Conference Paper, Published Version

Vanlede, J.; Chu, Kai; Fonias, E.

Introducing the VIMM toolbox for the (re)calibration of the SCALDIS model

Introducing the VIMM toolbox for the (re)calibration of the SCALDIS model

J. Vanlede
Flanders Hydraulics Research (FHR)
Antwerp, Belgium
joris.vanlede@mow.vlaanderen.be

K. Chu; E. Fonias
International Marine and Dredging Consultants (IMDC)
Antwerp, Belgium

***Abstract*— The SCALDIS model is the most recent model schematisation of the Scheldt estuary, including the Belgian Coastal Zone, the Western Scheldt (Nl.), the Sea Scheldt (Be.) and its tributaries. SCALDIS is implemented using TELEMAC-3D, and is being recalibrated in an ongoing project. For a reference model with a broad possible set of applications, it is important to have a calibration strategy that is sufficiently broad in scope. In order to achieve this goal, the VIMM toolbox is used for the hydraulic model. It is in-house developed at Flanders Hydraulics Research (FHR) and runs in MATLAB. It has a data model to store model results and measurements. This enables the statistical core of the code to be agnostic of model type (currently, Mike11, Delft3D, Delft-FM, Simona and TELEMAC are supported) and to be flexible with regards to measurement data sources.**

I. INTRODUCTION

The SCALDIS model [1] is a reference model on which many different applications are based. It provides the hydrodynamic input for sediment transport calculations (sand and mud). The hydrodynamic results are also used to delineate habitats in the subtidal zone of the Sea Scheldt. In particular, the maximum flood current velocity defines robust boundary values for high and low dynamic subtidal zones. This is based on an extensive validation with monitored spatiotemporal variability of benthic fauna [2]. Furthermore, the SCALDIS model is used to evaluate the effect of flood control areas on the hydrodynamics in the Scheldt estuary. Tracer dispersion experiments in the model are used to calibrate an ecosystem model. The flow fields it produces are also processed as a flow atlas, and are implemented in the shipping simulator of Flanders Hydraulics Research (FHR).

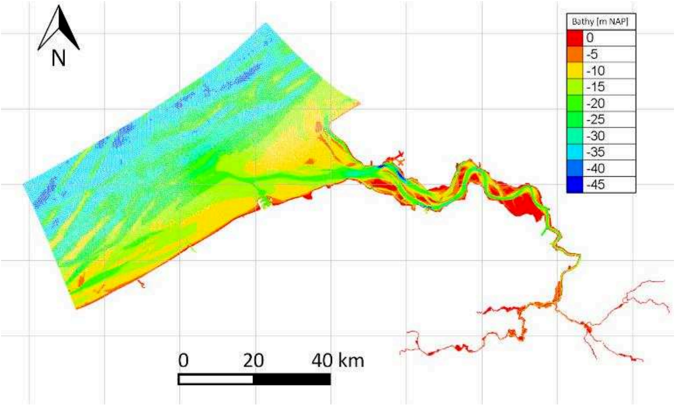

Figure 1 - Mesh and bathymetry (m NAP) of the SCALDIS 2019 model

II. THE SCALDIS MODEL

A. Model mesh and bathymetry

The model bathymetry (shown in Figure 1) is defined in the vertical datum “Normaal Amsterdams Peil” (NAP).

To fulfil the requirement of the nautical shipping simulator at FHR, the mesh is locally refined in zones of interest (e.g. access channels to locks). The total number of computational nodes is 478,290 with 915,622 triangular elements.

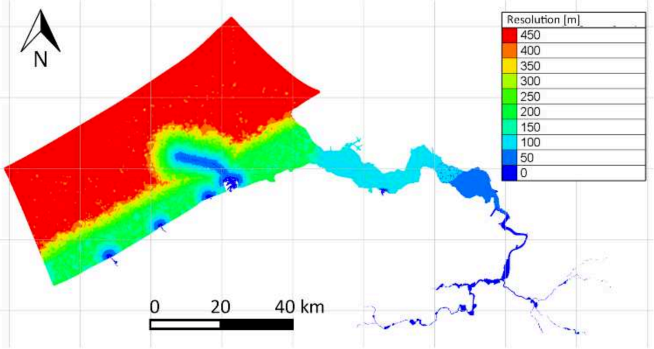

Figure 2 - Mesh resolution (edge length expressed in meters)

The SCALDIS model runs in 3D mode with 5 vertical planes at (generalised) sigma coordinates (bottom to top) 0, 0.12, 0.3, 0.6 and 1.0.

