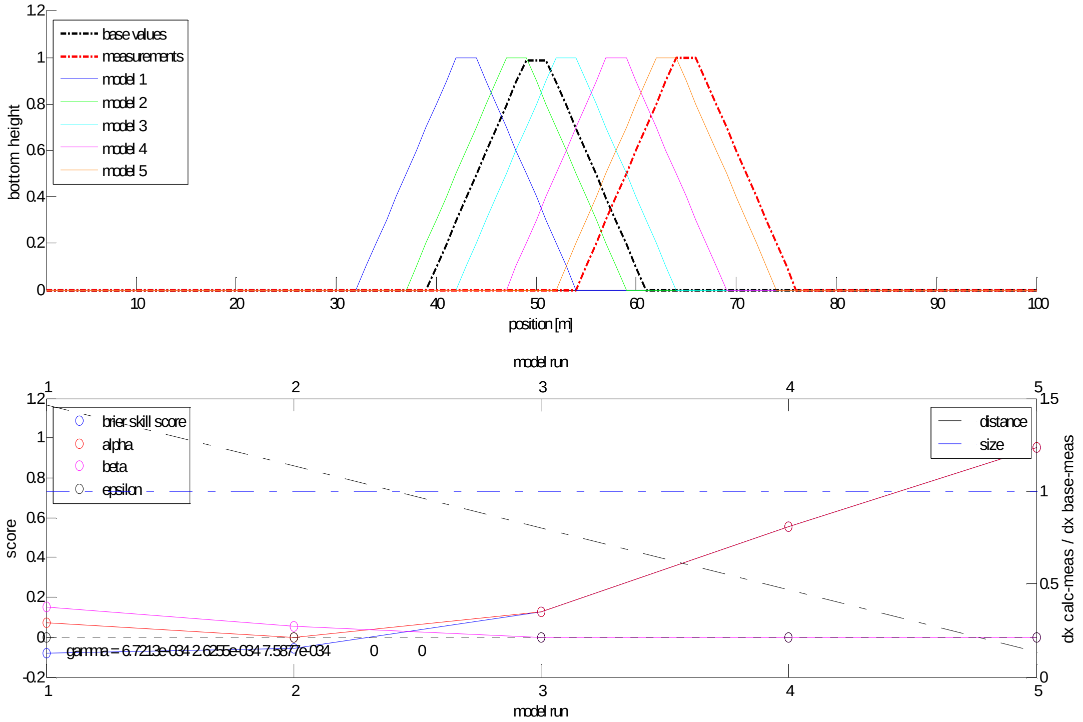


Figure 3-7 | 2D bottom profile test 1 decomposed Brier skill score values

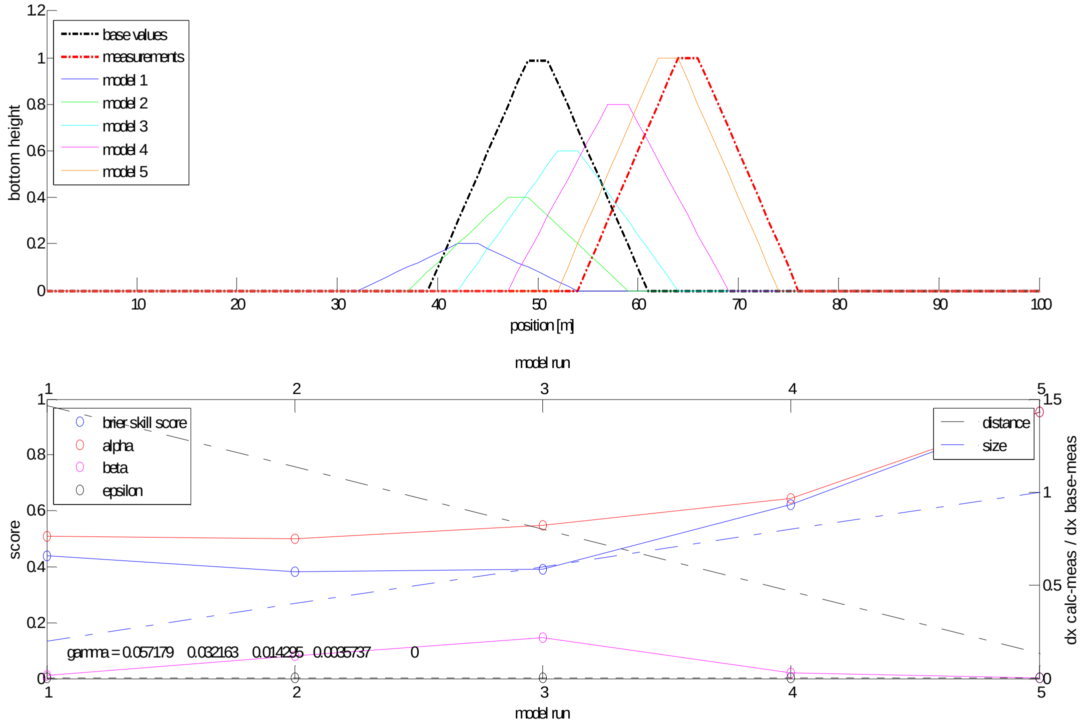


Figure 3-8 | 2D bottom profile test 2 decomposed Brier skill score values

In the first test, the BSS for the blue and green bars are negative. This is correct, as these bars have been shifted to the left side of the start position of the sand bar, whereas the measured sand bar moved to the right. The results of these simulations therefore are worse than the no change scenario with regards to the start position of the sand bar, indicated by a negative Brier skill score. The BSS for run 3, 4 and 5 are positive and increase as the bars move closer towards the measured bar as expected. In the second test, where both position and size of the sand bars vary, the BSS for the first three bars are more or less equal and all positive, but the scores are larger than the first three scores of the first test. This is not expected, as the positions of the bars are similar, but their sizes differ from the measured sand bar. The scores should therefore be worse.

In the first test, the values of alpha start at zero and increase towards one as the simulated bars are moving closer towards the measured sand bar. The values of alpha in the second test are increasing as well, but start at a much higher value. This is not expected, as the position of the sand bars does not change throughout the two tests, only the shape of the sand bars changes. Like the scores of the BSS, the scores for the first three deformed and shifted sand bars are higher than those of the first test. This would imply that the smaller bars represent the measurements better than the bars which have the same position which is not true.

The values of beta in the first test decrease as the bars come closer to the measured bar. In the second test however, the value of beta is increasing for the first three bars. This is wrong, as the bars move towards the measured bar, both in position and shape. For the last two bars however, which are closest to the measured bar, the values decrease again.

The values of gamma provide insight in the error of the volume of sand that is moved. In the first test, gamma therefore is almost 0, as is expected, as the size of the bars does not change. In the second test, gamma decreases as the bars grow in shape, which is correct. The results of the 3D tests are discussed next.

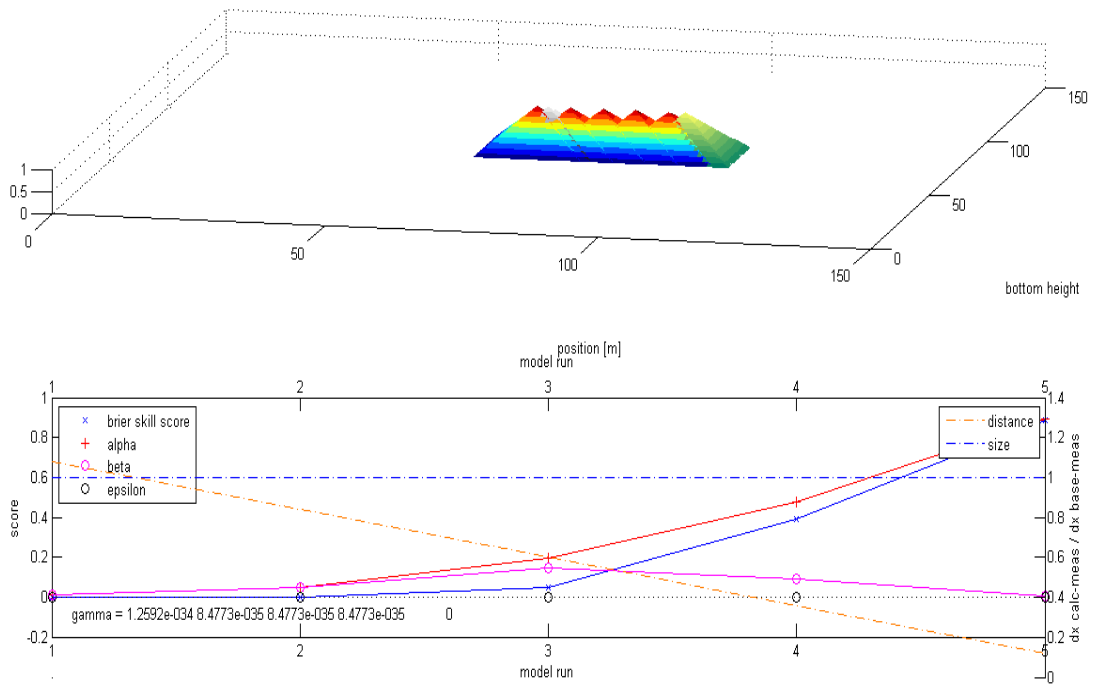


Figure 3-9 | 3D bottom profile test 1 decomposed Brier skill score values

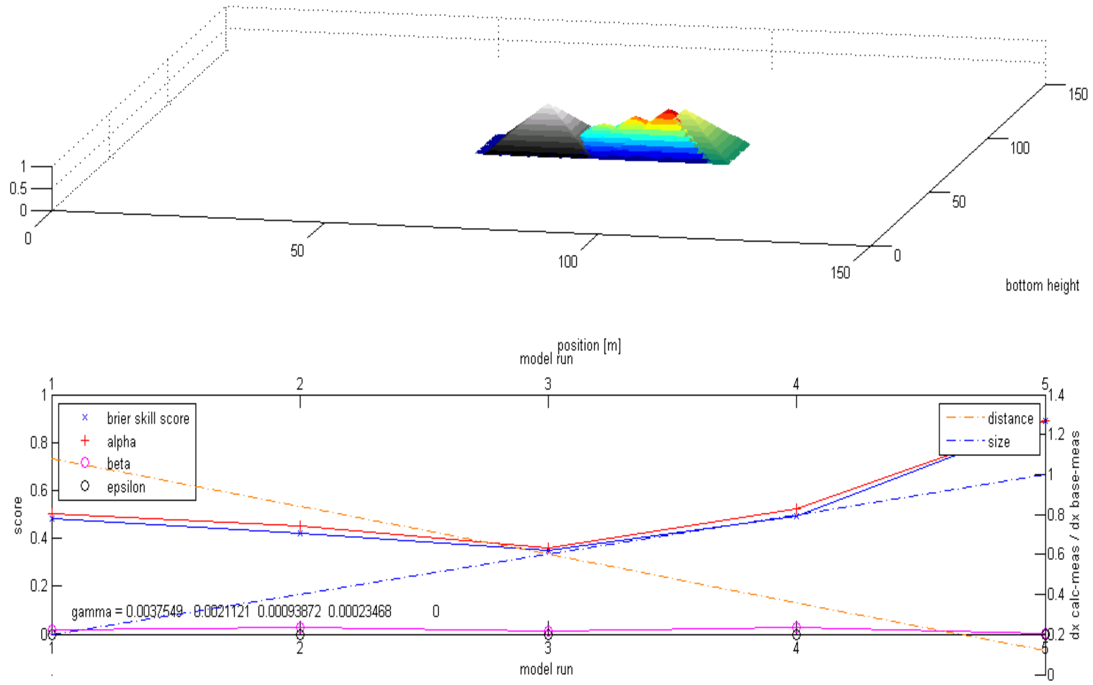


Figure 3-10 | 3D bottom profile test 2 decomposed Brier skill score values

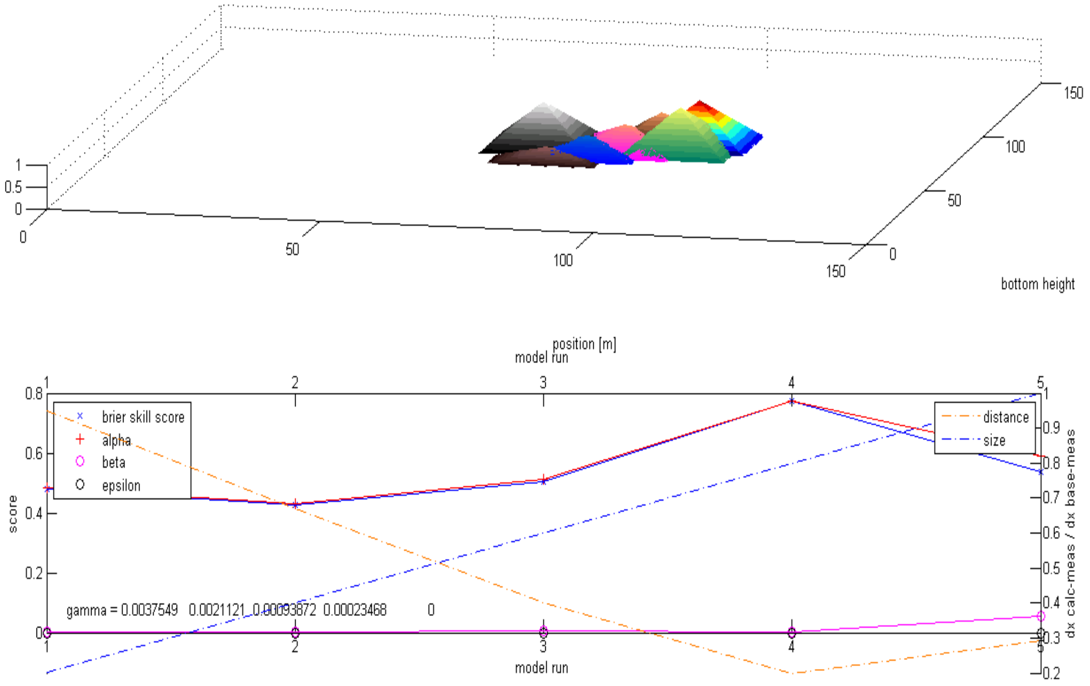


Figure 3-11 | 3D bottom profile test 3 decomposed Brier skill score values

The results of the three 3D tests are presented in Figure 3-9, Figure 3-10 and Figure 3-11. In the first test, the sand bars have only been moved from left to right. In the second test, the size of the sand bars has been altered together with the movement from left to right. In the third test, the bars are moved from left to right and top to bottom, whilst simultaneously altering the sand bars sizes. Throughout the three tests, a gray pyramid represents the start position of the simulated sand bar. The green pyramid represents the measured position of that same sand bar after a certain amount of time. In between these two sand bars, colored sand bars have been plotted that correspond to different hypothetical simulations. The separate scores corresponding to these simulated sand bars that together make up the decomposed Brier skill score are discussed one by one.

The BSS in the first test starts at zero and increases towards one as the sand bars move closer towards the measured sand bar. The value of the BSS is higher for the first three sand bars in the second test. The scores should be lower, because the size of the sand bars varies, while the position of the sand bars remains the same in both tests. Furthermore, the BSS are decreasing during the first three test, which again in unexpected, as these sand bars are increasing in size and are moving in the right direction. The BSS of the third test perform better. Although the initial values of the BSS are still too high compared to those of the first test, the trend in these values is more realistic. The values are increasing as the sand bars increase in size and move closer towards the measured sand bar.

The values of alpha of the first test look very good. Starting at 0, alpha increases as the sand bars move in the right direction. During the second test, alpha showed the same behavior as the BSS of this test; the initial values are too high and the first three scores are decreasing instead of increasing. During the third test, the trend in the line representing the alpha values looks good, but again, the initial values are too high.

The value of beta should provide insight in whether the volumes of sand that have been moved are correct. Therefore, this value should vary when the size of the sand bars is altered. The value of beta however varies most during the first test, in which the size of the sand bars has been kept constant.

Gamma represents a measure for the map mean error. It compares the average of the measurements with the average of the predictions. In the first test, gamma is very small, which is expected as the size of the simulated sand bars does not change. In tests two and three, gamma does change, reaching its highest value when the simulated sand bars are the smallest.

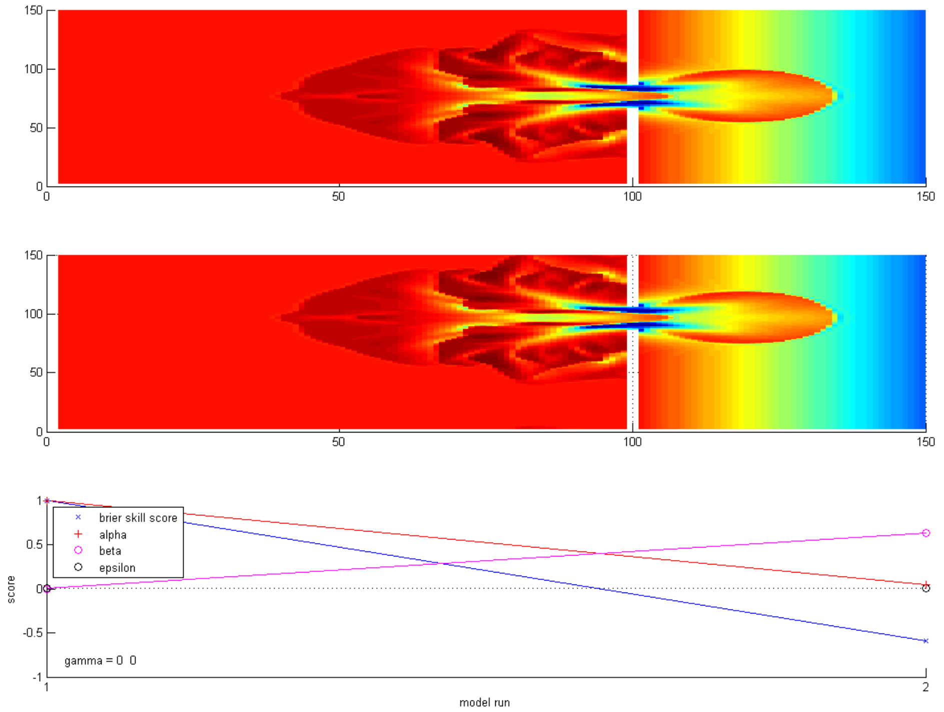


Figure 3-12 | Basin bottom profile test 1 decomposed Brier skill score values

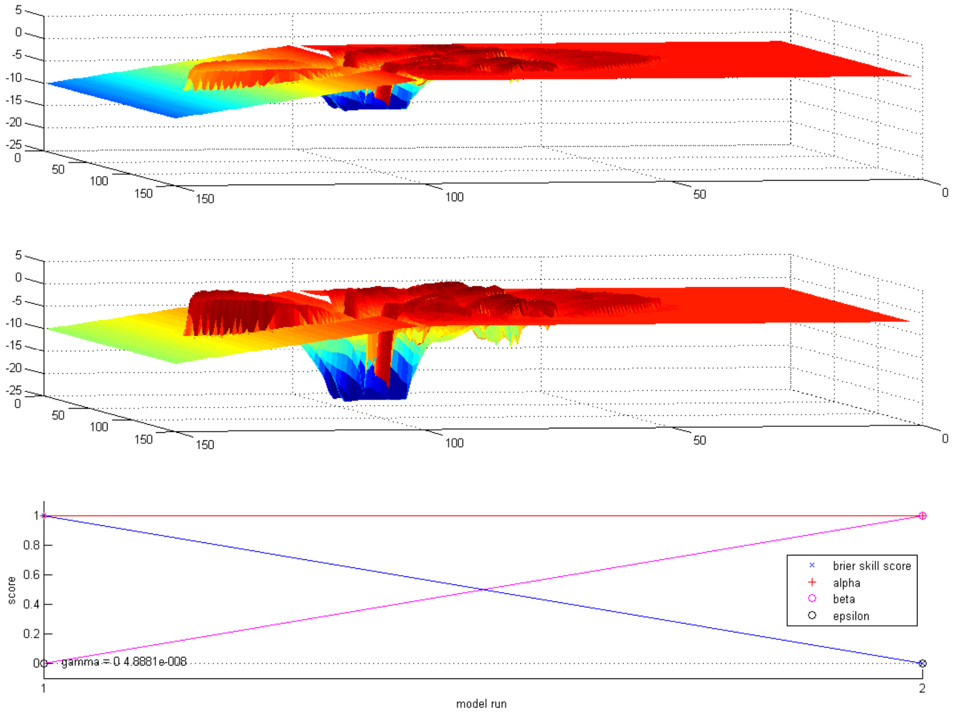


Figure 3-13 | Basin bottom profile test 2 decomposed Brier skill score values

The final tests (refer Figure 3-12 and Figure 3-13) have been performed on the basin model. In the first test, the entire bottom profile has been shifted to the left. The amplitude of the bottom fluctuations has been kept constant. In the second test, the position has been kept constant, but the fluctuations have been enlarged.

In the first test, alpha and beta respectively drop to zero and rise to 0.6, indicating that the position of the bottom profile is contributing most to the error, which is correct. The value of gamma remains constant at zero, which is also correct, as the shape and amplitude of the predicted bottom profile have not been changed. The results of the second test look promising as well. Alpha has a constant value of 1, indicating that the position of the predicted bottom is in perfect alignment with the measurements. This is correct, as the bottom has not been shifted with regards to the measurements. The values of beta and gamma on the other hand do show that the map mean error and volumes of sand that have been moved are wrong.

This section is concluded with a comment on the values of alpha and gamma with respect to the normalized distances and sizes of the sand bars. The normalized distances of the sand bars and the values of alpha show different patterns throughout the different tests. Therefore, it is not possible to express a certain alpha value in an absolute length scale. The same holds for the values of gamma. The relationship between the values of gamma and the values of the normalized sizes of the sand bars is different for the 2D and the 3D tests.

3.2.3 IMPLEMENTATION OF THE KIRCHHOFER METHOD

The Kirchhofer method has been developed in the field of synoptic weather classification. This science examines the relationship between local weather- and atmospheric circulation patterns. The Kirchhofer method is an objective classification technique which classifies synoptic type categories that are subsequently linked to weather phenomena.

This sum of squares method calculates S scores which determine whether or not two atmospheric pressure maps can be considered to be equal. The threshold values from which maps are considered to be equal or not, are user defined. For a rectangular grid, a score is calculated for the whole numerical grid, as well as for the separate rows and columns of the grid. In this thesis, the S scores will determine whether morphological measurements can be considered equal to the models predictions. The S scores between model runs can be compared to check if the models predictions are improving or not. The row and column scores can provide insights in the performance of the different parts of the grid.