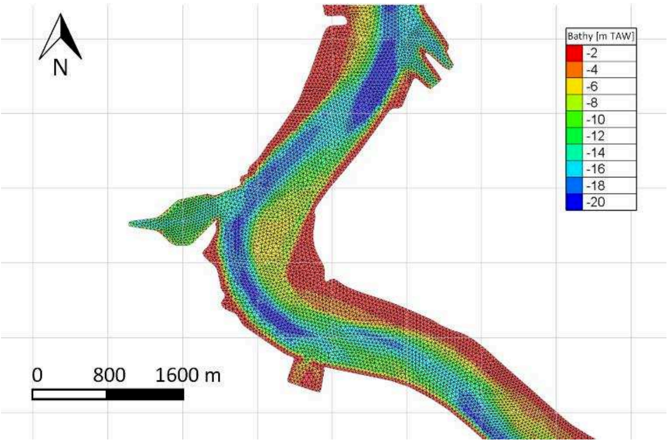

Figure 3 – Zoom of the grid, centered on the access channel to the Kallo lock

B. *Update cycle*

Every 6 years, a new bathymetry becomes available that covers the Belgian part of the Scheldt estuary and its tributaries. Some zones are surveyed more frequently. The model update cycle follows this surveying cycle.

The SCALDIS 3D hydrodynamic model is maintained in an update cycle where every year, there is an *actualisation* of the model with any new bathymetric information that has become available in the preceding year. In an actualisation step, grid and model parametrisation (e.g. roughness) are not changed.

The model is subsequently verified against available measurements of that year, to keep track of the impact of the actualisation on the model skill.

Every 6 years the model is re-calibrated. The original model was calibrated against data of 2013 [3], so the re-calibration (currently ongoing) focuses on the year 2019. During re-calibration both the model grid and model parametrisation can be changed. After calibration, the model will also be validated against data from 2019 that were not used in calibration.

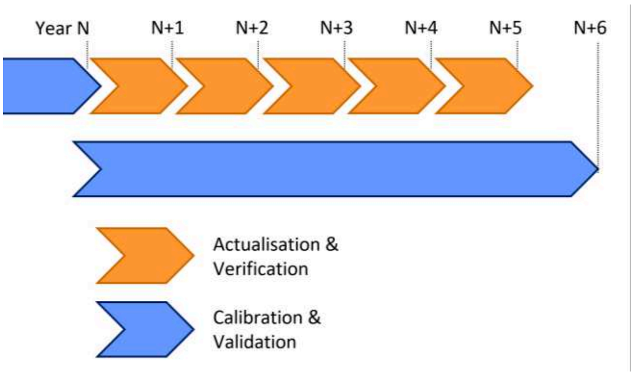

Figure 4 – The update cycle for the SCALDIS model

III. VIMM Toolbox

VIMM stands for "Visualisation of Model and Measurements", and is a toolbox developed at FHR in MATLAB for the purpose of assisting the modeller in the tasks of comparing models to measurements (e.g. during model calibration) and inter-model comparison (e.g. during scenario analysis).

A. *Why?*

Comparing models to measurements, or models with each other is a non-trivial task that can require a lot of scripting to compare data in selected stations over a specified time interval. A modeller often faces multiple measurement data sources (each with their own data format), and/or different model output formats from different modelling suites.

Our aim when developing VIMM was to develop a generic codebase that can be deployed rapidly in any project, regardless of which data sources or which modelling suites are used. This improves efficiency, and ensures a high standard in code quality.