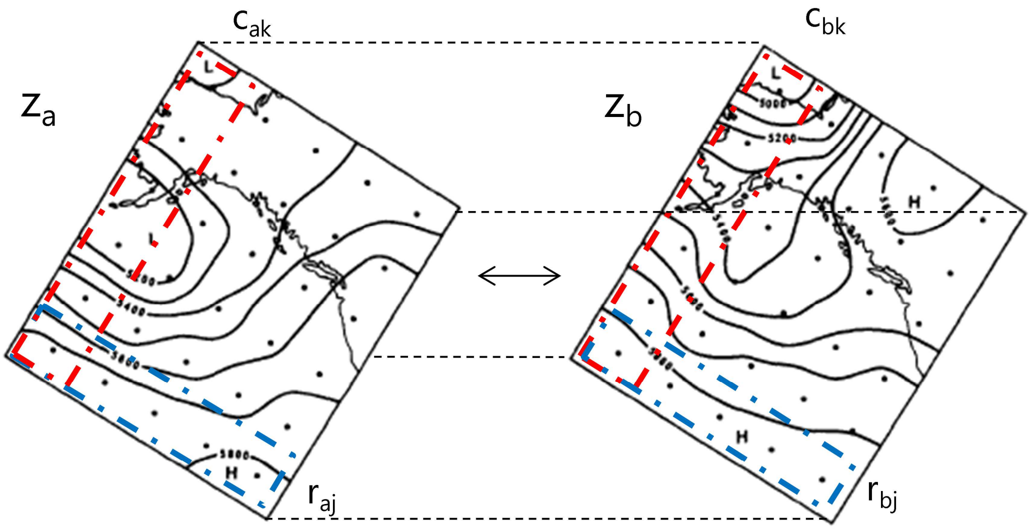


Figure 3-14 | Atmospheric pressure maps a and b

The Kirchhofer method as described in (Yarnal, 1984) consists out of two steps. The first step of the Kirchhofer method is the normalization of the dataset. For each data point, the mean of the set is subtracted and the residual is divided by the standard deviation of the set.

$$z_{ai} = \left(\frac{x_{ai} - \overline{x_a}}{s_a}\right) \tag{3-14}$$

In the second step, the normalized data sets representing model predictions and measurements are subtracted from each other, squared and summed to calculate the Kirchhofer score for the entire grid:

$$S = \sum_{i=1}^{n} (z_{ai} - a_{bi})^2 \tag{3-15}$$

Subsequently, the scores for the rows and columns are determined using formulas 3-16, 3-17, 3-18 and 3-19. (Blair, 1998) has shown that for the calculation of the row and column scores, the mean and standard deviation of the corresponding rows and columns should be used instead of the mean and standard deviation of the whole set.

$$z_{aij} = \left(\frac{x_{aij} - \overline{x_{aj}}}{s_{aj}}\right) \tag{3-16}$$

$$S_j = \sum_{i=1}^{n} \left(z_{aij} - z_{bij}\right)^2 \tag{3-17}$$

$$S_k = \sum_{i=1}^{n} (z_{aik} - z_{bik})^2 \tag{3-18}$$

$$z_{aik} = \left(\frac{x_{aik} - \overline{x_{ak}}}{s_{ak}}\right) \tag{3-19}$$

3.3 PREVIEW OF OUTPUT FROM ADJUSTED SOFTWARE

Figure 3-15 is an example of the output that is generated by the adjusted software. The figure consists of 12 graphs, which together provide all the information about a certain calibration run. The graphs from this figure will be discussed from top left down to bottom right, indicated by the row and column number (1,1 for the first graph).

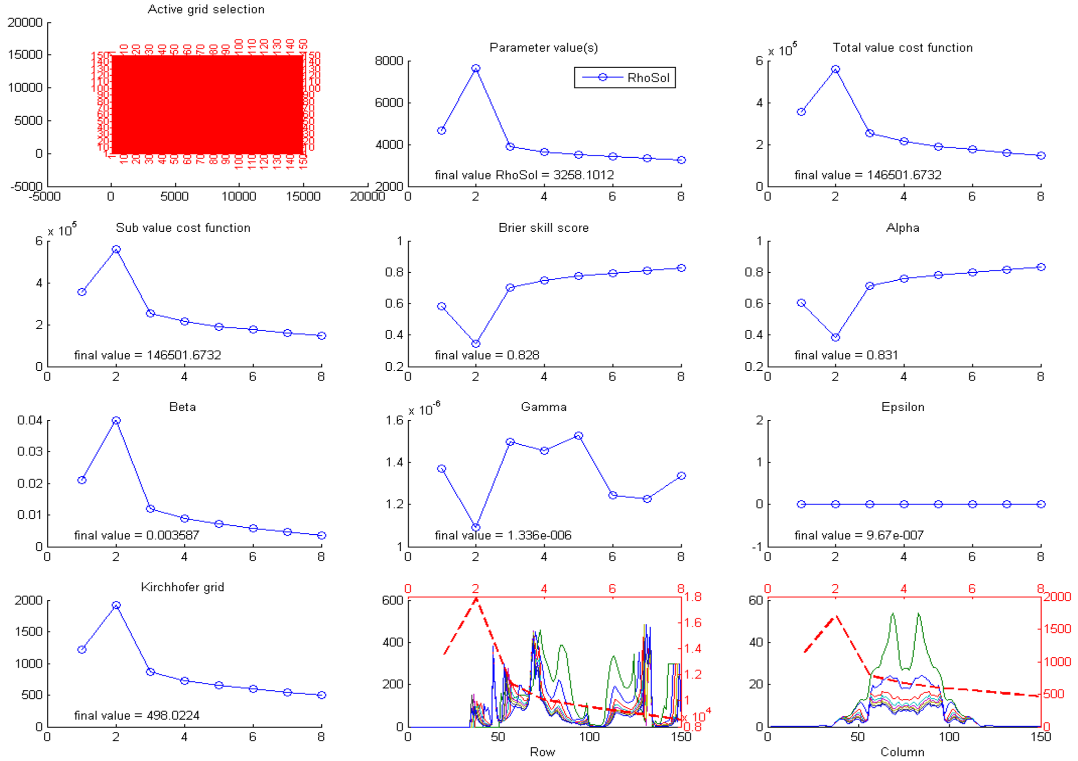


Figure 3-15 | example output TWIN experiment trench model

- Graph (1,1): Overview of the numerical grid in blue with the active grid selection in red. If multiple grid selections have been made, a full figure with all 12 graphs is generated for each grid selection. If on top of these multiple grid selections, multiple time steps are selected, a full figure is generated for each grid selection during each time step.
- Graph (1,2): Values of the parameters that have been calibrated. The values used in each iteration are plotted against the iteration number. The final value of the parameter(s) is printed in a textbox.
- Graph (1,3): The total value of the cost function corresponding to the different calibration iterations. The total value of the cost function sums the sub values of the cost function belonging to the different grid parts on different time steps.
- Graph (2,1): Sub value of the cost function, corresponding to a certain grid selection for a certain time step.
- Graph (2,2): Value of the Brier skill score.
- Graph (2,3): Value of the decomposed Brier skill score parameter α.
- Graph (3,1): Value of the decomposed Brier skill score parameter β.
- Graph (3,2): Value of the decomposed Brier skill score parameter γ.
- Graph (3,3): Value of the decomposed Brier skill score parameter ϵ.

- Graph (4,1): Value of the Kirchhofer score corresponding to the entire grid.
- Graph (4,2): Value of the Kirchhofer score corresponding to the different rows. The dotted red line sums the values of all the rows.
- Graph (4,3): Value of the Kirchhofer score corresponding to the different columns. The dotted red line sums the values of all the columns.

3.4 VALIDATION OF THE CALIBRATION TOOL: TWIN EXPERIMENTS

The alterations that have been made to the OpenDA software package have to be tested. It has to be shown that the altered software is able to calibrate morphodynamic parameters by comparing measured bottom changes with simulation results. TWIN experiments provide a means to test the calibration procedure.

3.4.1 SETUP OF TWIN EXPERIMENT

The setup of a TWIN experiments is as follows. Using a set of base parameter settings, a model application is run. The results of this simulation are referred to as the base run. The final bottom changes of this baserun simulation, with known parameter settings, are then fed into the calibration software as measurements. Next, one or more of the base run parameter values are changed and the calibration is started using these settings. If the calibration procedure succeeds, it will return the exact parameter value(s) that were used to create the measurements. Figure 3-16 shows a graphical interpretation of a TWIN experiment.

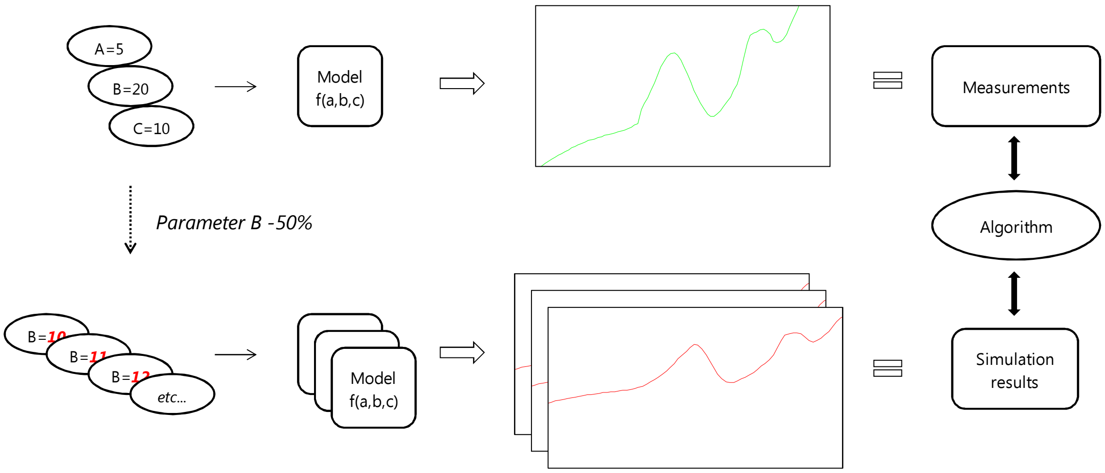


Figure 3-16 | Graphical interpretation of a TWIN experiment

Multiple types of TWIN experiments have been performed, in which different simulation settings have been changed. The variables that changed throughout the different test settings are the following; type of parameter, number of parameters, initial value parameters, parameter transformation and the initial parameter perturbations. A total number of 17 single parameters has been adjusted and calibrated using all four model applications. Furthermore, several runs have been performed with multiple parameters in different combinations and configurations.

3.4.2 ADDITIONAL OUTPUT FROM TWIN EXPERIMENTS

Before discussing the results from the TWIN experiments, an example is provided of the additional output that is generated from the single parameter TWIN experiments. Apart from the standard output graph, which shows values of the cost function, BSS and Kirchhofer method (refer Figure 3-15), an additional graph is generated, which is presented below:

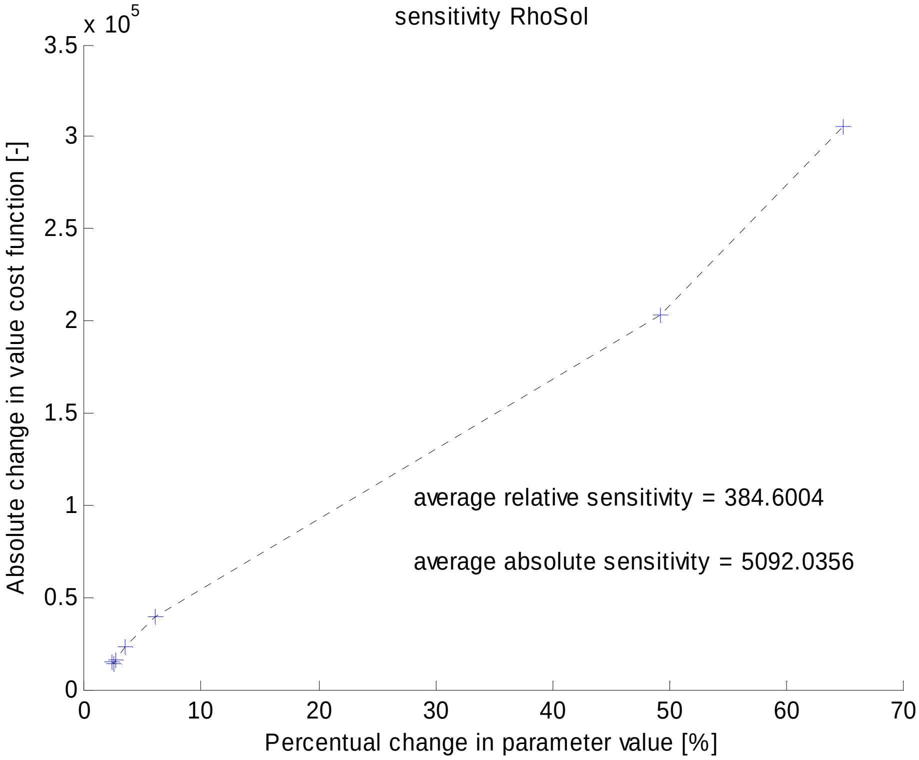


Figure 3-17 | Example of sensitivity plot single parameter TWIN experiments

The graph shows the change in parameter value, expressed as the percentual change with regards to the previous parameter value (equation 3-20), plotted against the corresponding absolute change in value of the cost function (equation 3-21).

$$x = \frac{|parameter\ value_t - parameter\ value_{t-1}|}{parameter\ value_{t-1}} \qquad 3\text{-}20$$

$$y = |value\ cost_t - value\ cost_{t-1}| \qquad 3\text{-}21$$

Every graph contains a textbox, displaying the average absolute (equation 3-22) and relative (equation 3-23) sensitivity of the parameter throughout the TWIN experiment. The average absolute sensitivity represents the average absolute change in value of the cost function per percent change of parameter value. For every point in the graph, the change in absolute value of the cost function is divided by the corresponding percentual change in parameter value. Subsequently, the average of these numbers is calculated, with n representing the number of iterations per TWIN experiment:

$$\langle|s|\rangle = \frac{\sum_1^n \frac{y}{x}}{n} \tag{3-22}$$

Calculating the average absolute sensitivity this way is valid, as long as the relationship between the change in parameter value and the absolute change in value of the cost function is linear. Whether or not this relationship is linear can be judged from the shape of the line in the graph, which has to be straight for this approach to be valid.

The average relative sensitivity is calculated by dividing the average absolute sensitivity by the average of the maximum of the values of the cost function of all 17 single parameter TWIN experiments:

$$\langle|s|\rangle \left(\frac{\sum_1^i \max(value\ cost_i)}{i}\right)^{-1} \tag{3-23}$$

Where i is the number of single TWIN experiments. As the relative sensitivity is normalized by the average maximum of the value of the cost function of every single TWIN experiment, this allows for comparison of sensitivity of parameters between the three test models.

Lastly, in Figure 3-18, an example is provided of the bottom changes that occurred during a TWIN experiment on the Egmond model. The green line shows the measurements, representing the simulation results of the base run with known parameter settings. The red line shows the final bottom depth after a run with the perturbed parameter settings, before calibration. The yellow line shows the final bottom depth after calibration. If the TWIN experiment is a success, the green and the yellow line overlap each other, which was not the case in this example.

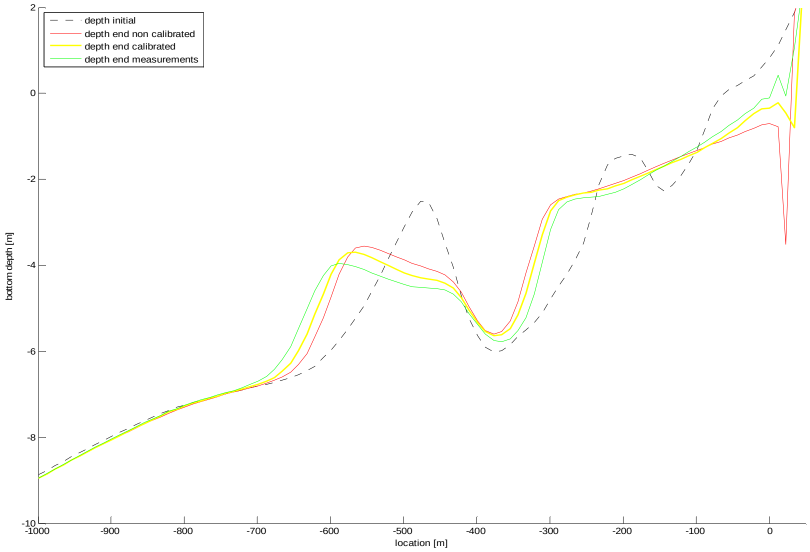


Figure 3-18 | Example figure bottom changes during TWIN experiment

3.4.3 RESULTS OF SINGLE AND MULTIPLE PARAMETER TWIN EXPERIMENTS

The results of the single and multiple parameter TWIN experiments are presented in five tables. For each model application, a table shows the outcome of the single parameter TWIN experiments, in which 17 different parameters were tested. The fifth table presents the combined results of the TWIN experiments with multiple parameters that have been peformed on the three test models. The tables show the baserun parameter values and the perturbed initial parameter values with which the experiments have been started, including the magnitude of this perturbation with regards to the baserun parameter value. In a separate column, the final values of the parameter are presented, which equal the baserun values if the experiments have been a success. If the final parameter values are either larger than 90% or smaller than 110% of the baserun values, the TWIN experiment is considered a succes, resulting in green highlighting in the tables. Failed experiments are highlighted red. An extensive explanation on the outcome of the TWIN experiments is presented in chapter 5.

Table 3-1 | Results single TWIN experiments Egmond model

Egmond							
Run	*Parameter*	*Change*	*Baserun*	*Initial*	*Final*	*Abs sens*	*Rel sens*
16	*SedDia*	-0,50	2,65E-04	1,33E-04	2,72E-04	4273,345	451,126
11	*Morfac*	0,25	1,000	1,250	0,993	3275,757	345,813
17	*RhoSol*	0,75	2650,00	4637,50	2679,57	2257,362	238,304
6	*Ccofv*	-0,2	65,000	52,000	64,751	1498,812	158,226
14	*Sus*	-0,50	1,000	0,500	0,985	687,595	72,588
4	*Dicouv*	1	1,00E+00	2,00E+00	1,00E+00	637,823	67,333
5	*Ccofu*	-0,2	65,000	52,000	66,774	476,550	50,308
9	*RDW*	-0,75	0,020	0,005	0,020	251,408	26,540
12	*RDC*	0,50	0,010	0,015	0,000	123,639	13,052
8	*BedW*	-0,50	1,000	0,500	0,922	59,711	6,303
15	*Bed*	-0,75	1,000	0,250	0,977	33,547	3,541
13	*AlfaBs*	0,75	1,000	1,750	1,723	11,112	1,173
3	*Vicouv*	1,5	1,00E+00	2,50E+00	1,60E+00	5,453	0,576
7	*SusW*	-0,25	0,100	0,075	0,104	3,638	0,384
1	*Vicoww*	-0,95	1,00E-06	5,00E-08	8,49E-08	1,991	0,210
2	*Dicoww*	1	1,00E-06	2,00E-06	2,00E-06	0,000	0,000
10	*Rwave*	-0,85	2,000	0,300	0,300	0,000	0,000

The table has been sorted on the absolute sensitivities of the parameters. This shows that the sensitive parameters for the Egmond model are SedDia, Morfac, RhoSol, Ccofv, Sus, Dicouv, Ccofu, RDW, RDC and BedW. Bed, AlfaBs, Vicouv, SusW and Vicoww do have an effect on the morphological results, but it is so small, that these parameters can be considered not sensitive. Dicoww, Rwave and do not have an effect on the morphological results at all.

Table 3-2 | Results single TWIN experiments trench migration model application

Trench							
Run	*Parameter*	*Change*	*Baserun*	*Initial*	*Final*	*Abs sens*	*Rel sens*
17	*RhoSol*	0,75	2650,00	4637,50	2650,00	15,380	460,052
16	*SedDia*	-0,50	1,60E-04	8,00E-05	1,60E-04	9,262	277,049
11	*Morfac*	0,25	15,000	18,750	15,000	8,582	256,703
12	*RDC*	0,50	0,010	0,015	0,010	7,579	226,701
14	*Sus*	-0,50	1,000	0,500	1,000	4,522	135,277
15	*Bed*	-0,75	1,000	0,250	1,000	1,921	57,448
5	*Ccofu*	-0,4	0,025	0,015	0,025	1,058	31,655
6	*Ccofv*	-0,4	0,025	0,015	0,025	0,105	3,155
13	*AlfaBs*	0,75	1,000	1,750	1,051	0,037	1,093
4	*Dicouv*	1	9,99E-06	2,00E-05	1,36E-05	0,000	0,000
3	*Vicouv*	1,5	1,00E-06	2,50E-06	1,74E-06	0,000	0,000
1	*Vicoww*	-0,95	1,00E-06	5,00E-08	5,00E-08	0,000	0,000
2	*Dicoww*	1	1,00E-06	2,00E-06	2,00E-06	0,000	0,000
7	*SusW*	-0,25	1,000	0,750	0,750	0,000	0,000
8	*BedW*	-0,50	1,000	0,500	0,500	0,000	0,000
9	*RDW*	-0,75	0,020	0,005	0,005	0,000	0,000
10	*Rwave*	-0,85	2,000	0,300	0,300	0,000	0,000

The results table above shows the results of the 17 single parameter TWIN experiments that have been performed on the trench migration model application. Sensitive parameters are RhoSol, SedDia, Morfac, RDC, Sus, Bed, Ccofu, Ccofv and AlfaBs. Non sensitive parameters are Dicouv, Vicouv, Vicoww, Dicoww, SusW, BedW, RDW and Rwave.

Table 3-3 | Results single TWIN experiments basin model

Basin							
Run	*Parameter*	*Change*	*Baserun*	*Initial*	*Final*	*Abs sens*	*Rel sens*
6	*Ccofv*	-0,4	80,000	48,000	8,00E+01	6302,167	476,005
17	*RhoSol*	0,75	2650,00	4637,50	3258,10	5092,036	384,604
11	*Morfac*	0,25	10,000	12,500	10,850	5041,667	380,799
5	*Ccofu*	-0,4	80,000	48,000	79,995	4647,949	351,062
14	*Sus*	-0,50	1,000	0,500	0,975	2378,666	179,662
16	*SedDia*	-0,50	2,00E-04	1,00E-04	1,87E-04	2315,808	174,914
12	*RDC*	0,50	0,010	0,015	0,010	855,044	64,582
4	*Dicouv*	1	1,00E+00	2,00E+00	1,00E+00	95,428	7,208
3	*Vicouv*	1,5	1,00E+00	2,50E+00	1,00E+00	86,392	6,525
15	*Bed*	-0,75	1,000	0,250	0,998	59,927	4,526
13	*AlfaBs*	0,75	1,000	1,750	1,973	0,144	0,011
7	*SusW*	-0,25	0,300	0,225	0,225	0,000	0,000
8	*BedW*	-0,50	1,000	0,500	0,500	0,000	0,000
9	*RDW*	-0,75	0,020	0,005	0,005	0,000	0,000
10	*Rwave*	-0,85	2,000	0,300	0,300	0,000	0,000

Table 3-3 shows the results of the 17 single parameter TWIN experiments that have been performed on the basin model. Sensitive parameters are Ccofv, RhoSol, Morfac, Ccofu, Sus, SedDia, RDC, Dicouv, Vicouv and bed. Non sensitive parameters are AlfaBs, SusW, BedW, RDW and Rwave.

Table 3-4 | Results single TWIN experiments Sand engine model

Sand engine							
Run	*Parameter*	*Change*	*Baserun*	*Initial*	*Final*	*Abs sens*	*Rel sens*
5	*Ccofu*	-0,4	65,000	39,000	64,300	1112466,414	528,149
6	*Ccofv*	-0,4	65,000	39,000	64,300	1001234,917	475,341
11	*Morfac*	0,25	15,000	18,750	15,816	44697,220	21,220
16	*SedDia*	-0,50	2,15E-04	1,08E-04	2,14E-04	42193,486	20,032
4	*Dicouv*	1	1,00E+00	2,00E+00	1,07E+00	6140,319	2,915
14	*Sus*	-0,50	1,000	0,500	0,977	5138,256	2,439
17	*RhoSol*	0,75	2650,00	4637,50	4637,50	2053,230	0,975
7	*SusW*	-0,25	0,200	0,150	0,195	1720,349	0,817
3	*Vicouv*	1,5	1,00E+00	2,50E+00	2,29E+00	624,223	0,296
8	*BedW*	-0,50	0,200	0,100	0,171	398,920	0,189
15	*Bed*	-0,75	1,000	0,250	0,250	348,682	0,166
13	*AlfaBs*	0,75	10,000	17,500	14,097	260,391	0,124
12	*RDC*	0,50	0,010	0,015	0,015	249,573	0,118
9	*RDW*	-0,75	0,020	0,005	0,011	248,678	0,118
10	*Rwave*	-0,85	1,000	0,150	0,150	213,861	0,102

Table 3-4 shows the results of the 17 single parameter TWIN experiments that have been performed on the Sand engine model. The only really sensitive parameters are Ccofu and Ccofv. The rest of the parameters all have an influence on the results, but it is relatively small.

Table 3-5 | Results TWIN experiments multiple parameters

Model	*Run*	*Iterations*	*Value*	*Betaro*	*SusW*	*BedW*	*Morfac*	*RDC*	*AlfaBs*	*AlfaBn*	*Dicoww*	*Vicouv*	*Sus*	*Bed*	*SedDia*	*RhoSol*
Egmond	7	10	Baserun	0,030		1,000							1,000			
			Change	-0,500		-0,500							0,500			
			Initial	0,015		0,500							1,500			
			Final	0,029		0,489							1,145			
	8	12	Baserun		0,100	1,000										
			Change		0,500	-0,500										
			Initial		0,150	0,500										
			Final		0,148	0,908										
Trench	10	49	Baserun				15,000		1,000				1,000		1,60E-04	2650,0
			Change				0,400		-0,400				-0,400		0,4	-0,4
			Initial				21,000		0,600				0,600		2,24E-04	1590,0
			Final				63,540		0,639				0,449		7,97E-05	2605,6
	11	37	Baserun						1,000				1,000		1,60E-04	2650,0
			Change						-0,400				-0,400		0,4	-0,4
			Initial						0,600				0,600		2,24E-04	1590,0
			Final						1,460				0,809		1,98E-04	2632,0
	12	35	Baserun					0,010		1,500	1,00E-06	1,00E-06		1,000		
			Change					-0,400		0,400	4,00E-01	-4,00E-01		0,400		
			Initial					0,006		2,100	1,40E-06	6,00E-07		1,400		
			Final					0,035		4,490	1,87E-08	7,20E-02		1,400		
	13	17	Baserun				15,000						1,000			
			Change				0,500						-0,500			
			Initial				22,500						0,500			
			Final				14,990						1,000			
Basin	7	17	Baserun				10,000						1,000		2,00E-04	2650,0
			Change				-0,300						0,300		3,00E-01	-0,3
			Initial				7,000						1,300		2,60E-04	1855,0
			Final				7,008						1,304		2,84E-04	2244,0
	8	14	Baserun												2,00E-04	2650,0
			Change												0,5	-0,5
			Initial												3,00E-04	1325,0
			Final												2,81E-04	2212,7

The final table of this section shows the results of the multiple parameter TWIN experiments (refer Table 3-5). The results show that the algorithm has trouble recovering the initial parameter values when more than 2 parameters are used in the experiments.

4 APPLICATION OF AN AUTOMATED CALIBRATION TOOL

The adjusted OpenDA software has been tested and implemented. Section 4.1 shows the results of a sensitivity analysis on four parameters on all three test models. The chapter concludes with the actual implementation of the calibration tool on both the Egmond, trench and Sand engine models, as described in section 4.2.

4.1 SENSITIVITY ANALYSIS OF MODEL PARAMETERS & TRANSPORT FORMULATIONS

Apart from the sensitivity analysis that was carried out based on the single TWIN experiments, another type of sensitivity analysis has been carried out in this section. This sensitivity analysis provides insights in the sensitivity of parameters, as well as providing information on the interrelationships between model parameters.

Model applications should only be calibrated using sensitive parameters, because a sensitive parameter has a large influence on the models predictions. Whether or not a parameter is sensitive depends on what results of the model are being calibrated. This depends on the goal of the calibration. When calibrating a model on morphodynamic results, it has no use choosing a parameter that has no influence on these results. A sensitivity analysis can therefore save time and decrease complexity of a calibration procedure. Furthermore, only non-inter-related parameters should be chosen when more than one parameter is calibrated. Choosing non-inter-related parameters decreases the chances of the algorithm ending up in a local minimum of the cost function solution plane.

4.1.1 SETUP OF SENSITIVITY ANALYSIS

The gridded full search algorithm has been applied to the Egmond, trench migration and basin model to perform a sensitivity analysis of four parameters; Bed, Sus, AlfaBn and SedDia. These four parameters have been coupled into six distinct pairs with which the model applications have been run while systematically altering the parameter values. For every parameter, a minimum value, maximum value and step size is prescribed. The algorithm will simply start with the minimum values of both parameters and alters their values with the step size in each run, until all possible combinations have been run.

Table 4-1 | Overview parameter pairs sensitivity runs

Run	*Parameter 1*	*Parameter 2*
1	AlfaBn	Sus
2	AlfaBn	Bed
3	AlfaBn	SedDia
4	Sus	Bed
5	Sus	SedDia
6	Bed	SedDia

Table 4-2 | Parameters values used in the sensitivity runs

Parameter	Minimum	Maximum	Step size
Sus	0,1	2,1	0,5
Bed	0,1	2,1	0,5
AlfaBn	0	50	25
SedDia	1,00E-04	3,00E-04	5,00E-05

For each model run, the Brier skill score is plotted in a surface plot against the corresponding parameter values on the x- and y-axes. Repeating equation 3-5, the BSS is calculated as follows:

$$BSS = 1 - \frac{MSE(Y,X)}{MSE(B,X)} = 1 - \frac{\langle (Y-X)^2 \rangle}{\langle (B-X)^2 \rangle}$$

Where Y would normally equal the model predictions, X the measurements and B the base run results. In case of a sensitivity analysis, the model is run once in advance, with known parameter values; the so called base run. The results of this base run are used as measurements X in the calculations, while the initial bathymetry used in the base run are read as base run values B. When altering the parameter values during the sensitivity runs, the predictions Y are produced. If these predictions have changed with regards to the measurements X, the base run with known parameter settings in this case, the models skill will decrease, indicating that a parameter is sensitive.

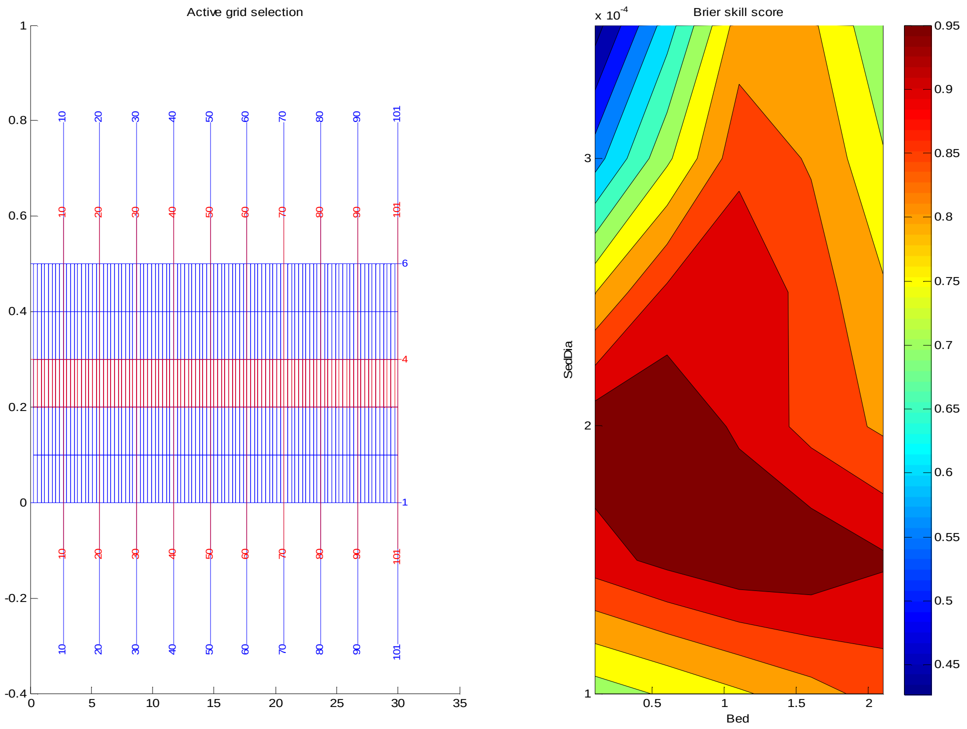


Figure 4-1 | Example of result sensitivity run

An example of the plot that is generated after a sensitivity run is presented in Figure 4-1. The numerical grid is indicated in blue and the active grid selection in red. Only the active grid selection takes part in the calculations. The surface plot shows the calculated brier skill score corresponding to the different parameter pair values, which are plotted on the x- and y-axes.

One additional sensitivity run has been performed with each of the three test models to investigate the sensitivity of the model to the various transport formulations that are described in section 2.3. The models are run with fixed parameters settings, only varying the transport formulations. A BSS is than calculated for each of the runs.

4.1.2 RESULTS OF SENSITIVITY ANALYSIS

Sensitivity of model parameters

Although the results are presented in this chapter, the reader is directed to chapter 5 for a more extensive explanation of the parameter sensitivities and the parameter inter-relationships. The outcome of the 6 sensitivity runs of all three model applications is presented in Appendix A. The information from these graphs is summarized in tables below:

Table 4-3 | Bandwidth Brier skill score sensitivity runs Egmond model

Model	Run	Parameter	BSS min	BSS Max	Range	Sensitivity		Parameter	Range	Sensitivity
Egmond	1	AlfaBn	0,20	0,30	0,10	Low		SedDia	2,50	High
		Sus	-0,35	0,60	0,95	High		Sus	1,60	High
	2	AlfaBn	0,30	0,40	0,10	Low		SedDia	1,60	High
		Bed	0,30	0,50	0,20	Medium		Sus	0,95	High
	3	AlfaBn	0,00	0,00	0,00	Low		Sus	0,90	High
		SedDia	-2,00	0,50	2,50	High		SedDia	0,60	High
	4	Sus	-0,30	0,60	0,90	High		Bed	0,20	Medium
		Bed	0,00	0,00	0,00	Low		AlfaBn	0,10	Low
	5	Sus	-0,80	0,80	1,60	High		Bed	0,10	Low
		SedDia	-0,80	0,80	1,60	High		AlfaBn	0,10	Low
	6	Bed	0,50	0,60	0,10	Low		Bed	0,00	Low
		SedDia	0,00	0,60	0,60	High		AlfaBn	0,00	Low

From the sensitivity runs with the Egmond model, it becomes clear that SedDia in combination with Sus can have a big influence on the simulation results. Sus and SedDia are always highly sensitive, whilst Bed and AlfaBn hardly have an influence on the models results. The following conclusions can be drawn on the relationships between parameters:

- The influence of Sus increases, as SedDia decreases
- The influence of Bed increases, as AlfaBn decreases
- The influence of Bed increases, as SedDia increases

The original setup of the parameter values and step sizes is presented in Table 4-1. These bounds have been adjusted for some of the runs, as it became clear that some of the combinations yielded unrealistic model results, leading to numerical instability of the model runs. This was especially true for the combination of Sus and SedDia in the Egmond model, which can be explained by the fact that both parameters separately are very sensitive and are inter-related as well.

Table 4-4 | Bandwidth Brier skill score sensitivity runs trench migration model

Model	Run	Parameter	BSS min	BSS Max	Range	Sensitivity		Parameter	Range	Sensitivity
Trench	1	AlfaBn	0,00	0,00	0,00	Low		Sus	0,70	High
		Sus	0,78	0,98	0,20	Medium		Sus	0,60	High
	2	AlfaBn	0,00	0,00	0,00	Low		SedDia	0,50	High
		Bed	0,88	0,98	0,10	Low		Bed	0,35	High
	3	AlfaBn	0,00	0,00	0,00	Low		SedDia	0,30	High
		SedDia	0,83	0,98	0,15	Medium		Bed	0,25	Medium
	4	Sus	0,30	0,90	0,60	High		Sus	0,20	Medium
		Bed	0,50	0,85	0,35	High		SedDia	0,15	Medium
	5	Sus	0,20	0,90	0,70	High		Bed	0,10	Low
		SedDia	0,60	0,90	0,30	High		AlfaBn	0,00	Low
	6	Bed	0,45	0,70	0,25	Medium		AlfaBn	0,00	Low
		SedDia	0,45	0,95	0,50	High		AlfaBn	0,00	Low

As with the Egmond model, Sus and SedDia are the most sensitive parameters. AlfaBn does not have even the slightest of influences on the morphological developments. Bed is somewhere in between, but compared to the other model applications, the influence of Bed is the largest in the trench model. About the interrelationships between parameters, the following can be said:

- The influence of Sus increases, as SedDia decreases
- The influence of Bed increases, as SedDia increases

Table 4-5 | Bandwidth Brier skill score sensitivity runs basin model

Model	Run	Parameter	BSS min	BSS Max	Range	Sensitivity		Parameter	Range	Sensitivity
Basin	1	AlfaBn	0,00	0,00	0,00	Low		Sus	0,60	High
		Sus	0,40	0,90	0,50	High		Sus	0,52	High
	2	AlfaBn	0,98	0,99	0,01	Low		Sus	0,50	High
		Bed	0,98	0,99	0,01	Low		SedDia	0,40	High
	3	AlfaBn	0,00	0,00	0,00	Low		SedDia	0,22	Medium
		SedDia	0,73	0,95	0,22	Medium		SedDia	0,22	Medium
	4	Sus	0,35	0,87	0,52	High		Bed	0,03	Low
		Bed	0,00	0,00	0,00	Low		AlfaBn	0,01	Low
	5	Sus	0,30	0,90	0,60	High		Bed	0,01	Low
		SedDia	0,40	0,80	0,40	High		AlfaBn	0,00	Low
	6	Bed	0,73	0,76	0,03	Low		AlfaBn	0,00	Low
		SedDia	0,73	0,95	0,22	Medium		Bed	0,00	Low

Table 4-5 shows that Sus and SedDia are the most sensitive parameters in the basin model. However, compared to the other two model applications, SedDia has a smaller influence. Both Bed and AlfaBn hardly have any influence on the models outcome. The following can be concluded on the interrelationships between the parameters:

- The influence of Bed increases, as SedDia increases
- The influence of Sus increases, as SedDia decreases

Sensitivity of transport formulation

The results of the transport formula sensitivity runs are presented in Figure 4-2, Figure 4-3 and Figure 4-4. The figures contain two plots. The upper plot shows the values of the BSS corresponding to the different transport formulations. The second plot shows the final bottom depths belonging to these scores and transport formulations. The results are summarized in Table 4-6:

Table 4-6 | Overview Brier skill scores, model applications and transport formulations

Iteration	*Transp. Form.*	*Egmond*	*Trench*	*Basin*
1	van Rijn 1993	1,000	1,000	1,000
2	Meyer Peter Muller	0,047	0,515	0,113
3	Engelund-Hansen	0,254	0,447	0,361
4	Bijker	-0,372	0,708	0,182
5	van Rijn 2004	0,739	0,852	0,605
	Range	1,372	0,553	0,887

The Brier skill scores are calculated using the results of the base runs as measurements. A BSS of 1 indicates that the predictions of the model application are similar to the base run values, calculated using the van Rijn 93 transport formulation. The table therefore shows the difference between the van Rijn transport formulation and the other formulations. The BSS is not calculated using the actual measured data. The actual measured data however is plotted in the figures of the Egmond and trench model (dashed black line), to indicate which transport formula performs best.

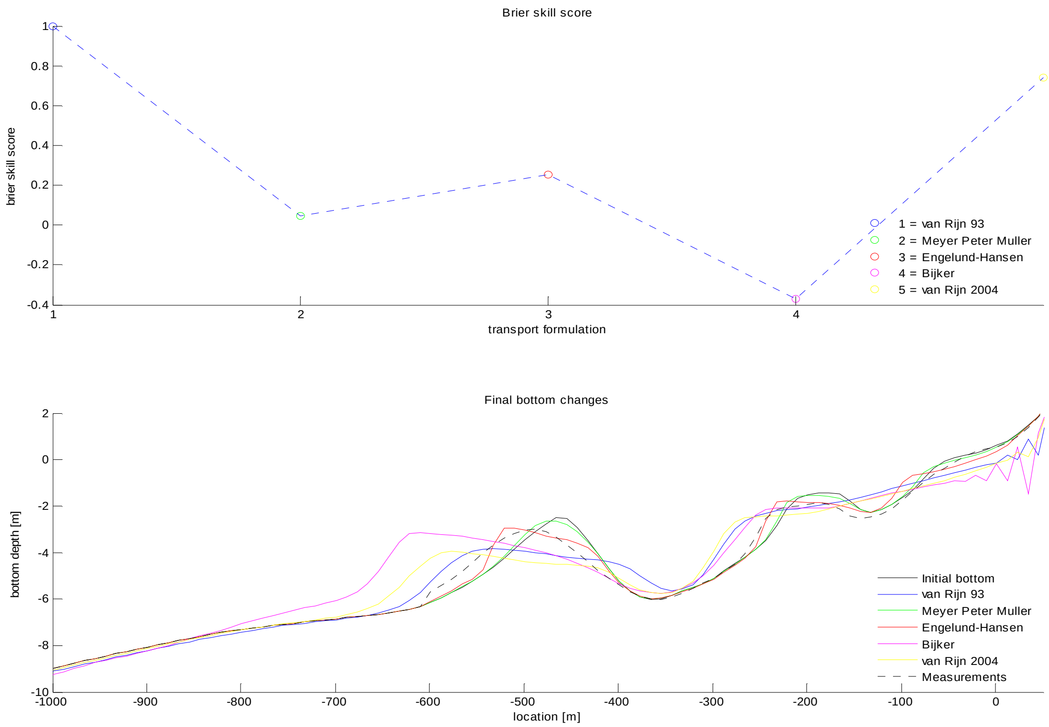


Figure 4-2 | Results transport formulation sensitivity run Egmond model

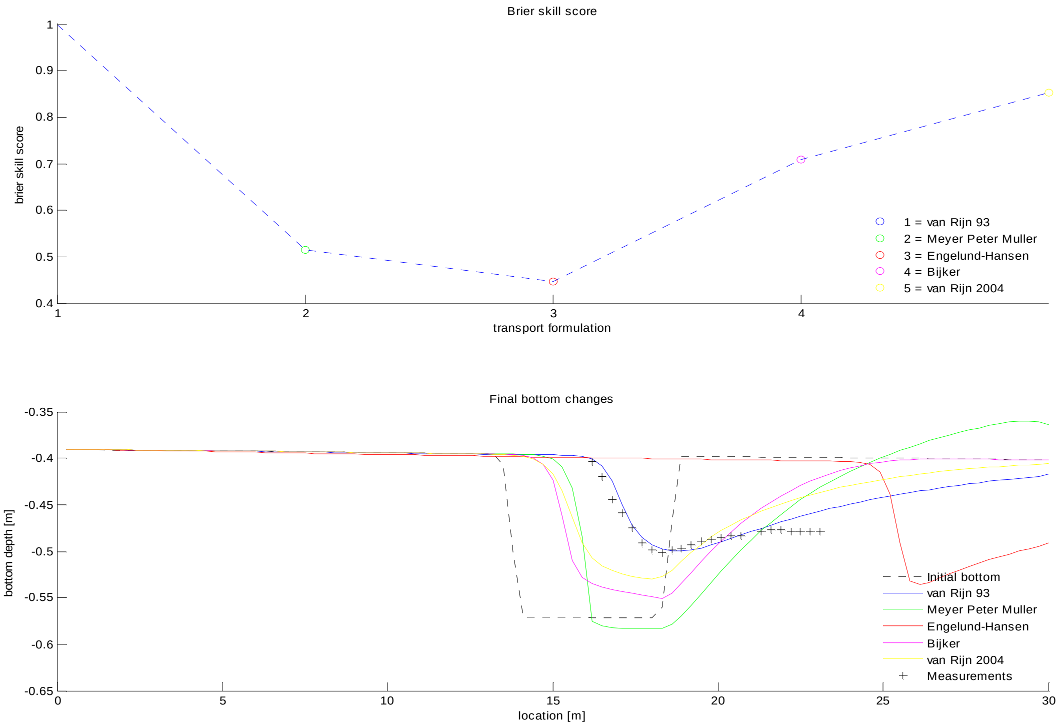


Figure 4-3 | Results transport formulation sensitivity run trench migration model application

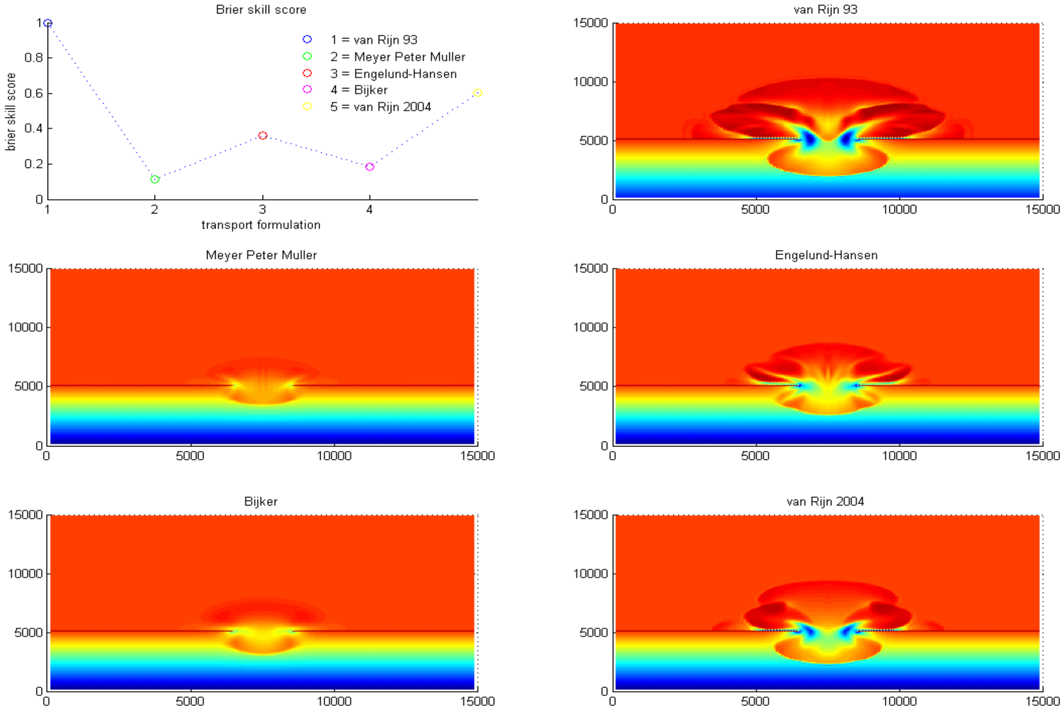


Figure 4-4 | Results transport formulation sensitivity run basin model

The table shows that the difference between the outcomes of the simulations in which the van Rijn 1993 and the van Rijn 2004 transport formulations were used is the smallest, indicated by the highest BSS in all three tests. This was expected, as the van Rijn 2004 formulation builds upon the principles of the van Rijn 1993 transport formulation. The bottom plots show that the simulation results depend very much on the type of transport formula used; the transport formulation is thus a very sensitive parameter which has to be chosen with care. Furthermore, these plots show that different transport formulations perform best in the Egmond and trench models; van Rijn 1993 performs best in the trench model, whereas in the Egmond model, van Rijn 2004 produces the fit with the actual measured data. Vice versa, different transport formulations also perform worst in both model applications. This again shows that choosing a transport formulation is not as straightforward as it might seem.