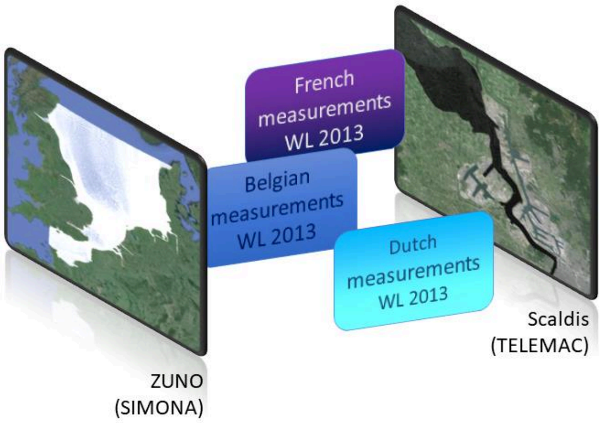
French measurements WL 2013
Belgian measurements WL 2013
Dutch measurements WL 2013
Scaldis (TELEMAC)
ZUNO (SIMONA)

Figure 5 - Typical usecase where VIMM is useful: two models of different modelling suites (one structured, one unstructured) need to be compared against a dataset of water levels coming from different data sources

B. *VIMM as a knowledge platform*

The typical use case for VIMM at FHR is model calibration and validation in a data rich environment [3] [4] [5]. The year-long use of the toolbox has de-facto standardised the model skill assessment of hydraulic models at FHR. It has helped in creating a common language to communicate on model skill, which in turn helps in the internal training of new modellers.

Because it is a shared codebase that is maintained in a central repository, any new developments are immediately available for all modelling teams. This way, knowledge on assessing model skill and calibration strategies can easily be captured and shared between researchers.

C. *Data source abstraction layer (DSAL)*

A straightforward way to ensure that the code can handle different data sources (both file-based and based on webservices) is to design a data model to store measurement data. This introduces a data source abstraction layer (DSAL) in the typical workflow.

The DSAL is illustrated in Figure 6 for the case that the modeller uses water level (WL) data from different data sources. Below the DSAL in the schematised dataflow, VIMM becomes agnostic of the data source.

This requires that the VIMM data structure is self-describing (so containing both data and metadata) and feature-complete. Secondly, a set of scripts needs to be developed (typically one for every data source) to convert the data from any proprietary format to the VIMM data structure.

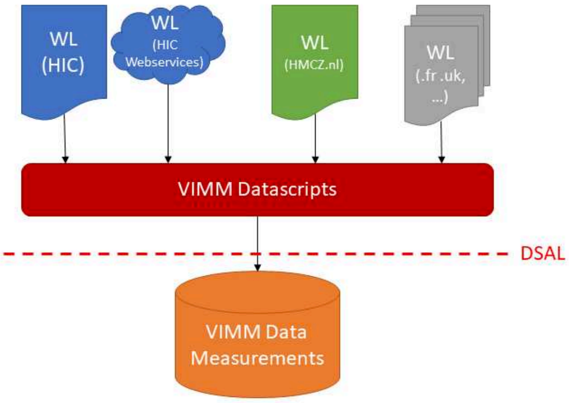

Figure 6 - The data source abstraction layer (DSAL) as the red dashed line in a VIMM workflow with different data sources of water level measurements

D. *Model abstraction layer (MAL)*

In a similar way, we can make sure that the code can easily handle different modelling software suites by creating a Model Abstraction Layer (MAL) and a self-describing data model to store model (meta)data, whether it is 1D, 2D or 3D, and for the last two cases, both for the structured and unstructured grid cases.

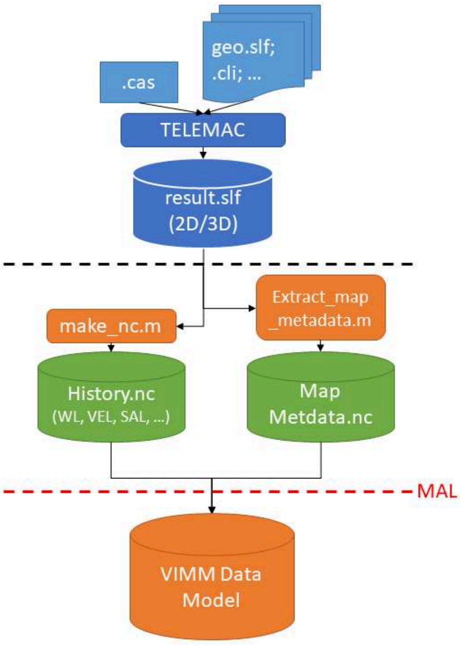

Figure 7 - The Model Abstraction Layer (MAL) as the red dashed line for a VIMM workflow involving TELEMAC results

The MAL is illustrated in Figure 7. Below this MAL, the code becomes agnostic of the modelling suite that was used.