4.2 CALIBRATION CASES

The final step in the development of the calibration instrument is the application of the instrument. This implies calibrating the different model applications for which actual measured data is available. In this section, calibration runs have been performed on the Egmond, trench migration and Sand engine models. The measurements have been used to calibrate different model parameters using different setups of the models. The first set of calibration runs has been performed to find out whether OpenDA can pinpoint sensitive parameters automatically. Next, two different parameters have been calibrated using both the Egmond and trench models to check for similarities in the optimum parameter values. Last, a calibration run has been performed on the Sand engine model, to check whether this complex morphodynamic model application can be improved using the calibration instrument.

4.2.1 CAN OPENDA PINPOINT SENSITIVE PARAMETERS AUTOMATICALLY?

The first calibration case centres on the trench migration model application. The goal of this calibration is to investigate whether the calibration instrument can pinpoint sensitive parameters automatically. The word automatic is used here to indicate that no separate sensitivity analysis has to be performed to indicate the sensitive parameters.

To investigate whether this is the case, three calibration runs have been performed. From the sensitivity analysis of the previous chapters, it has become clear which parameters are sensitive and which are not. In the first run, two sensitive parameters, SedDia and Sus are calibrated. In the next run, one non-sensitive parameter is added to the calibration. In the third run, two non-sensitive parameters are added. If the results of all three calibration runs produce similar values for the sensitive parameters and the values of the non-sensitive parameters stay close to their initial values, it would not matter how many non-sensitive parameters are added to the calibration. In that case, any set of parameters can be chosen in a calibration, in which then only the sensitive parameters would change value. This saves valuable time, as no sensitivity analysis would have to be performed. The results of the three calibration runs are presented in Figure 4-5, Figure 4-6 and Figure 4-7.

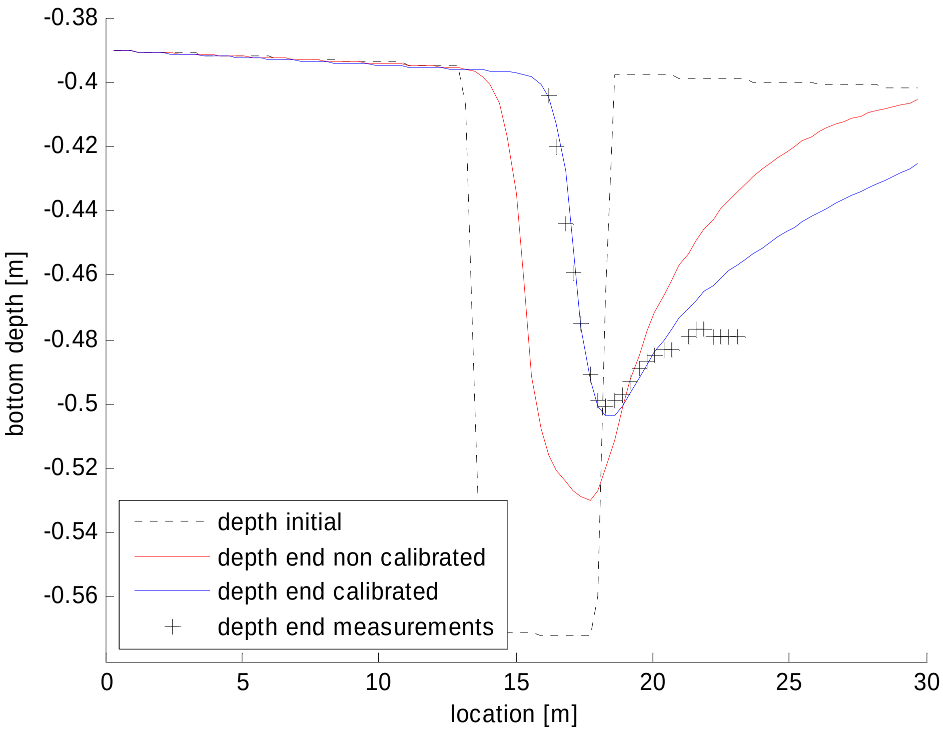


Figure 4-5 | Results run 1 calibration trench

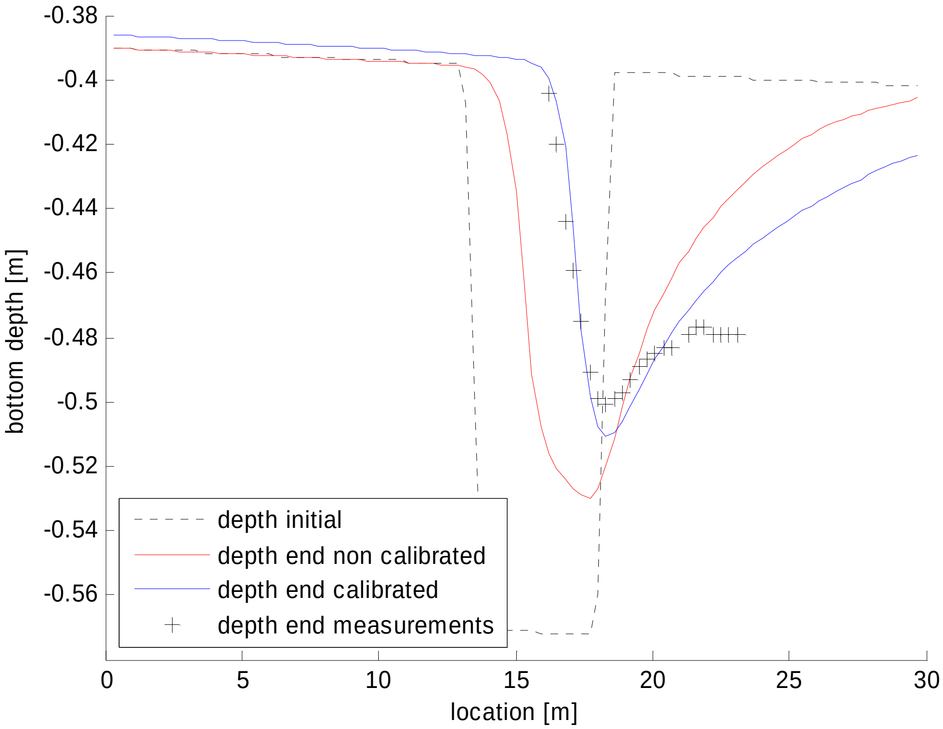


Figure 4-6 | Results run 2 calibration trench

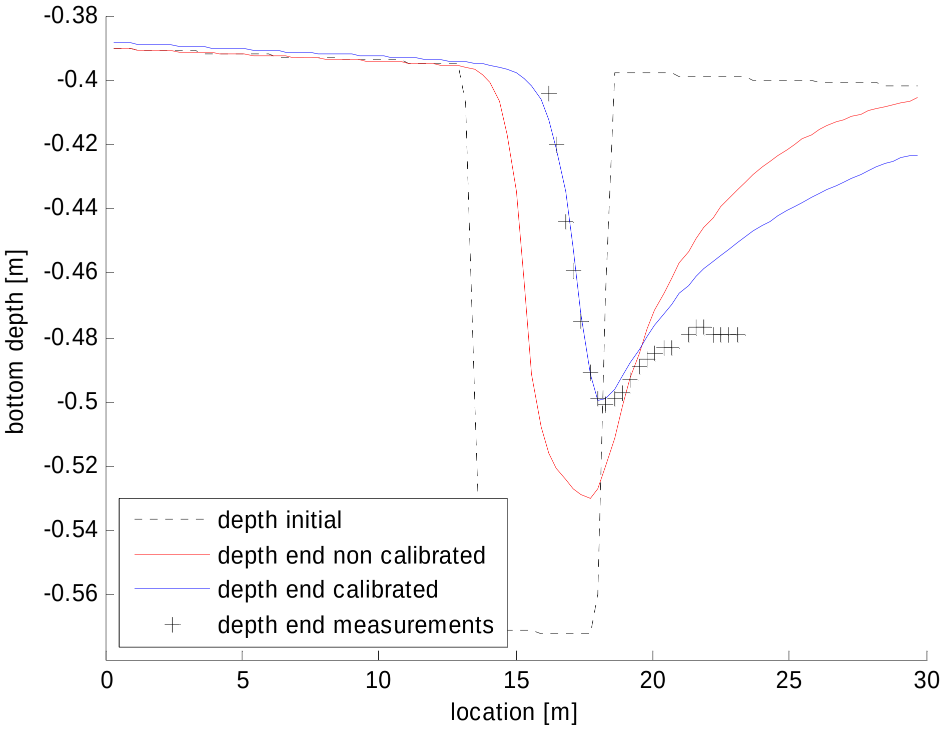


Figure 4-7 | Results run 3 calibration trench

Table 4-7 presents numerical data corresponding to the three calibration runs. The initial and final values of the parameter, as well as the cost function are shown. The table also shows the percentual improvement of the cost function and the amount of iterations that were needed.

Table 4-7 | Overview calibration runs trench model

				Cost function				
Run	*Parameter(s)*	*Initial value*	*Final value*	*Initial value*	*Final value*	*Δ cost abs*	*Δ cost rel*	*Iterations*
1	SedDia	1,60E-04	2,13E-04	247	11	236	95,41%	52
	Sus	1	2,2643					
2	SedDia	1,60E-04	2,27E-04	247	16	231	93,40%	44
	Sus	1	3,0616					
	Ccofv	0,025	0,0155					
3	SedDia	1,60E-04	2,04E-04	247	17	230	93,12%	39
	Sus	1	2,078					
	Ccofv	0,025	0,0212					
	AlfaBs	1	10,2247					

Unfortunately, the values of SedDia and Sus vary for each run. The values of the cost functions however are almost the same. This implies that different parameter values can lead to comparable values of the cost function. Furthermore it can be seen that although the cost function values are similar, the bottom results corresponding to these values are not. This makes it even harder to interpret the results. Concluding, it is not possible for the calibration instrument to pinpoint sensitive parameters automatically.

4.2.2 CALIBRATING TWO PARAMETERS USING DIFFERENT MODELS APPLICATIONS

The second calibration case aims to investigate whether calibrated parameter values differ when different model applications are used. Furthermore, the goal is to determine the influence of the transport formula on the parameter values. In an ideal situation, it would not matter what model application or transport formula is used. No matter what the model setup, the optimum parameter values would be the same. This would imply that the physics underlying the governing morphodynamic processes are represented well by the Delft3D model.

To investigate whether this is true, eight different calibration runs have been performed using both the Egmond and the trench migration model application. Furthermore, two different transport formulations have been applied. The runs have been performed using two different sensitive parameters, SedDia and Sus. Table 4-8 presents an overview of the different runs that have been performed:

Table 4-8 | Setup calibration runs

Run	Model	Parameter	Transp. form.
1	Egmond	SedDia	Rijn 93
2	Egmond	SedDia	Rijn 2004
3	Egmond	Sus	Rijn 93
4	Egmond	Sus	Rijn 2004
5	trench	SedDia	Rijn 93
6	trench	SedDia	Rijn 2004
7	trench	Sus	Rijn 93
8	trench	Sus	Rijn 2004

The bottom changes during the eight calibration runs are presented in Figure 4-8 through Figure 4-15.

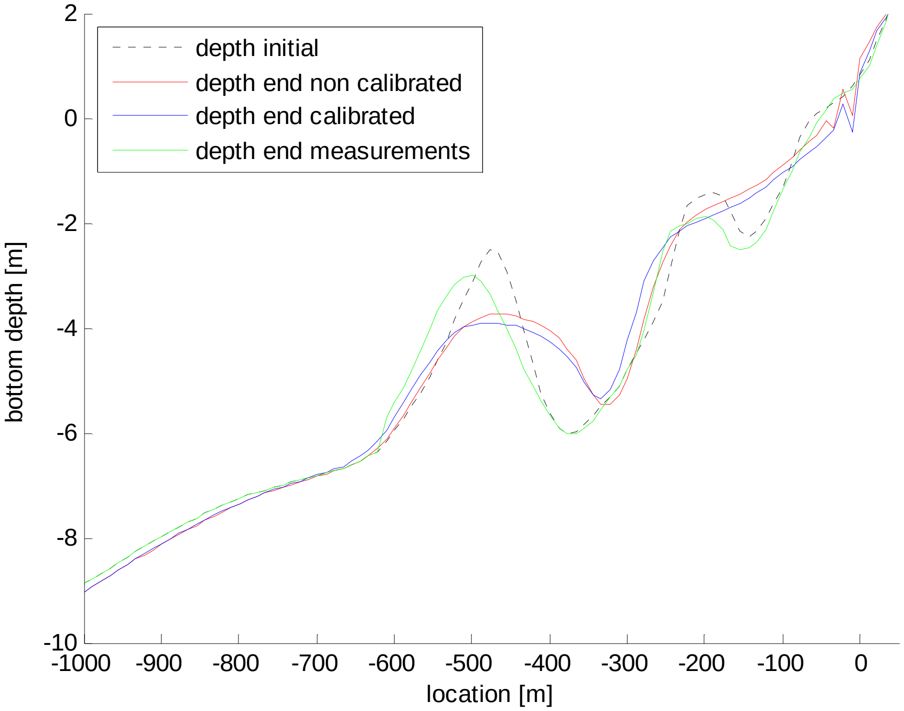


Figure 4-8 | Results run 1 Egmond model

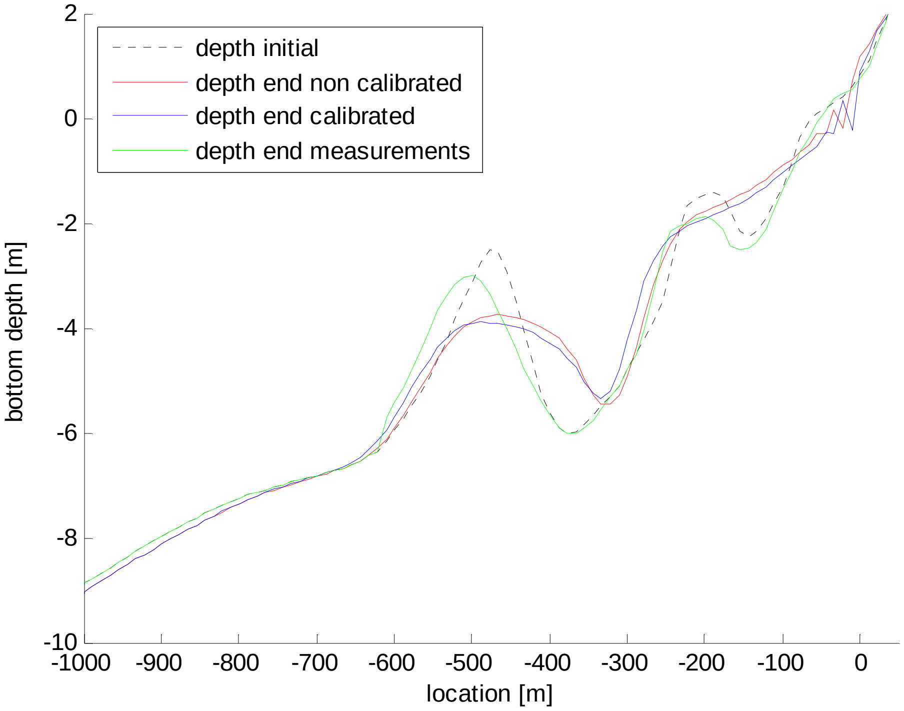


Figure 4-9 | Results run 3 Egmond model

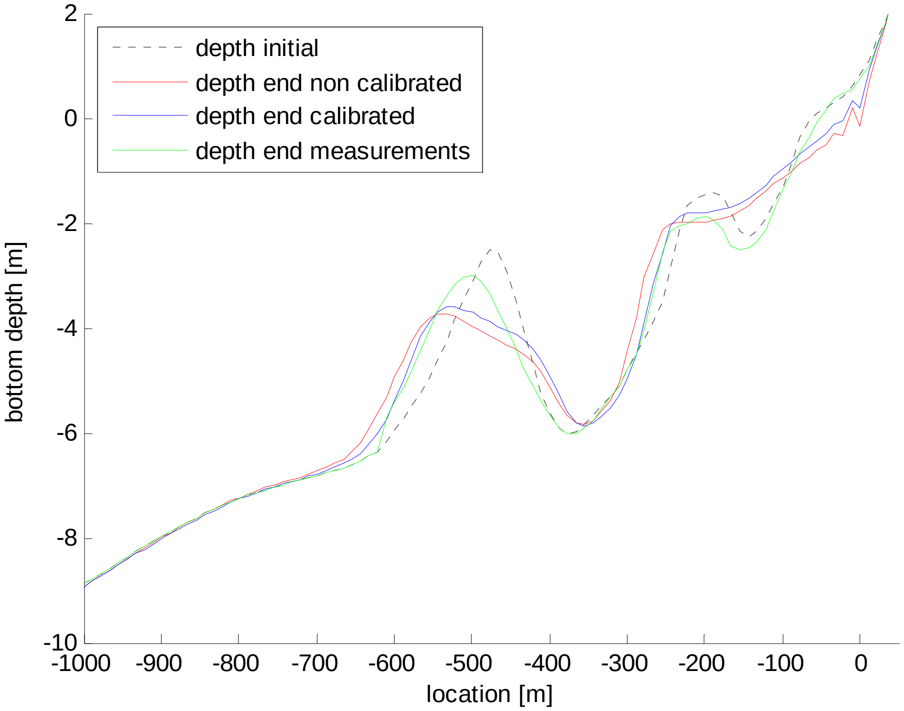


Figure 4-10 | Results run 2 Egmond model

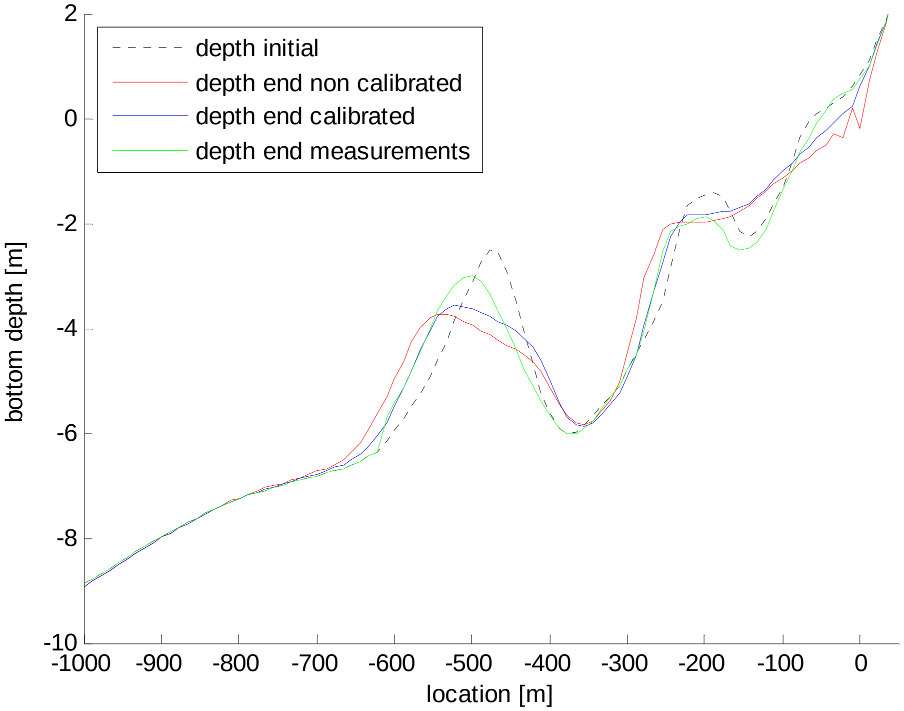


Figure 4-11 | Results run 4 Egmond model

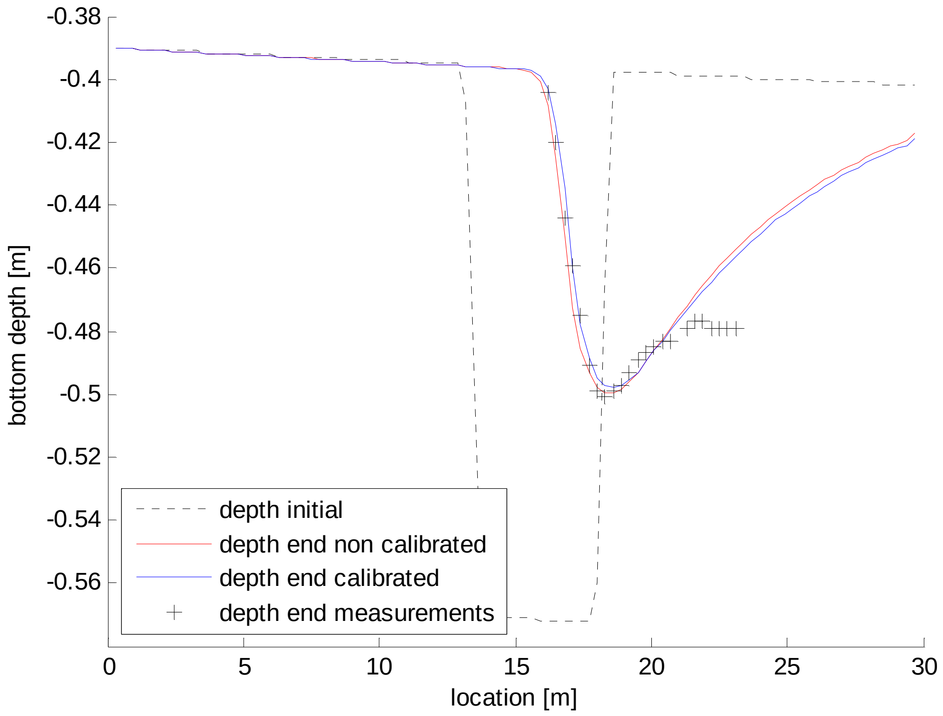


Figure 4-12 | Results run 5 trench model

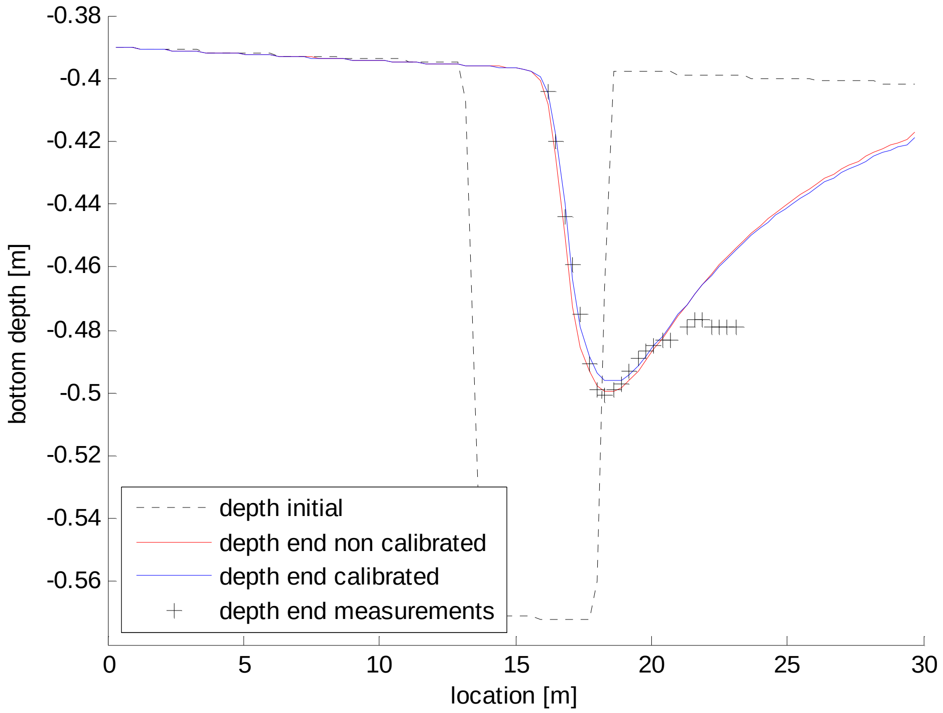


Figure 4-13 | Results run 7 trench model

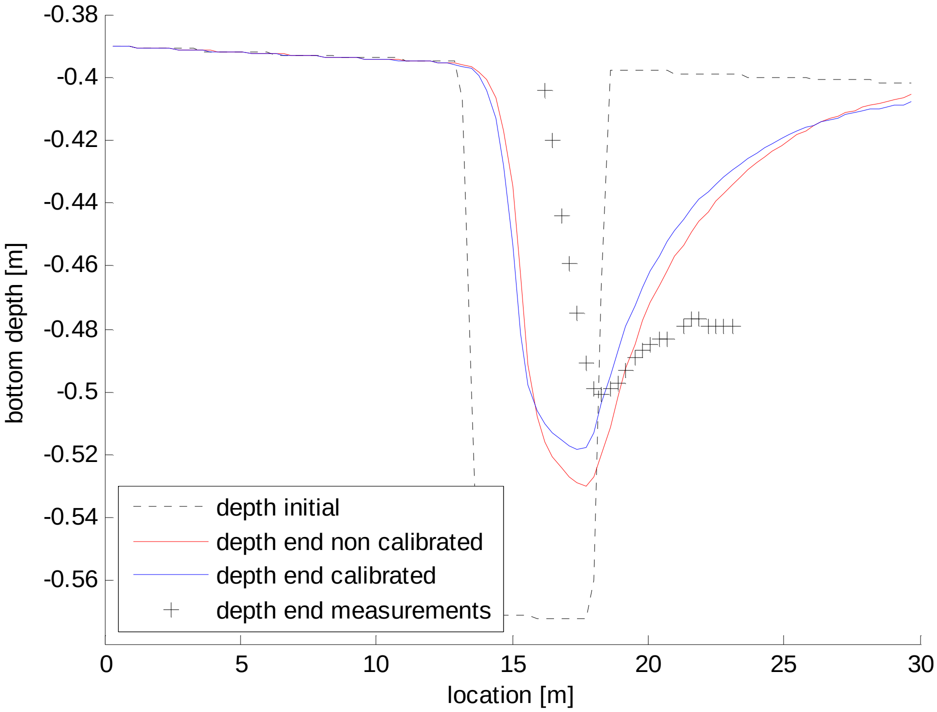


Figure 4-14 | Results run 6 trench model

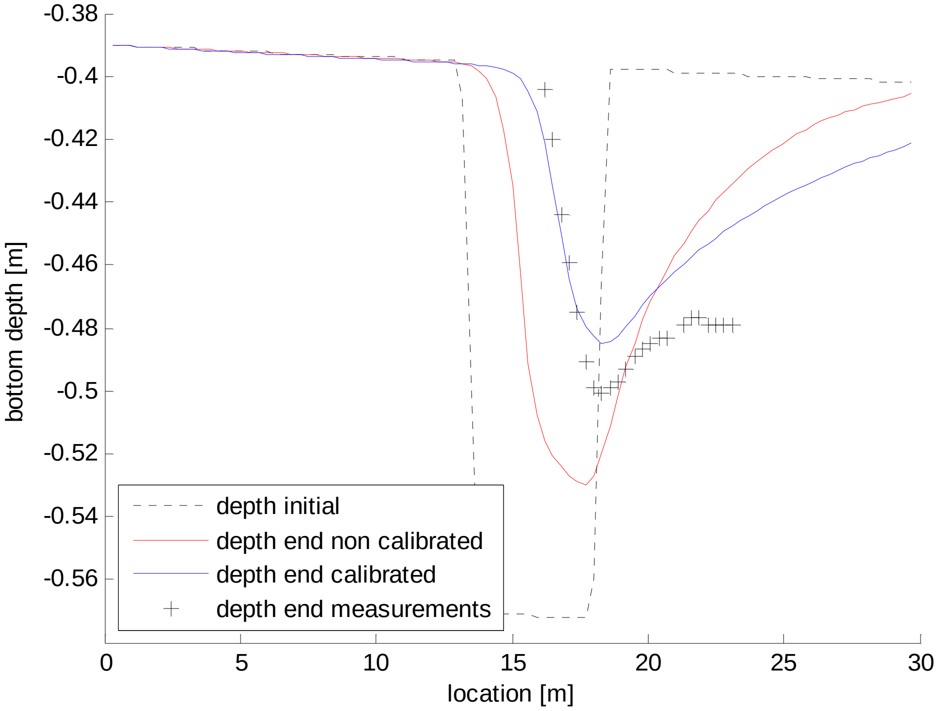


Figure 4-15 | Results run 8 trench model

The results of the eight calibration runs are summarized below in Table 4-9 and Table 4-10:

Table 4-9 | Results calibration runs Egmond model

Egmond						*Cost function*			
Run	*Parameter*	*Transp. form.*	*Initial value*	*Final value*	*Iterations*	*Initial value*	*Final value*	*Δ cost abs*	*Δ cost rel*
1	SedDia	Rijn 93	2,65E-04	2,49E-04	6	163783	139425	24358	14,9%
2	SedDia	Rijn 2004	2,65E-04	4,02E-04	6	83973	58341	25632	30,5%
3	Sus	Rijn 93	1,00	1,19	17	160603	136128	24475	15,2%
4	Sus	Rijn 2004	1,00	0,58	5	82847	47471	35376	42,7%

Table 4-10 | Results calibration runs trench model

Trench						*Cost function*			
Run	*Parameter*	*Transp. form.*	*Initial value*	*Final value*	*Iterations*	*Initial value*	*Final value*	*Δ cost abs*	*Δ cost rel*
5	SedDia	Rijn 93	1,60E-04	1,64E-04	53	12,3	9,0	3,3	26,9%
6	SedDia	Rijn 2004	1,60E-04	1,19E-04	47	247,2	234,0	13,2	5,3%
7	Sus	Rijn 93	1,00	1,03	56	12,3	10,5	1,8	14,9%
8	Sus	Rijn 2004	1,00	1,77	55	247,2	37,6	209,6	84,8%

From the tables, it immediately becomes clear, that none of the optimum parameter values are the same. The optimum values differ depending on the transport formulation chosen and on the model application used. This leads to believe that by calibrating these parameters, one is not necessarily improving the physics underlying the models. The models are tuned to fit the data at hand. This implies that the morphodynamic processes can still be improved in the Delft3D model.

A final note on the parameters that were chosen in the calibration; although Sus and SedDia are strongly inter-related, this does not play a role in the calibrations of SedDia. This is true, because the values of Sus are similar in both model applications. The inter relationship between the parameters could play a role in the calibration runs of Sus, as the values of SedDia are not similar in both runs. Choosing two sensitive, non-inter-related parameters could therefore improve the calibration results, i.e. result in optimum parameter values that are closer to each other.

4.2.3 CALIBRATING THE SAND ENGINE MODEL

The final calibration case revolves around the Sand engine model. This is the most complex of the four model applications and was developed and applied during the design phases of the actual project. Because of its complexity, it takes a lot of time to simulate one month of morphological developments around the Sand engine; more than 7 days for one model run. A typical calibration using two parameters will require 10 up to 50 iterations before the optimum values are determined. This would imply that the calibration could take up almost one year before completion. For practical reasons, this is not possible.

To be able to use the calibration instrument on the Sand engine, the runtime has to be drastically decreased. As measurements become available every four weeks, the model is normally run for a full month. In (Man, 2012) research has been done on the short term development of the Sand engine. The hydrodynamic conditions around the Sand engine have been classified and the impact on the short term morphology was determined. From this analysis it became clear that most of the morphological developments of the Sand engine took place during storms. The storms in the months of July, August and September are summed in Table 4-11. The number and total duration of storms was the smallest in the month of August. Furthermore, all the storms took place within a five day span. To reduce the runtime of the Sand engine model, it will therefore be run only for those five days in August.

Table 4-11 | Duration of storm events on Sand engine in 2011

Runs	*#*	*Start*	*Time*	*End*	*Time*	*Duration (hours)*
5-7 / 3-8	jul-01	12-jul	8:00	13-jul	17:00	33
	jul-02	13-jul	17:00	14-jul	3:00	10
	jul-03	14-jul	3:00	15-jul	16:00	37
	jul-04	16-jul	12:00	19-jul	0:00	60
	jul-05	22-jul	12:00	25-jul	0:00	60
					Total	***200***
3-8 / 4-9	aug-01	6-aug	19:00	8-aug	4:00	33
	aug-02	9-aug	0:00	10-aug	0:00	24
	aug-03	10-aug	14:00	12-aug	0:00	34
					Total	***91***
4-9 / 16-10	sep-01	5-sep	13:00	8-sep	13:00	72
	sep-02	11-sep	17:00	14-sep	14:00	69
	oct-01	5-okt	11:00	6-okt	14:00	27
	oct-02	6-okt	14:00	7-okt	11:00	21
	oct-03	7-okt	11:00	9-okt	0:00	37
	oct-04	9-okt	22:00	11-okt	18:00	44
					Total	***270***

The approach of running only five days in which the storms of August took place has to be validated. This approach is only valid when the majority of the morphological developments took place during those five days. If nothing happened during the rest of the month, there is no use of simulating these days. Figure 4-16 shows six plots with information on the morphological developments during a full month run of August:

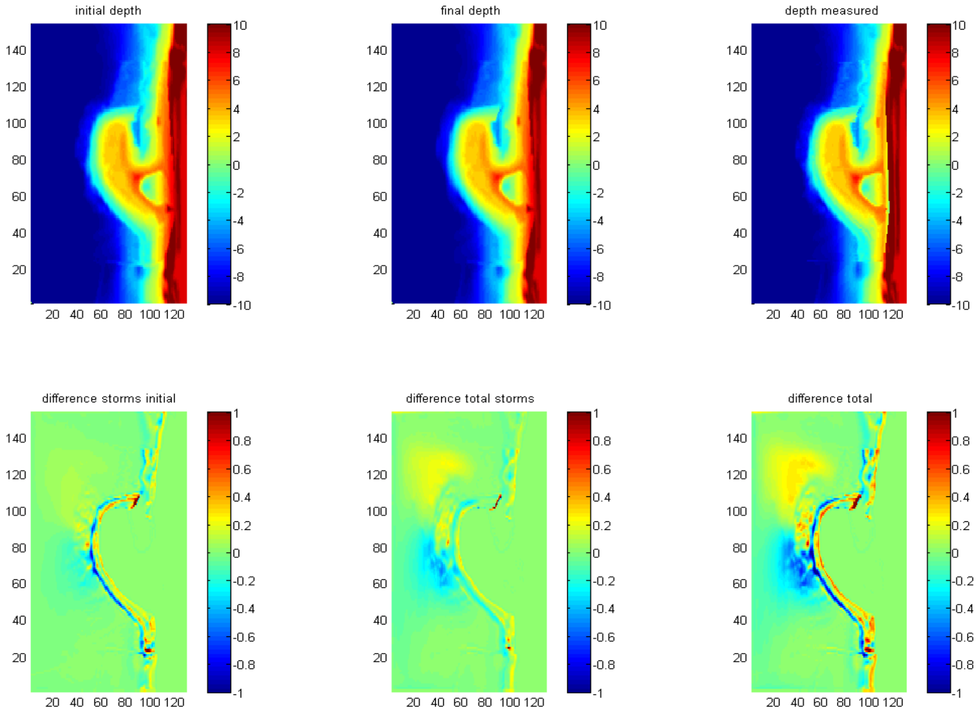


Figure 4-16 | Bottom changes before and after storms August

The top three pictures show respectively the initial bottom profile, the final profile after a full month run and the measured bottom profile at the end of August. These figures give an indication of the performance of the model. The bottom three pictures contain information about the morphological development that took place within the full month run. The first picture shows the changes to the bottom profile after the five days of storm in the beginning of the month. The areas indicated by red and dark blue are influenced the most. After those five days however, there are still almost 20 days left of running. The second plot shows the morphological changes that took place in those 20 days. In an ideal situation, there would have been no additional changes to the bottom profile in those remaining 20 days, but this is not the case. Finally, in the final plot, the total difference in bottom profile between the start and the end of the run are presented.

Although not all morphological changes took place during the five days of storm, there are area's in the model were almost all bottom changes were caused by the storms. The bottom changes after the storms can therefore be neglected by using only these area's in the calibration. The active grid selection that has been chosen to calibrate the model on is located in the far north east of the Sand engine. The parameter that have calibrated are Sus and SusW. The results of the calibration are presented in Figure 4-17.

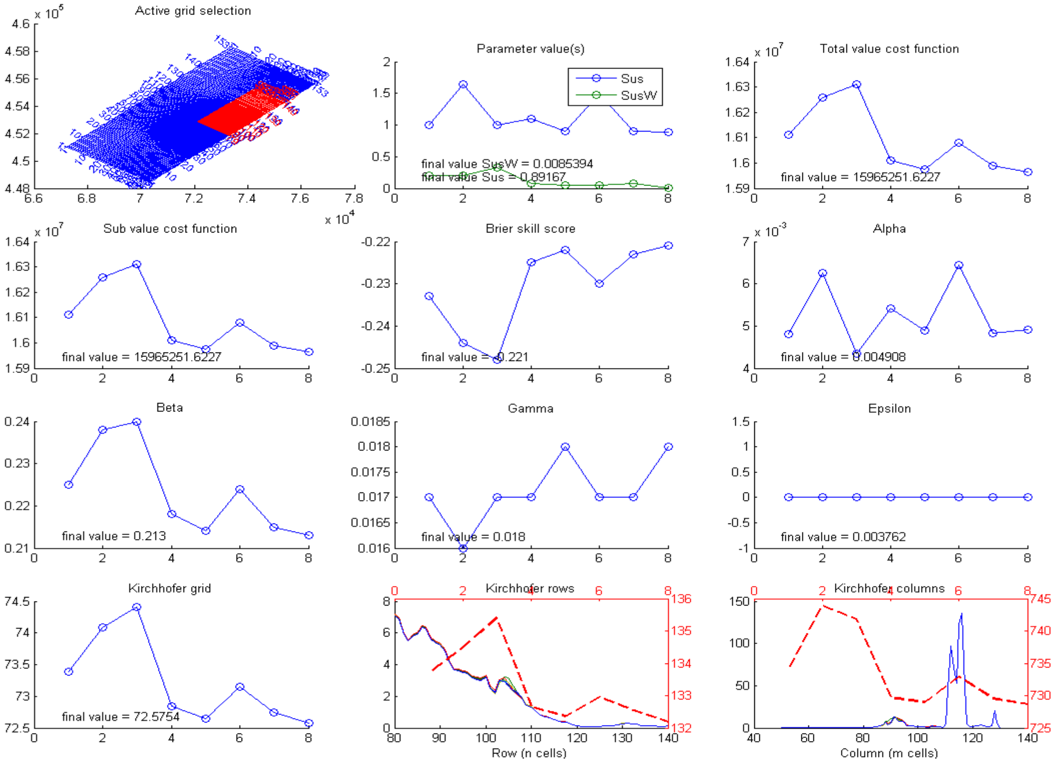


Figure 4-17 | Output calibration run Sand engine

The results show that the calibration instrument is able to improve the performance of the Sand engine model as the cost function decreases in value. However, when reviewing these values more closely, the cost function drops from 16111794,74 to 15965251,62 which equals an improvement of 146543,11 or 0,91%. This is improvement therefore is negligible. The BSS is negative and the decomposed values show that both the position and the volumes and the mean are not modeled accurately. The difference between the measurements and the model results before and after the calibration are presented below:

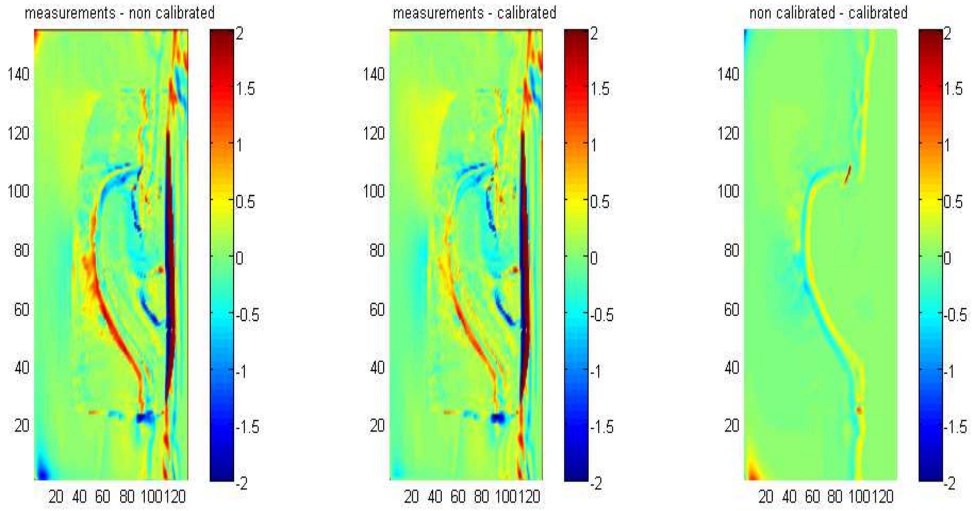


Figure 4-18 | Bottom results after calibration run

5 DISCUSSION OF RESULTS TESTS & CALIBRATION RUNS

After developing, testing and implementing the calibration instrument, many lessons have been learned. This chapter aims to give an overview of the most important conclusions from the previous chapters of the report, as well as providing an explanation for these conclusions.

5.1 LESSONS LEARNED FROM TWIN EXPERIMENTS

5.1.1 FACTORS DETERMINING SUCCESS OF TWIN EXPERIMENT

By determining what factors contribute to the success of the TWIN experiment, rules of thumb can be deduced for the setup of a 'real' calibration. Applying these handles will reduce the chance of the algorithm ending up in a local minimum of the cost function. Furthermore, a proper setup of a calibration will reduce the amount of runs that are needed to determine the optimum parameter values.

During the single TIWN experiments, when the parameters turned out to be sensitive, the experiments were successful. No matter whether the initial perturbation was small or large, negative or positive, the algorithm recovered the base run values for all these different settings. It does not matter whether the initial parameter value is higher or lower than the optimum value. It only influences the amount of runs that are needed. More runs are needed when the initial parameter value is lower than the optimum, as the initial parameter perturbation always is positive. Vice versa, when a parameter turned non-sensitive, the experiment failed, no matter how small the initial perturbation, negative or positive. Therefore, only sensitive parameters should be included in a calibration procedure.

If more than 2 parameters are calibrated at once, there is relatively high change of the search algorithm ending up in a local minimum of the cost function. Especially when parameters are calibrated that are interrelated, chances are slim that the algorithm will return the initial parameter values. In other cases, were multiple parameters were calibrated, the value of the cost function did look very good, but for a completely different set of parameter values. This shows the danger of calibrating multiple, inter-related parameters. The parameter settings are adjusted in such a way to best fit the measurements, but this does not imply that the physical processes are represented better after calibration. It is therefore recommended not to calibrate more than two parameters at once.

When initial values are chosen that differ more than 75% with regards to their optimum settings, it is not very likely that the calibration method will recover this optimum. If the amount of outer iterations is large enough, OpenDA can still recover the optimum value if a parameter is sensitive. This however requires a lot of runs. Well-chosen initial values minimize the amount of runs needed and maximize the chance of the algorithm finding an actual optimum. It is therefore important to choose the initial parameter values with care; if possible based on physical knowledge of the processes involved.

The Dud algorithm has shown to be able to recover the initial parameter values in case of the single parameter TWIN experiments within a reasonable amount of runs. Average numbers of runs during the single TWIN experiments were comparable for all four model applications; Basin = 9.1; Egmond = 13.3; trench = 4.6; Sand engine = 11.2.