Note a peculiarity in the case of post-processing TELEMAC results. Because the code (to the best of our knowledge) does not provide a standardised way to define output points in the model input to get history data (timeseries) in the model output, the user is left with two options. Either he/she implements a new subroutine to introduce output in history points in the TELEMAC source code (e.g. by extending the HERMES module). This requires knowledge however by the modeller of both FORTRAN and of the code structure of TELEMAC, and may lead to possible risks in quality assurance, as errors can be made in this module. Therefore, at FHR we choose to follow the dataflow in Figure 7, in which the history output is generated in post-processing out of the map data that is the standard output in the 2D and 3D .slf output file. This is of course not the most efficient way in terms of processing steps and the size of the model output.

E. *Modular design*

Using a MAL and DSAL simplifies the task of extending the toolbox to include new measurement sources (or data formats) and/or software suites.

Currently Mike11, Delft3D, Delft-FM, Simona and TELEMAC are supported. To the best of our knowledge, there is no alternative available on the market that provides this kind of flexibility in terms of the modelling suites it can process.

The modular design also means that the VIMM processing suite (the most important components of which are shown in Figure 8) is easily extendable, since these functions are below the MAL and DSAL in the VIMM workflow, so they work on fixed data structures that are independent of the file formats of model output and measurements.

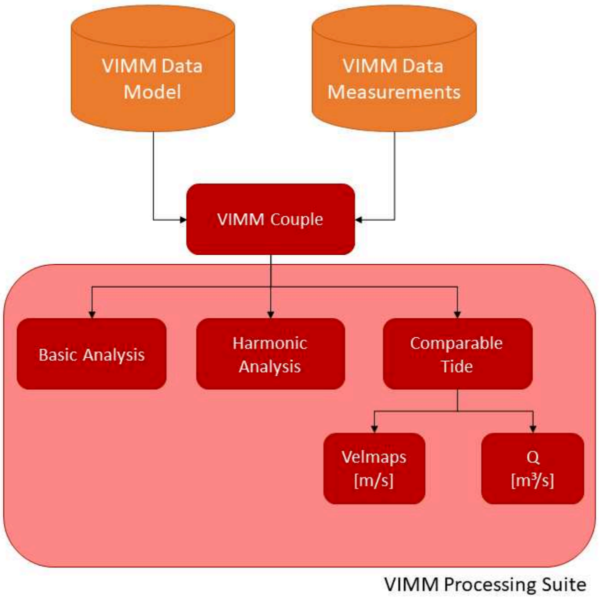

Figure 8 – Data flow in the VIMM processing suite

Two ideas of the VIMM processing suite are worked out below: Comparable Tide and Velmaps

IV. THE COMPARABLE TIDE (CT) METHOD

During model calibration of a hydraulic model, it is important to assess not only the model skill based on water levels (vertical tide), but also on the available velocity data (horizontal tide).

One useful data source for horizontal velocities is the so-called 13 hour measurement, in which a vessel sails the same transect over an entire tidal cycle, while measuring the water velocity with a bottom-mounted ADCP (Acoustic Doppler Current Profiler) device.

Due to the cost of performing them, such measurements are rare and far between, both in space and in time. Suppose the modeller wants to calibrate a model for the year 2019 against a 13h ADCP velocity measurement performed in 2017. He/she has three options: (1) drop the measurement out of the calibration dataset, because the model period and measurement period do not overlap; (2) perform an additional model run of a period in 2017 so that the measured tide is a part of the modelled period; (3) search for a comparable tide in the modelled period in 2019 that closely matches the tidal conditions during the measurement in 2017.

It is clear that option (1) is sub-optimal, because we would be shrinking the calibration dataset. Option (2) is only marginally better. In the real-life usecase of the calibration of SCALDIS, we plan to use a set of 34 different ADCP campaigns distributed over the entire estuary in order to assess the model skill in reproducing velocities. If we would follow option (2), we could (in the worst case) be forced to model 34 different subperiods for every calibration step. This would significantly complicate the calibration task, as the required number of runs would quickly explode.