Specifying the initial parameter perturbation identity or lognormal does not influence the results much. A lognormal distribution however does prevent parameter values of becoming negative, which is useful for almost all morphological parameters. Furthermore, applying an initial parameter perturbation of 0.5 will ensure that the search algorithm finds the optimal value of the parameters in a reasonable amount of iterations.

5.2 ABSOLUTE AND RELATIVE PARAMETER SENSITIVITY

In both chapters 3 and 4, information has been gathered with regards to the sensitivity of the different model parameters. This section bundles that information and provides explanations for these findings.

5.2.1 WHY ARE PARAMETERS (NON) SENSITIVE?

First, it is explained which parameters are sensitive. Bed, SusW and BedW are all factors which are used to multiply the effects of currents and waves on the bed load and suspended load transport. These parameters therefore have a large influence on the morphological changes. Sus is a factor that is used in the calculation of the reference concentration $c_a^{(l)}$ (refer equation 2-28) at van Rijn's reference height a, which is directly linked to the suspended sediment transport. When the reference concentration at the reference height increases, there is more sediment available for suspended transport. Morfac is a factor that is used to directly multiply the erosion and deposition fluxes that are interchanged between the bed and the flow at every time step (refer equation 2-22). SedDia and RhoSol, the sediment diameter and sediment fraction density, determine to a large extend the magnitudes of both the bed load and suspended load fluxes and are found in multiple formulas used in the morphological calculations. The bottom roughness Ccofu and Ccofv influence both the flow as well as the bed shear stress calculations (refer equation 2-4), which are directly linked to the sediment transport. Dicouv and Dicoww, the horizontal and vertical Eddy diffusivities, directly influence the suspended sediment fluxes (refer equations 2-23, 2-25 & 2-26). The suspended load represents the governing transport flux in most models, implying that both Dicouv and Dicoww can influence the morphological developments. RDW and RDC are related to the current and wave related roughness k_c and k_w which determine the roughness of the bottom (refer equation 2-10). These parameters therefore influence the flow as well as the sediment calculations. The parameter Rwave is also used to calculate k_w. This wave related roughness directly influences both the flow and the magnitude of the reference concentration at the reference height a. This parameter should thus have an influence on the simulation results.

Next, the reasons are presented for the non-sensitivity of the remaining parameters. AlfaBs and AlfaBn, the longitudinal and transverse bed slope correction factors, correct the bed load for slopes in the longitudinal and transverse direction (refer equations 2-13 & 2-14). Depending on the governing flow direction, one of the two scaling factors dominates the other. The effects of these slopes however are relatively small. The contribution of the bed load to the total transport is rather small as well in most model applications, which decreases the sensitivity of AlfaBn and AlfaBs even further. The horizontal and vertical Eddy viscosities are a means of mixing of momentum in the flow; the viscosities of the flow (molecular and artificial) represent a means to distribute the forces in the flow. Furthermore, the Eddy viscosities are implemented to account for energy losses due to turbulence on the smallest spatial scales. The vertical and horizontal viscosities however only play a role in case gradients are found in the velocities of the flow: $\nu_t^H \frac{\partial u}{\partial x}$ & $\nu_t^V \frac{\partial v}{\partial z}$ (refer equations 2-1 & 2-2), which is not always the case.

The grid selection can also have an influence on the sensitivity of the parameters. The active grid selection preferably has chosen as far away from the edges of the numerical grid as possible. At the edges, near the boundaries, only little morphological change takes place, which would imply that none of the parameters is sensitive. Furthermore, not every time step is suitable in a sensitivity analysis, as was found during the runs of the trench model. If a too large time step was chosen, the trench had already left the system.

A final remark is made with regards to the sensitivity of the transport formulations. The results from the sensitivity analysis of section 4.1 have shown that the simulation results are very much dependent on the choice of transport formulation. From these simulations it also became clear that different transport formulations resulted in the best representation of the actual measured data. Choosing a transport formulation therefore is not as straightforward as it may seem and should be done with care, depending on the governing processes in the model application.

5.2.2 WHY DOES THE PARAMETER (NON)-SENSITIVY DIFFER THROUGHOUT MODEL APPLICATIONS?

Both the single TWIN experiments of chapter 3 and the sensitivity analysis of chapter 4 provided valuable insights in the sensitivity of all tested parameters throughout the different model applications. The top 5 sensitive parameters of all model applications are very much comparable. The sensitivity of parameters however does differ throughout the model applications. An explanation for these differences is provided in this sub-section.

Table 5-1 shows the values of the relative sensitivities of the parameters in each model application in three different columns. The difference between the columns is found in the number of values n that have been used to calculate the average values (refer equation 3-22). Although the relative sensitivities vary somewhat for different values of n, the ranking of the parameters sensitivities per model stays the same; the most sensitive parameter still is the most sensitive. Also, the sensitivity of one parameter compared between the model applications does not change. This leads to believe that the approach used to calculate the parameter sensitivities is valid. Not all values from the sensitivity plots are used in the calculation, because these values become very small in the final iterations. Both the parameter perturbations and the changes in absolute values of the cost function can approach values of O(10^{-3} - 10^{-7}). Using these values in the calculations of the relative sensitivity leads to unrealistic values.

Table 5-1 | Overview relative sensitivities parameters

		Values used = 2				Values used = 3				Values used = 4			
Model		*Egmond*	*Trench*	*Basin*	*Sand*	*Egmond*	*Trench*	*Basin*	*Sand*	*Egmond*	*Trench*	*Basin*	*Sand*
Run	*Parameter*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*	*Rel sens*
1	*Vicoww*	0,157	0,000	0,000	0,000	0,210	0,000	0,000	0,000	0,223	0,000	0,000	0,000
2	*Dicoww*	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
3	*Vicouv*	0,658	0,000	9,251	0,386	0,576	0,000	6,525	0,296	0,680	0,000	9,475	0,409
4	*Dicouv*	92,309	0,000	9,925	4,079	67,333	0,000	7,208	2,915	51,625	0,000	5,516	2,241
5	*Ccofu*	74,380	47,093	414,220	613,314	50,308	31,655	351,062	528,149	40,740	31,655	350,350	616,697
6	*Ccofv*	158,226	4,589	410,257	700,669	158,226	3,155	476,005	475,341	158,226	3,155	367,948	376,918
7	*SusW*	0,177	0,000	0,000	0,817	0,384	0,000	0,000	0,817	0,752	0,000	0,000	0,817
8	*BedW*	4,415	0,000	0,000	0,250	6,303	0,000	0,000	0,189	5,979	0,000	0,000	1,602
9	*RDW*	6,867	0,000	0,000	0,118	26,540	0,000	0,000	0,118	22,137	0,000	0,000	0,118
10	*Rwave*	0,000	0,000	0,000	0,032	0,000	0,000	0,000	0,102	0,000	0,000	0,000	0,094
11	*Morfac*	487,338	231,429	418,164	27,765	345,813	256,703	380,799	21,220	261,101	224,163	347,922	16,664
12	*RDC*	10,369	298,974	71,017	0,107	13,052	226,701	64,582	0,118	11,554	176,502	54,011	0,309
13	*AlfaBs*	1,036	1,579	0,012	0,142	1,173	1,093	0,011	0,124	11,904	1,093	0,011	0,167
14	*Sus*	87,037	197,397	174,523	2,743	72,588	135,277	179,662	2,439	58,722	101,520	169,767	1,970
15	*Bed*	5,016	57,448	0,730	0,030	3,541	57,448	4,526	0,166	3,130	57,448	3,490	0,182
16	*SedDia*	556,972	350,266	121,675	26,270	451,126	277,049	174,914	20,032	338,975	213,946	201,137	15,955
17	*RhoSol*	293,417	686,116	333,470	0,035	238,304	460,052	384,604	0,975	202,598	638,415	413,466	0,810

Vicouv & Vicoww; The horizontal and vertical Eddy viscosities are a means of mixing of momentum in the flow. Forces are distributed in the flow via the viscosities of the flow; the higher the viscosity, the large the area of the flow over which the forces are distributed. Therefore, the values of the Eddy viscosities have a direct influence on the flow. The flow in return determines the bed shear stresses and thus the sediment transport, implying that the viscosity also has an influence on the sediment transport. This influence can be seen in the Egmond and Basin models. Simulations of these models resulted in vertical flow patterns that vary over the water depth. These gradients in the velocity profile are distributed over the water column via the viscosities. The velocity profile in the trench model however hardly contains any gradients. The influence of the horizontal Eddy viscosity can therefore be neglected. In both 3D model applications, almost no vertical flows are present, implying that Vicoww is of minor importance. As the basin model is two dimensional, Vicoww does not exist.

Dicouv & Dicoww; The horizontal Eddy diffusivity Dicouv is used in the computation of the concentration field in a water column, as well as directly influences the current related suspended transport rates. According to equations 2-23, 2-25 & 2-26, the diffusivities only play a role when gradients are present in the concentration fields; $\epsilon_{s,x,y,z}\frac{\partial c}{\partial x,y,z}$. The influence of Dicouv is largest in the Egmond model, but almost zero in the other model applications, indicating that the gradients are much smaller in the trench and basin models. The vertical Eddy diffusivity does not have an influence in any of the models. In the basin model, only 1 layer is present. Therefore, Dicoww does not exist. For the two 3D model applications, this implies that there is no vertical mixing of sediment in between layers. This is explained by the fact that almost all flow patterns are horizontally orientated.

Ccofu & Ccofv; Ccofu and Ccofv are very sensitive parameters in all model applications. The bottom roughness directly influences both the flow as well as the shear stress related to the flow. The flow is directly influenced as the roughness partly determines the bottom boundary condition posed in the momentum equation that is solved by Delft3D. However, there are some differences in the sensitivity of these parameters throughout the model applications. In the trench model, Ccofv is not sensitive. This is explained by the fact that all water is flowing in u-direction. In the Egmond model, Ccofu is less sensitive than Ccofv. This implies that the bulk of the flow of the water is directed in the v-direction. This is true for the Dutch coast, where the tidal wave progresses from south to north through the North Sea basin. In the basin model, both parameters are almost equally sensitive, implying that flows in both u- and v-direction influence the morphological developments.

SusW & BedW; The wave related suspended and bed load are scaled through SusW and BedW. Both the basin and trench models lack the forcing of waves, implying a relative sensitivity of zero. In the Egmond model however, waves are present. Waves have a larger influence on the bed load which is expressed in the larger sensitivity of BedW compared to SusW. Compared to the current related suspended and bed loads however, the sensitivity is relatively small. It thus becomes clear that currents play a larger role in the total sediment transport fluxes than waves.

RDC & RDW; RDC & RWD determine the current and wave related roughness k_c and k_w (refer equation 2-10). These roughness predictors can vary in space and time and influence the flow. Not only do they influence the flow, they also directly influence the suspended sediment transport through the reference sediment concentration (refer equations 2-31 & 2-33). These parameters are therefore sensitive throughout the three model applications. As no waves are present the basin and trench models, RWD does not play a role in these applications.

Rwave; With the right model set up, Rwave is used to calculate the wave related roughness k_w. As described above, this roughness influences both the flow and the sediment transport. This parameter should have an influence when waves are applied to a model. The settings of the Egmond model however, even after many efforts, were still such that Rwave was not taken into account in the calculations.

Morfac; The factor with which the sediment fluxes are multiplied after each time step is a very sensitive parameter throughout all model applications. Its relative sensitivity has a more or less comparable value in all three models.

AlfaBs & AlfaBn; These scaling factors can be considered non sensitive in all model applications. The transverse and longitudinal bed gradients, affecting the magnitude of the bed load transport, do not have a big impact on the morphological development in any of the models. This can partly be explained by the fact that the bed load also does not play a very significant role in all three model applications. The main flow in the Egmond model is from north to south, rather than east to west, which is the orientation of the profile model, implying that longitudinal slopes have a marginal effect. AlfaBn is a highly non-sensitive parameter in the trench model. Since this is a river and the model is only 0,5 [m] wide, there is no slope, nor bed load vector in the transverse direction.

Sus & Bed; These scaling factors for the current related suspended load and bed load are sensitive in all three model applications. The influence of Sus is much larger than the influence of Bed throughout the three models, implying that suspended sediment load dominates the bed load transport. The influence of the bed load transport is the largest in the trench model, implying that bed load does play a significant role in development of the trench over time.

SedDia & RhoSol; The sediment diameter and density of the sediment fractions considered both are very sensitive parameters throughout all three model applications. The total transport, both suspended and bed loads, depends heavily on the size and the weight of the sediment fractions considered. Put simply, large heavy grains are more difficult to move than small light grains. Although the relationship between transport, grain size and weight is somewhat more complicated, it is not difficult to imagine that these parameters have a large influence on the total transports.

This sub-section ends with a concluding remark on the Sand engine results. These results have not been discussed in the paragraphs above. The sand engine has been run for less than two days, with a Morfac of 15 to simulate a full month of bottom changes. The results however look very different from the rest of the model applications, with only two very sensitive parameters. The sensitivity runs have to be repeated, using a longer run, to validate these results, which are probably wrong.

5.2.3 IMPLICATIONS OF PARAMETER SENSITIVITY

The values of the absolute sensitivity of the different parameters provide valuable insights in the relevant processes for each test model application. By comparing these values with one another, conclusions can be drawn about the relative influence of the different processes. The relative importance of long shore versus cross shore processes, as well as the relative influence of bed load versus suspended load becomes clear by interpreting the sensitivity of the different parameters. Put simple, it becomes clear which processes are more important than others.

Hydrodynamic parameters can have a significant influence on the morphodynamic results. The hydrodynamic parameters are: Vicouv, Vicoww, Dicouv, Dicoww, Ccofu and Ccofv. From this list, Ccofv and Ccofu are the most sensitive, followed by Dicouv, which is a sensitive parameter, but only in the Egmond model. The other three parameters have a negligible effect on the morphodynamic development within the model applications.

5.2.4 PARAMETER INTER-RELATIONSHIPS

In the sensitivity analysis of section 4.1, four different parameters (Sus, Bed, SedDia & AlfaBn) have been tested in distinct parameter pairs. This approach enables the investigation of inter-relationships amongst these parameters pairs. When two inter-related parameters are calibrated, it is unlikely that the algorithm will find the absolute minimum of the cost function. Chances are that the algorithm will end up in a local minimum. The parameter values corresponding to this local minimum are not the optimum parameter settings, but are presented as such by the calibration instrument. It is therefore important to gain insights in the inter-relationships between parameters to prevent calibration on inter-related parameters.

From this analysis, the following relationships were found in all model applications, of which an explanation is provided subsequently:

- The influence of Bed increases, as SedDia increases
- The influence of Sus increases, as SedDia decreases
- The influence of Bed increases, as AlfaBn decreases

A smaller sediment diameter implies smaller and lighter sand grains. As the gains become smaller and lighter, they are more easily thrown into suspension by the turbulent forces. The relative importance of the suspended load will therefore increase when the sediment diameter decreases. Vice versa, when the sediment diameter increases, less sand will be thrown into suspension. This increases the relative importance of the bed load.

This leaves the explanation of the inter-relationship between AlfaBn and Bed. AlfaBn is a tuning parameter, with which the relative influence of the bed slopes on the bed load can be determined (refer equation 2-14). AlfaBn directly influences the magnitude of the transverse bed load vector and therefore directly influences the contribution of the bed load transport to the total transport.

5.3 REVIEW OF CALIBRATION CASES

In the first calibration case it has been investigated whether the calibration instrument can pinpoint non-sensitive parameters automatically. To this purpose, three calibration runs have been completed in which both sensitive and non-sensitive parameters were included. Unfortunately, both the sensitive and the non-sensitive parameters values were altered by the instrument, which makes it impossible to determine afterwards which parameters were sensitive and which ones were not. If the instrument would have only altered the sensitive parameters values, it would not have been necessary to perform a sensitivity analysis before calibration, as only sensitive parameters would change value.

The first calibration case also showed that different parameter values produce different bottom profiles. These different profiles however can lead to comparable values of the cost function. This raises the question whether better techniques can be applied to determine the fit of a bottom profile with a certain set of measurements. The applied least squares cost function might not be accurate enough to evaluate the simulation results with, which especially holds for more complicated 3D bottom profiles. For relatively simple 2D bottom profiles, the method looks to be accurate enough, especially when only parts of the grid are taken into account in the calculations. The more grid cells are taken into account, the more difficult it becomes to tell which part of the grid is improving when the cost function decreases in value.

The second calibration case was introduced to find out whether optimum parameter values differ for different model applications and transport formulations. It was found that, depending on the model application and transport formula used, optimum parameter values are different. This implies that processes underlying the morphodynamic developments are not well enough represented by the Delft3D model. A calibration procedure does therefore not necessarily improve the physics underlying the Delft3D model, but fits the model to the data at hand. This does not mean that calibration is useless, but it does imply that the calibration results have to be interpreted in the right manner. There is no such thing as a universal best suited calibration strategy.

In the third calibration case, the Sand engine model has been calibrated. In order to be able to implement the calibration instrument on this complex morphodynamic model application, the runtime of the model had to be drastically reduced. Therefore, only part of a full month run was simulated, in which the storms of that month took place, in the hope that all morphodynamic developments would take place during those storms. The calibration however did not improve the model significantly. The cost function was reduced by less than one percent. Probably, the simplifications in the calibration and model setup are to blame for the disappointing results.

In order to improve the calibration results of complex morphodynamic model applications such as the Sand engine, a different approach has to be developed. The runtime of the Sand engine model posed the biggest problem; it takes more than 7 days to complete a single run. This runtime has to be reduced in order for the calibration instrument to be of use. Parallel computing could decrease the runtimes. Furthermore, the first step of the dud method requires multiple model runs in different directions in order for dud to determine in which direction to continue searching. These initial model runs can be done parallel. The steps after that however are more sequential, implying that these steps have to be performed one after another. Smart simplifications of the Sand engine model can also decrease the runtimes. A coarser grid, less forcings or a larger time step are all ways to decrease the runtime. Smart grid selections could also improve the results. Although not all morphodynamic developments took place in the simulated five days, some parts of the grid did not change much anymore in the remaining days of the simulation. Zooming in on those areas and using only these results partly solves this problem. If measurements were taken more often, before and after a storm for instance, only these days would have to be simulated. Concluding, more research is needed on how to reduce the runtimes.

5.4 REVIEW OF PERFORMANCE PARAMETERS

The calibration cases have shown that the least squares cost function which has been used to determine the quality of the calibration iterations might not be the ideal tool for the job. Therefore, two additional performance parameters have been implemented in the calibration instrument; the decomposed Brier skill score and the Kirchhofer method.

5.4.1 REVIEW OF DECOMPOSED BRIER SKILL SCORE

Mathematically, there is no difference between minimizing the least squares cost function and optimizing the Brier skill score. In both cases, the difference between the model results and the measurements has to be minimized. The Brier skill score therefore is not suited to replace the least squares cost function. However, the decomposed values of the Brier skill score do provide insights in the error that is found in the models results. Furthermore, the Brier skill score has proven to be useful when applied to determine the sensitivity of parameter pairs.

The tests of section 3.2.2 have shown that the decomposed values of the BSS are suited to determine which error in the simulation is contributing most to the BSS; the phase, amplitude or mean error. The values however are not accurate enough to be translated into an absolute normalized length scale.

5.4.2 REVIEW OF KIRCHHOFER METHOD

The Kirchhofer sub grid row and column scores provide insights in the performance of the separate rows and column of the numerical grid. In other words, it becomes clear which part of the grid is performing well. The shapes of the summed row and column scores match the shape of the overall Kirchhofer score. However, The Kirchhofer method does not distinguish between errors in the shape and the location of the bottom profile. Furthermore, the Kirchhofer column and row scores are only applicable to two dimensional grids containing multiple grid cells in both M- and N-direction.

If Kirchhofer method were to be implemented as a cost function, there is a problem with the row and column scores. It could be possible that during a simulation, some of the row and column scores are improved, whilst others deteriorate. It is than difficult to determine whether the simulation improved with respect to the previous simulation. It is therefore not straightforward to implement this method as a cost function.

The Kirchhofer grid scores follow the exact same pattern as the value of the cost function; the derivatives of the function always have the same sign. This implies that it is not useful to develop a separate cost function based on this method, as this probably would not produce different parameter values in the optimization process.

6 CONCLUSIONS

An efficient method for calibration and sensitivity analysis has been developed; OpenDA MOR. The instrument has proven to be a powerful tool for both calibration and sensitivity analysis.

TWIN experiments have shown that it is possible to calibrate morphodynamic model parameters using morphodynamic model results and -measurements. From these experiments, rules of thumb have been derived on how to apply the instrument. Calibrating on multiple interdependent parameters should be avoided, as this tremendously increases the difficulty of minimizing the cost function. The calibration parameters must be sensitive and when more than one parameter is calibrated, these parameters should be non-interrelated. However, no more than 2 parameters should be calibrated at once. The initial values of the parameters should be within a 50% - 75% range of their optimum values. Lastly, applying a lognormal transformation to the parameter values will prohibit unrealistic negative values. These rules of thumb minimize the chance of the algorithm finishing in a local minimum as well as reduce the amount of runs needed in the calibration.

Using the calibration instrument, processes have been identified that can be represented more accurately in the Delft3D model. The calibration cases have shown that optimum parameter values of one parameter can vary, depending on the model application and transport formula used. It is therefore not likely that the actual physics behind the model are better represented after a calibration. The fact that different transport formulations are available itself already shows that different interpretations of the physics behind the morphodynamic processes exist. The outcome of the calibration runs confirm that the physics can be interpreted in many ways, leading to different optimum values of the calibrated parameters. Although calibration might not improve the underlying physics, this does not imply that that a calibration instrument is useless. It does imply that it has to be applied with care, knowing that the instrument will fit the model to the data at hand; there is no such thing as a universal best suited calibration strategy!

The sensitivity analysis and the TWIN experiments have shown that there is a big overlap in parameter sensitivity throughout the different model applications. SedDia, Morfac, RhoSol, Ccofv, Ccofu, Sus and RDC turned out sensitive in all model applications. Furthermore it was shown that the transport formulation is a sensitive parameter as well, which should be chosen with care, based on the governing processes in the model application. The absolute and relative sensitivities of the parameters do differ somewhat, depending on the model application; some parameters are more sensitive than others. From the parameter sensitivity it also becomes clear which processes are important in every model and which inter-relationships exist in between the different parameters. It is not possible for the instrument to pinpoint sensitive parameters automatically, without performing a thorough sensitivity analysis upfront.

The uncertainties of simulation results can more accurately be mapped using the information on the sensitivity of all the parameters. When it is known which parameters are uncertain and between which physical bounds these parameters would normally vary, only few additional runs are needed to gain insights in the bounds of the models results.

Hydrodynamic parameters can have a significant influence on the morphodynamic results. Ccofv, Ccofu and Dicouv all exerted a large effect on the development of the bottom profile over time. Adjusting the Eddy viscosities however did not affect the simulation results significantly.

The least squares cost function has proven to be applicable in the calibration of morphodynamic model applications. However, from this cost function it does not become clear what part of the grid is functioning well, nor what type of error is contributing most to the total deviation of the simulated bottom profile. Therefore, two additional performance parameters have been introduced; the Brier skill score and the Kirchhofer method. None of the two methods however proved to be a good substitution for the least squares cost function. The decomposed values of the BSS do provide insight in the type of error found in the model results.

Concluding, an effective calibration instrument for morphodynamic Delft3D model applications has been developed. Although its practical application to complex model applications proved to be difficult from a practical point of view, it has been shown that the methods of calibration and sensitivity analysis are applicable on all model applications.

7 RECOMMENDATIONS

Additional research is required on how to decrease the runtimes of complex morphodynamic model applications. Smart simplifications of these applications are a necessity to make the calibration tool practically applicable on these models.

A different, more accurate cost function could improve the morphodynamic calibrations. Such a cost function must be able to distinguish between phase and amplitude errors. It could be very interesting to include the work on this topic of Judith Bosboom from the TU|Delft in the development of such a cost function.

OpenDA does not offer the option of implementing hard parameter constraints. It is recommended that such functionality is added to the software. Although a lognormal transformation already prevents parameters from taking negative values, it does not prevent the parameters from adopting unrealistically high values.

Weights should be added to the Brier Skill Score calculations, which take into account the sizes of the different grid cells in the calculations of the decomposed values. The larger the grid cell is, the larger the weight it will receive in the calculations. The OpenEarth Brier skill score Matlab function can be used as an example in which weights have already been included.

The single TWIN experiments on the Sand engine model should be repeated using another simulation period, in which more morphological changes take place. These runs have to validate the sensitivity results found in this Thesis.

Additional runs should be performed with the Egmond model to investigate the sensitivity of Rwave. Although forced by waves, this parameter did not have an influence on the simulations results. With the right model setup, in which Rwave is used to calculate the wave related roughness k_w, it should have an influence on the simulations results.

8 BIBLIOGRAPHY

Blair, D. (1998). The Kirchhofer technique of synoptic typing revisited. *International journal of climatology*, 1625-1635.

Deltares. (2010). *Delft3D FLOW User Manual.* Delft: Deltares.

Gautier, C. (2010). *SWAN calibration and validation for HBC2011*. Delft: Deltares.

Giardino, A., Briére, C. Werf van der, J. (2011, February). *Morphological modelling of bar dynamics with Delft3D.* Delft: Deltares.

Hummel, S., Verlaan, M., Sarafy, el, G., Velzen, van, N. (2011, July). Using OpenDA's Black Box Wrapper. Deltares.

Man, W. (2012). *Short term changes in the zandmotor morphology*. Twente: University of Twente.

Murphy, A., Epstein, E. (1988). Skill scores and correlation coefficients in model verification. *Monthly weather review*, 572-581.

OpenDA. (sd). *OpenDA*. Opgeroepen op May 10, 2012, van OpenDA: http://www.openda.org

Ralston, M.L., Jennrich, R.I. (1978). Dud, a derivative-free algorithm for nonlinear least squares. *Technometrics*, 7-14.

Rijn, v. L. (2007). Unified view of sediment transport by currents and Waves. I; Initiation of Motion, Bed Roughness, and Bed-Load Transport. *Journal of hydraulic engineering*, 649 - 667.

Rijn, van, L.,Walstra, D. (2003). *Modelling of sand transport in Delft3D Online.* Delft: WL | delft hydraulics.

Sutherland, J., Peet, A., Soulsby, R. (2004). Evaluating the performance of morphological models. *Coastal Engineering*, 917-939.

Verlaan, M. (sd). Calibration in OpenDA. Deltares.

Wenneker, I.,Gautier, C.,Gerritsen, H. (2009). *User manual calibration instrument SWAN*. Delft: Deltares.

Yarnal, B. (1984). A procedure for the classification of synoptic weather maps from gridded atmospheric pressure surface data. *Computers & Geosciences*, 397 - 410.

APPENDIX A RESULTS SENSITIVITY ANALYSIS

A.1 EGMOND MODEL

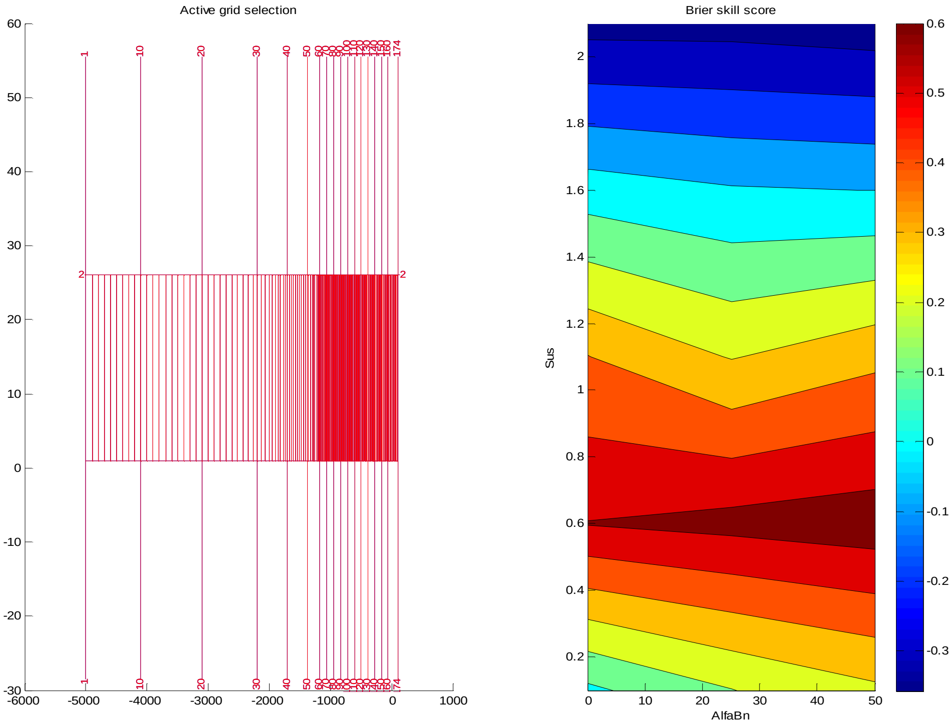


Figure 8-1 | run 1 (Sus vs. AlfaBn) sensitivity Egmond model

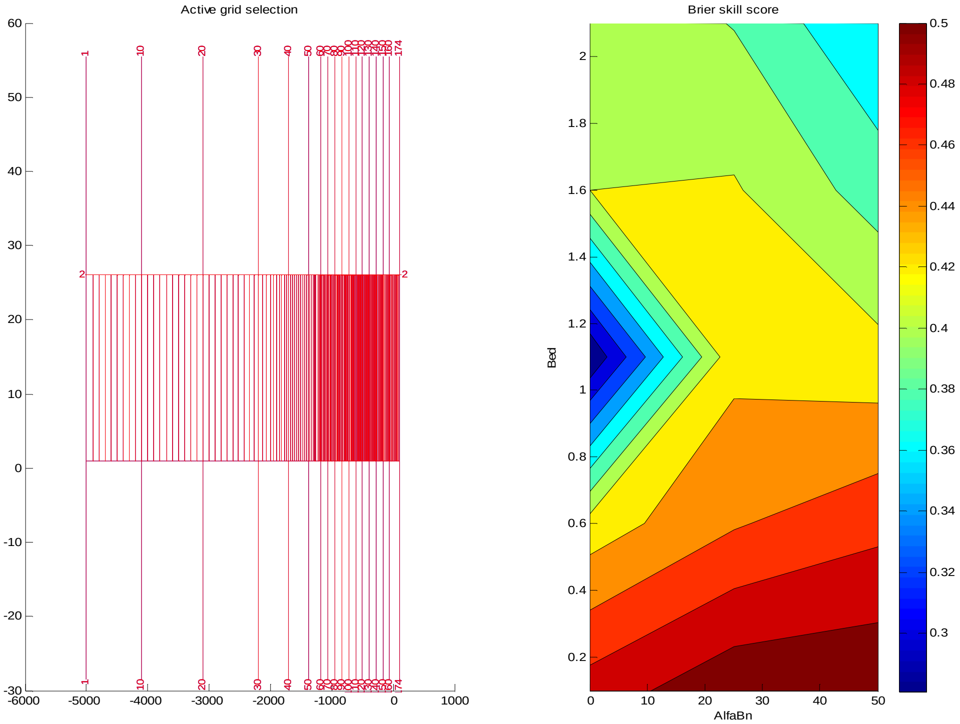


Figure 8-2 | run 2 (Bed vs. AlfaBn) sensitivity Egmond model

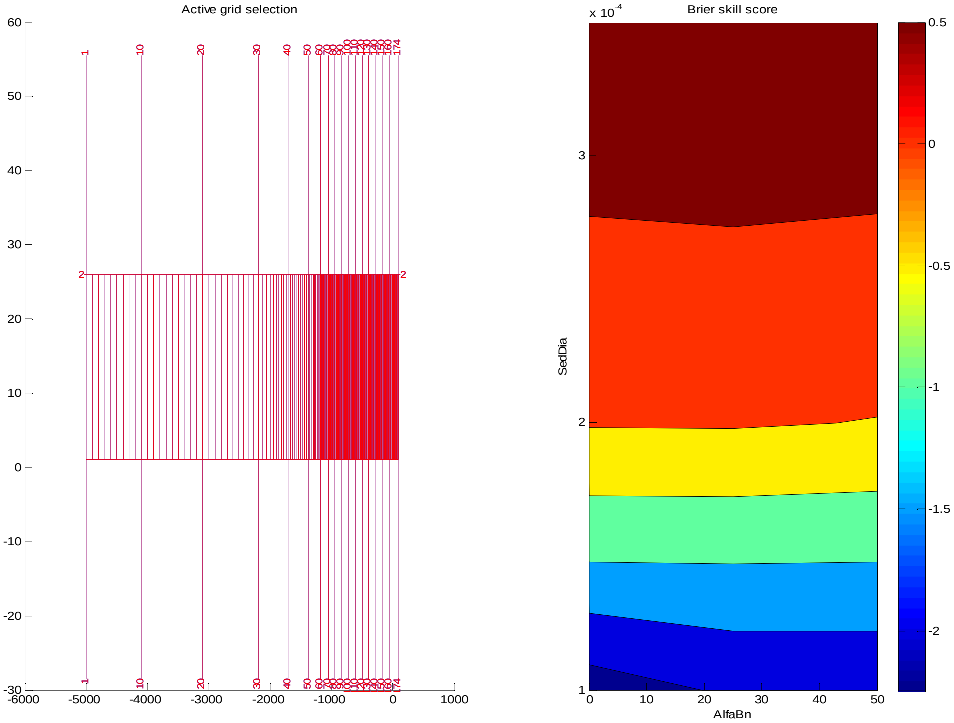


Figure 8-3 | run 3 (SedDia vs. AlfaBn) sensitivity Egmond model

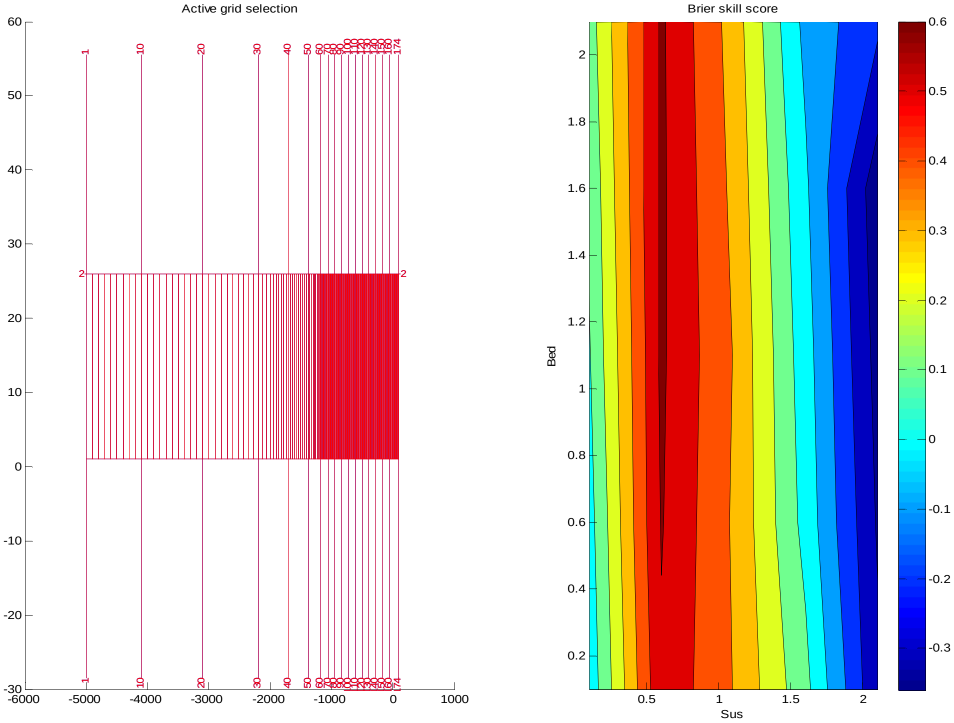


Figure 8-4 | run 4 (Bed vs. Sus) sensitivity Egmond model

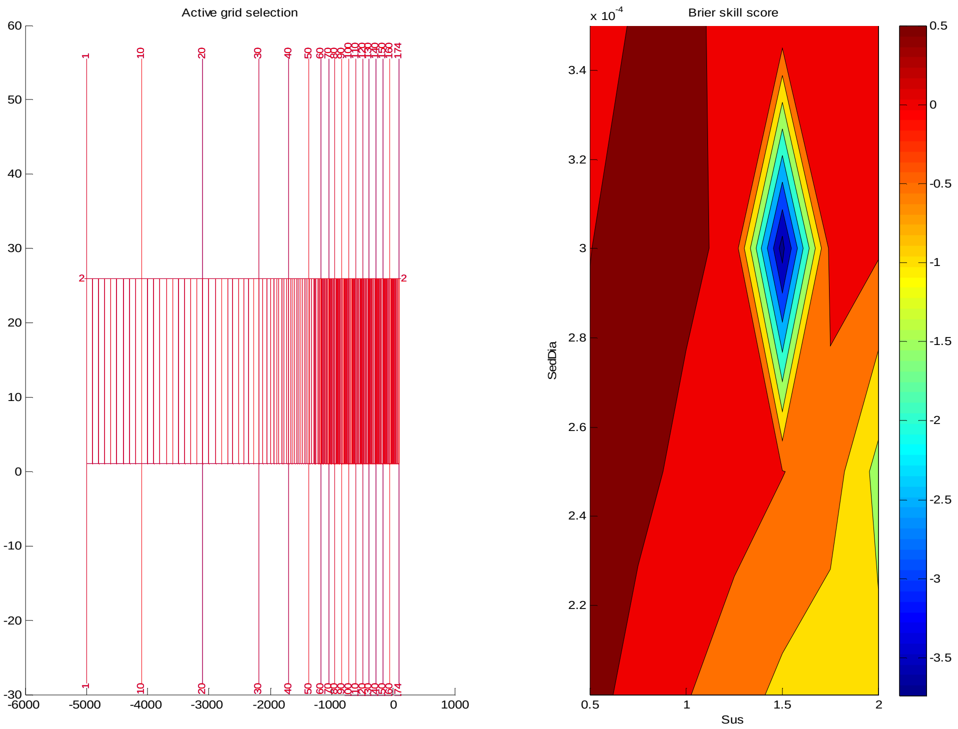


Figure 8-5 | run 5 (SedDia vs. Sus) sensitivity Egmond model

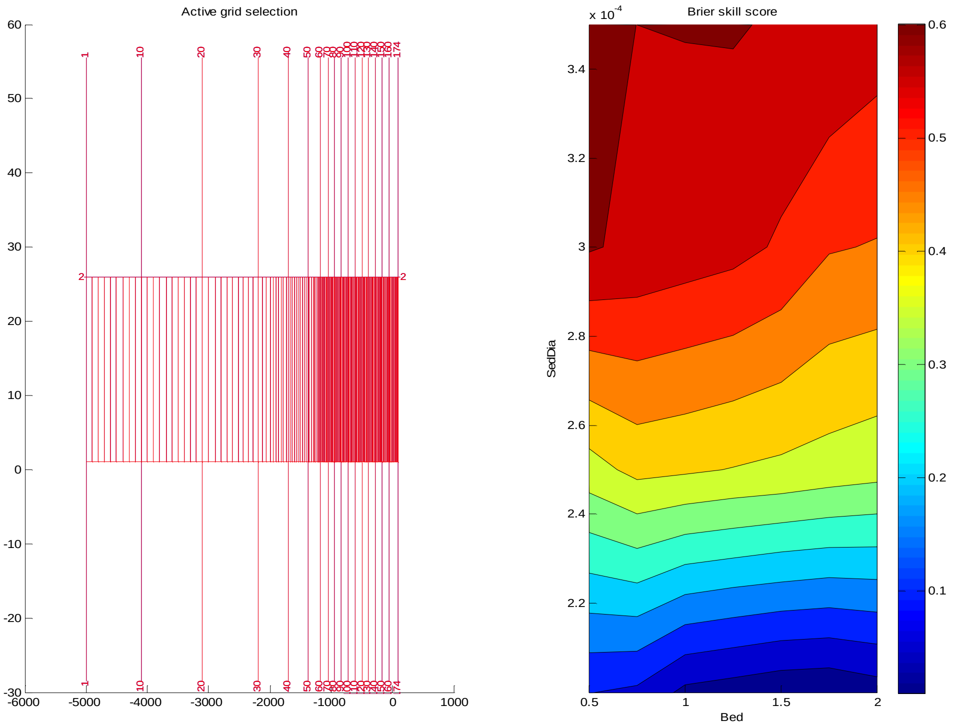


Figure 8-6 | run 6 (SedDia vs. Bed) sensitivity Egmond model

A.2 TRENCH MODEL

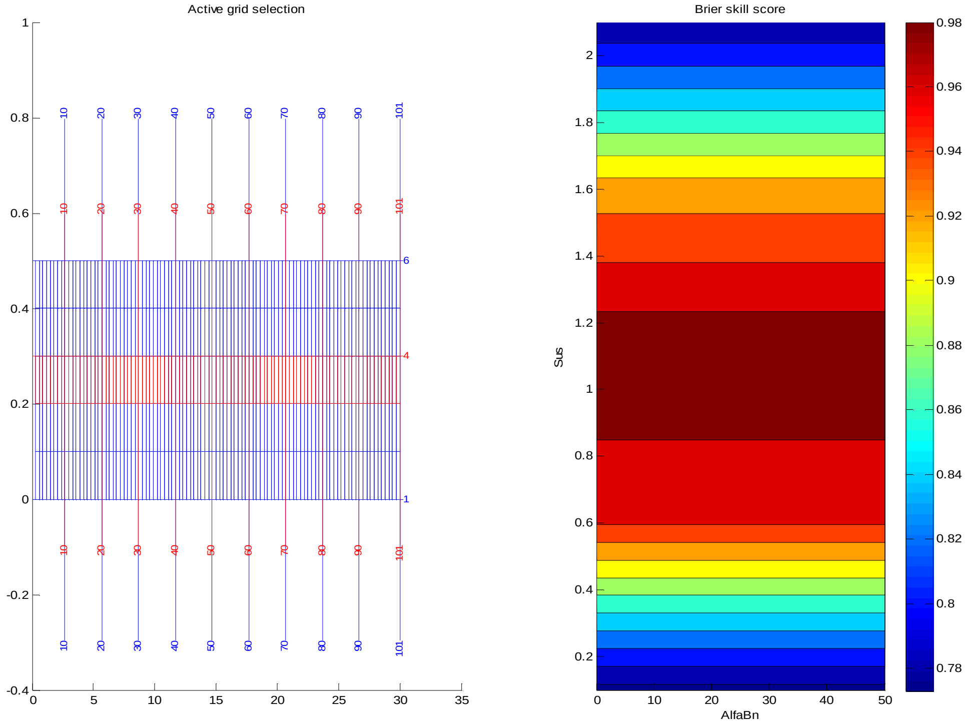


Figure 8-7 | run 1 (Sus vs. AlfaBn) sensitivity trench model

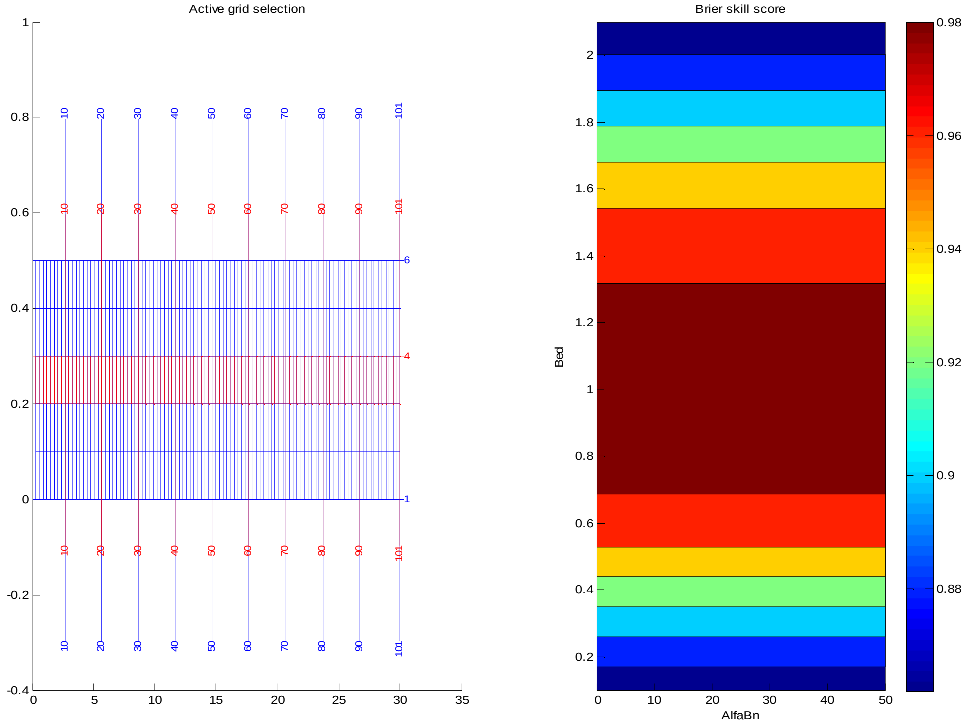


Figure 8-8 | run 2 (Bed vs. AlfaBn) sensitivity trench model

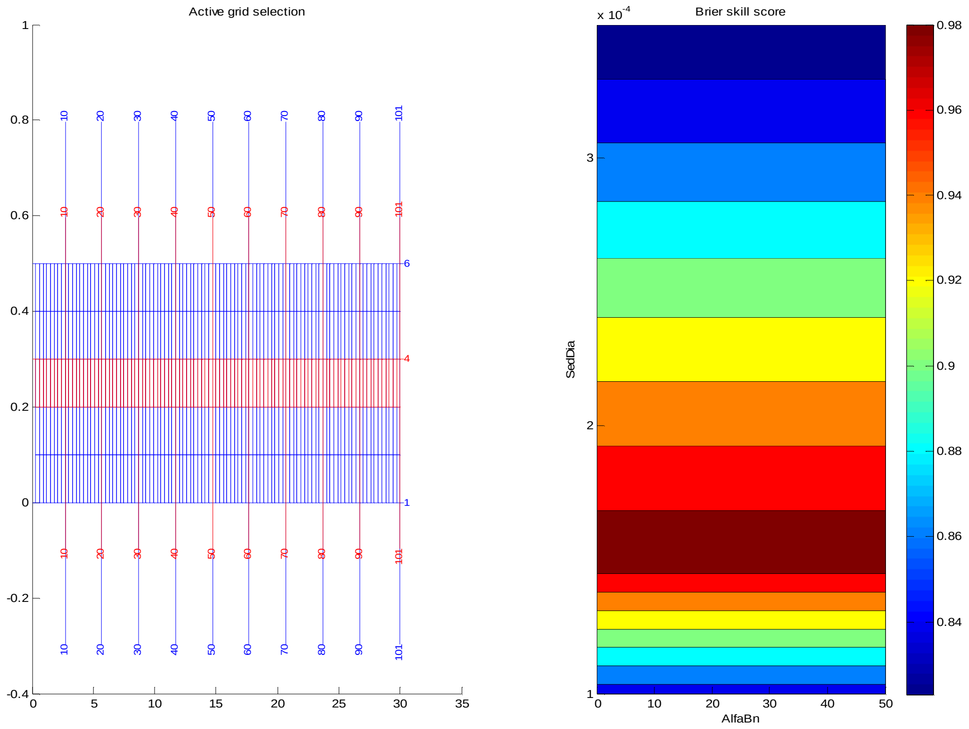


Figure 8-9 | run 3 (SedDia vs. AlfaBn) sensitivity trench model

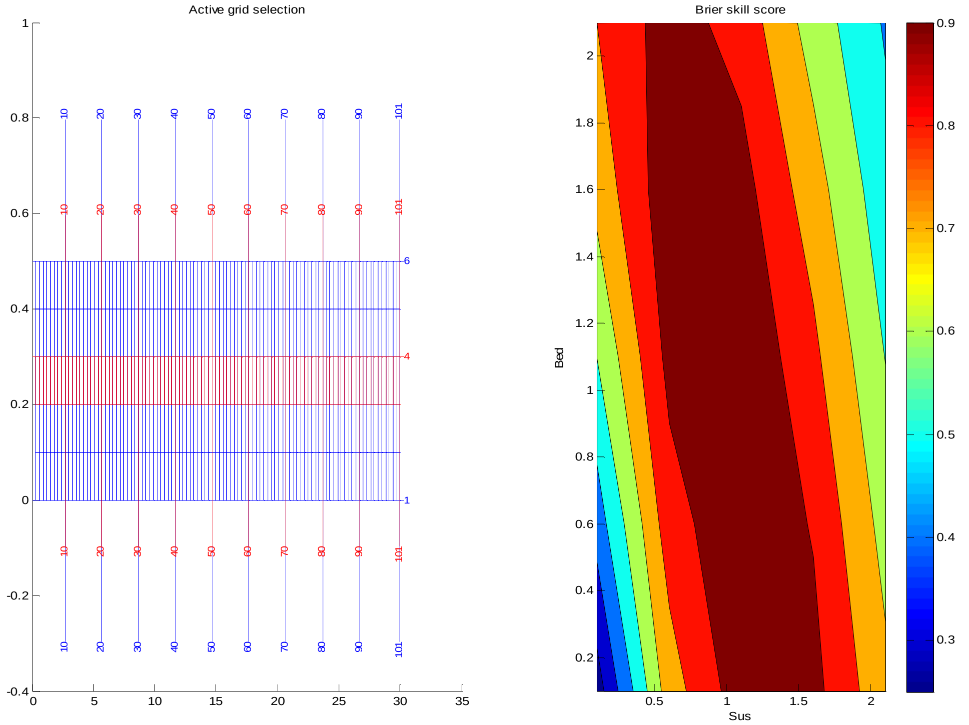


Figure 8-10 | run 4 (Bed vs. Sus) sensitivity trench model

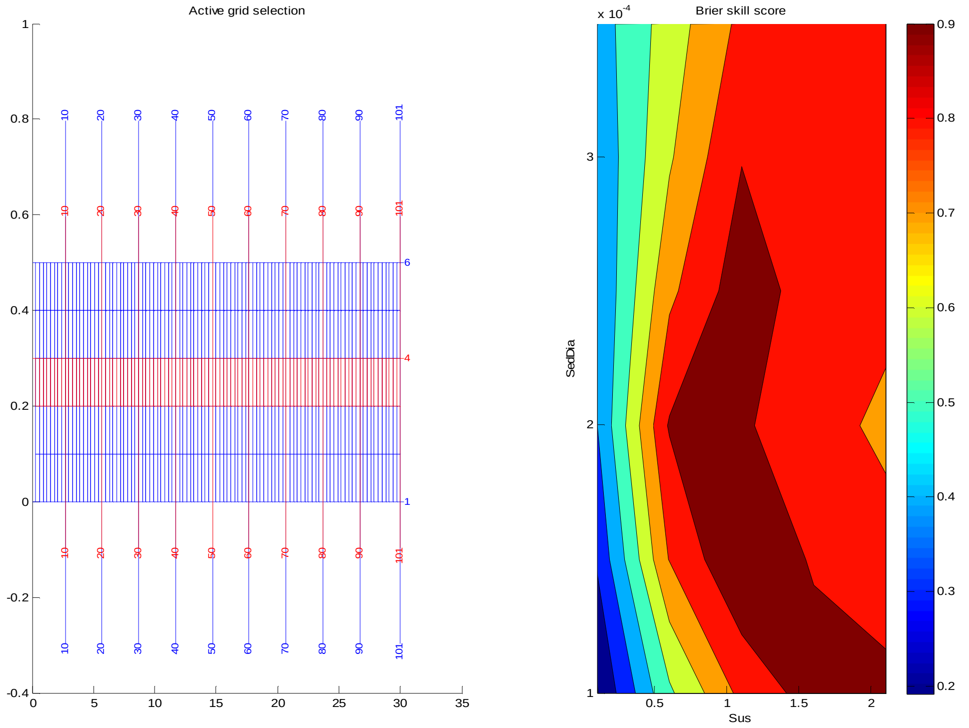


Figure 8-11 | run 5 (SedDia vs. Sus) sensitivity Egmond model

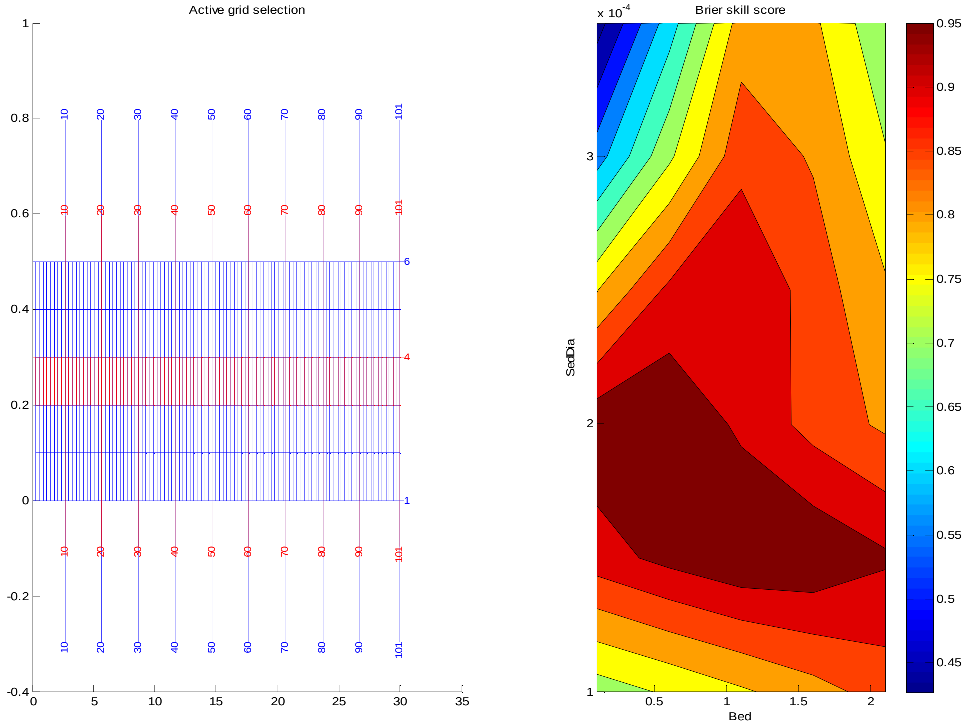


Figure 8-12 | run 6 (SedDia vs. Bed) sensitivity trench model

A.3 BASIN MODEL

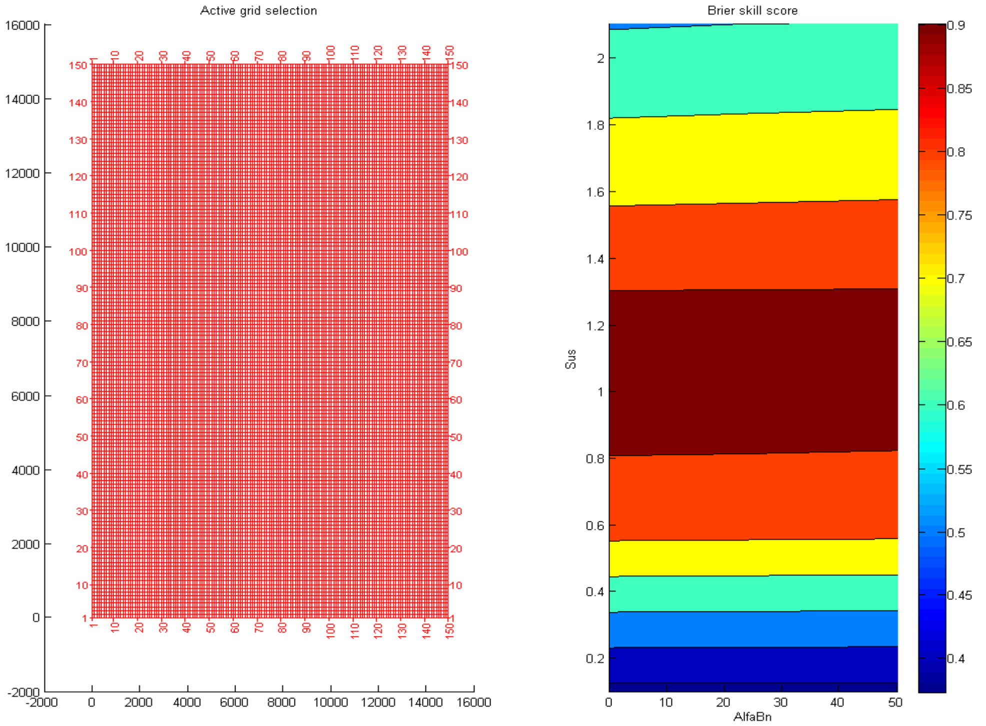


Figure 8-13 | run 1 (Sus vs. AlfaBn) sensitivity basin model

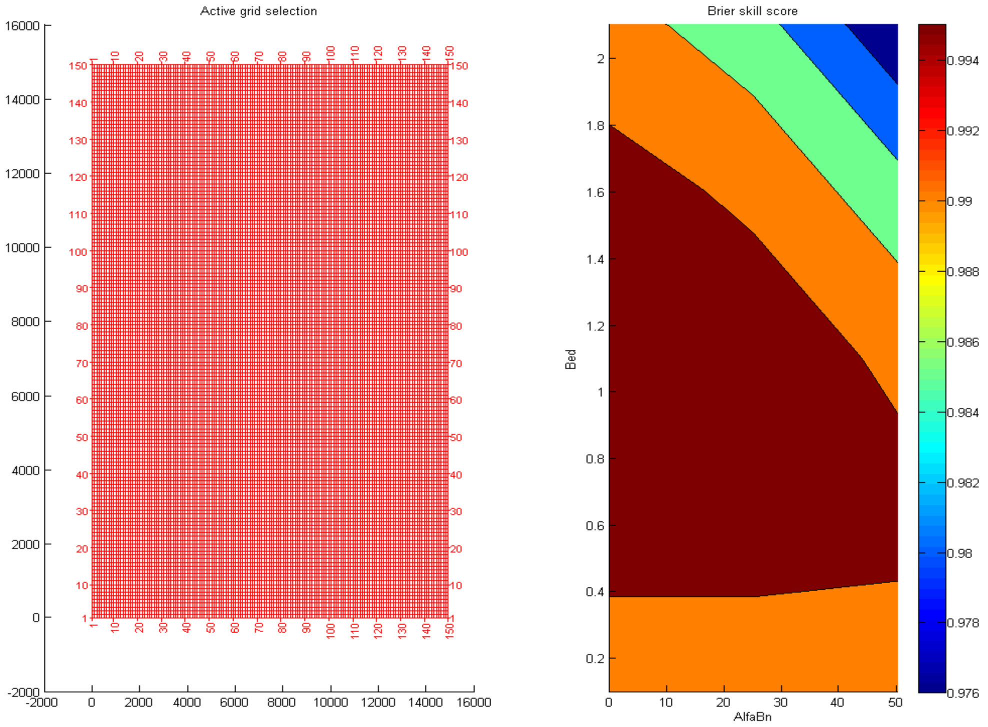


Figure 8-14 | run 2 (Bed vs. AlfaBn) sensitivity basin model

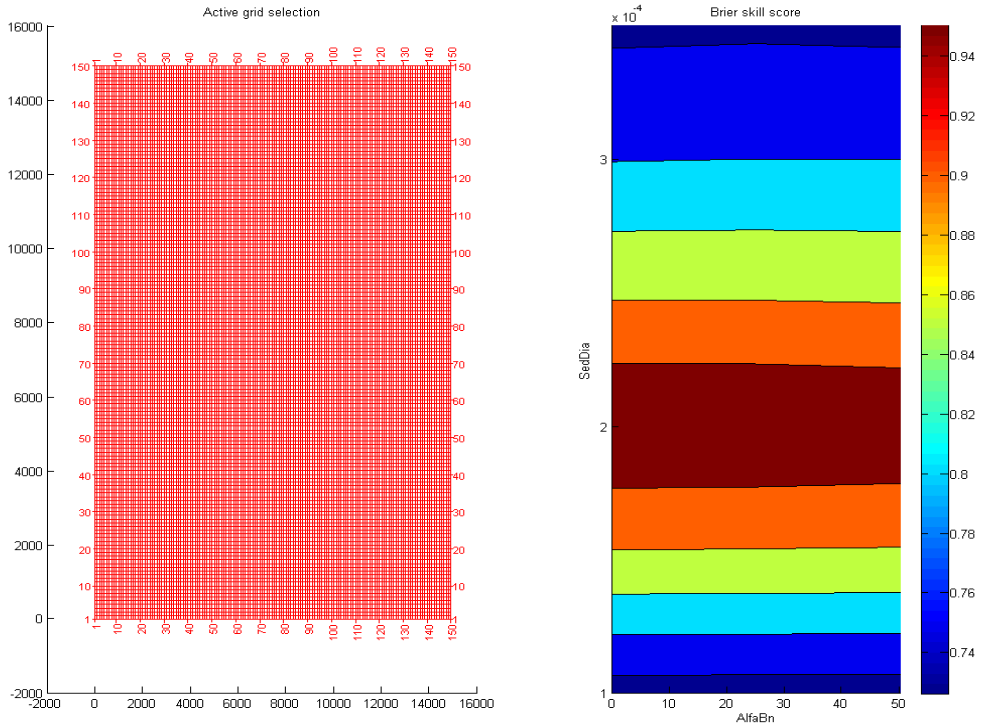


Figure 8-15 | run 3 (SedDia vs. AlfaBn) sensitivity basin model

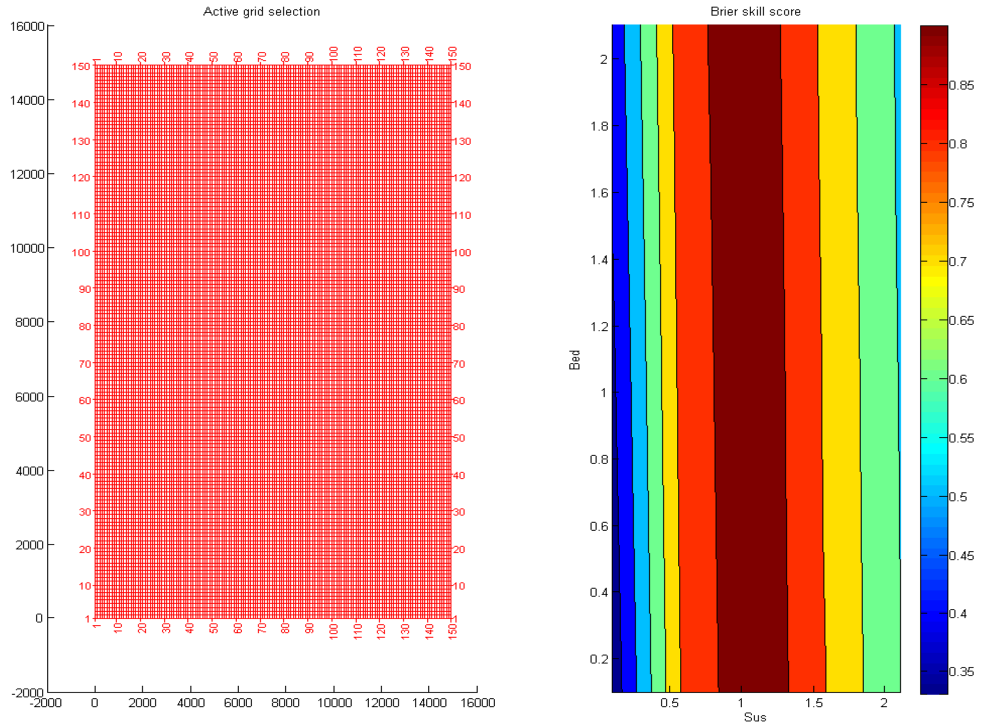


Figure 8-16 | run 4 (Bed vs. Sus) sensitivity basin model

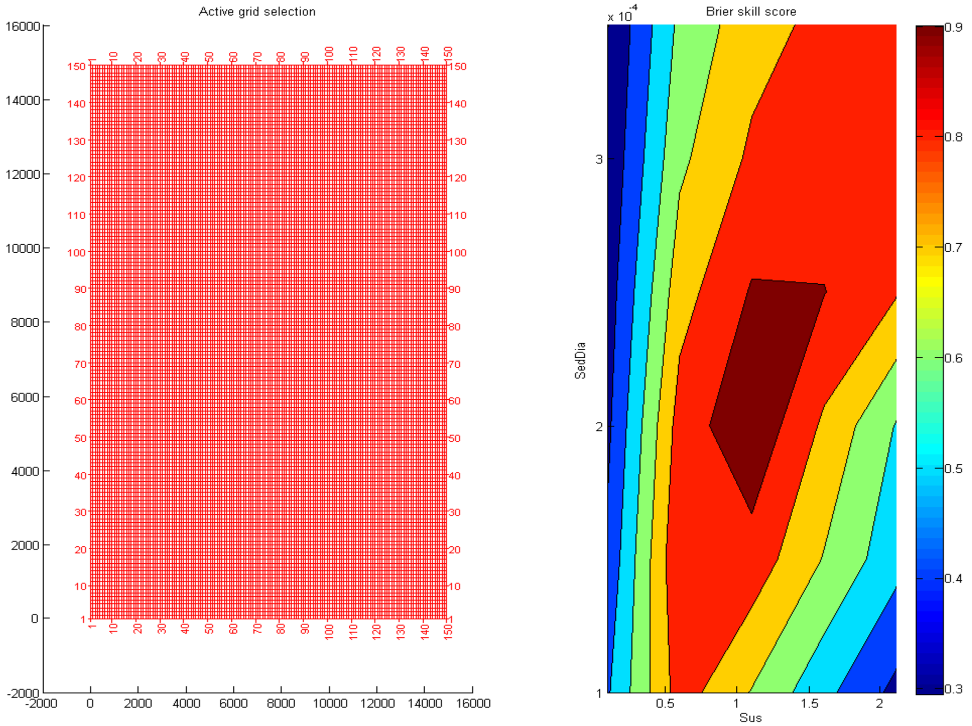


Figure 8-17 | run 5 (SedDia vs. Sus) sensitivity basin model

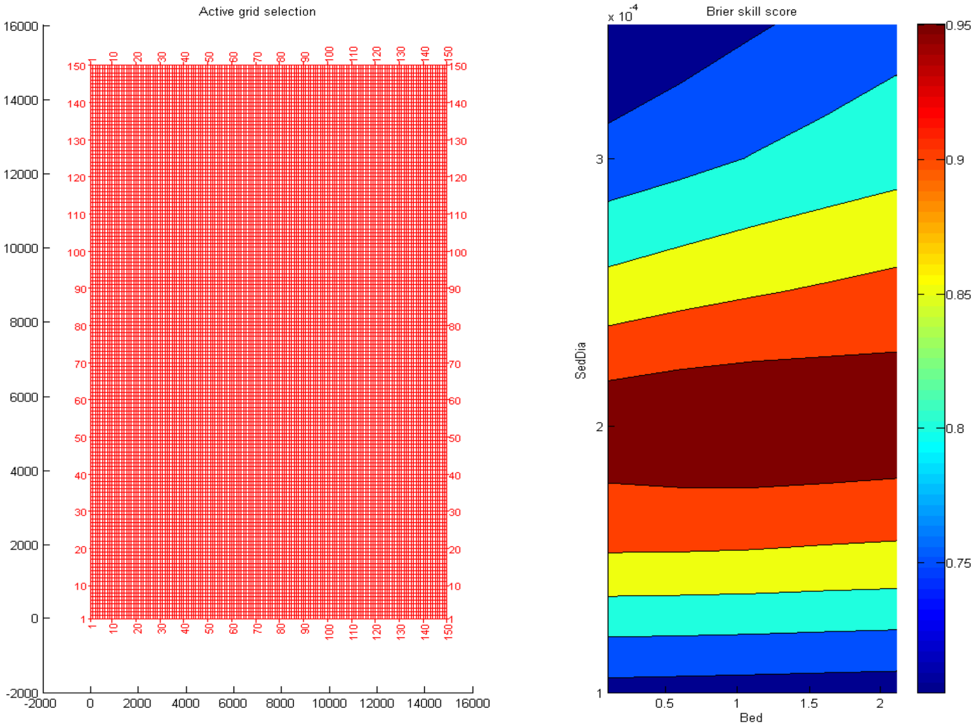


Figure 8-18 | run 6 (SedDia vs. Bed) sensitivity basin model