With the Comparable Tide (CT) algorithm, this usecase becomes much more manageable. Figure 9 illustrates the CT method.

A. The CT Algorithm

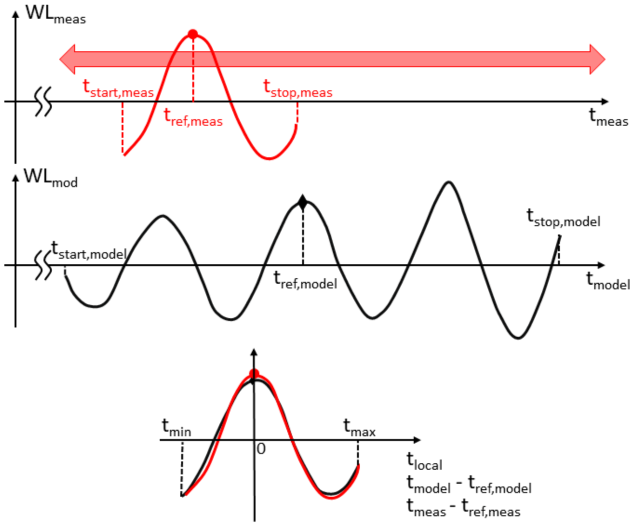

Figure 9 - The comparable tide method. Water level during the measurement campaign in red, modelled water level (at the same location) in black.

Imagine a measurement (e.g. a 13h ADCP campaign) that is executed in a timeframe [$t_{start,meas}$ $t_{stop,meas}$]. The model run is executed in the timeframe [$t_{start,model}$ $t_{stop,model}$]. Both periods do not necessarily overlap.

WL_{meas} is the measured water level at a station nearby the measurement location, for the period of that measurement campaign. In the case where the CT method is applied on a 13h ADCP measurement, this would be a water level station near the sailed transect. WL_{mod} is the modelled water level at the same station during the timeframe that the model results are available.

First one timestep in the measured time interval is designated as the reference time $t_{ref,meas}$ for the measurement. In Figure 9, we pick the moment of high water inside the measured interval, but the method works for any reference time. The aim is to construct a local time axis, expressed in hours before and after this reference time.

The algorithm now searches for the $t_{ref,model}$ that satisfies the following criteria: (1) it lies inside the modelled timeframe [$t_{start,model}$ $t_{stop,model}$]; (2) on the local time axis ($t_{model} - t_{ref,model}$) the error between the modelled and measured water level is minimised.

In VIMM, the user is presented with two options to express this error: RMSE and the bias-corrected $RMSE_0$. They are calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t_{min}}^{t_{max}} (\mathrm{WL}_{mod} - \mathrm{WL}_{meas})^2}{N}} \qquad (1)$$

$$RMSE_0 = \sqrt{\frac{\sum_{t_{min}}^{t_{max}} (\mathrm{WL}_{mod} - \mathrm{WL}_{meas} - \mu)^2}{N}} \qquad (2)$$

$$\mu = \overline{WL_{mod} - WL_{meas}}$$

With [t_{min} t_{max}] the overlapping timeframe on the local time axis and $\overline{WL_{mod} - WL_{meas}}$ the average error or bias between modelled and measured water level in the timeframe [t_{min} t_{max}].

By choosing RMSE as the error quantification, the algorithm will pick the modelled period where the modelled water level in the vicinity of the measurement campaign is the closest to the water level that was measured during the campaign. The idea is that by matching the water levels, the modelled velocity expressed on a local time axis (e.g. the velocity 1 hour before high water) will match the measured velocity on that local time axis, even if the measurement timeframe [$t_{start,meas}$ $t_{stop,meas}$] lies outside the modelled timeframe [$t_{start,model}$ $t_{stop,model}$].

By choosing $RMSE_0$ as the error quantification, the reasoning is much the same, with the important difference that a (small) bias between modelled and measured water levels is not counted in the cost function that is minimised to find $t_{ref,model}$. What is thought here, is that it is primarily the tidal amplitude and the shape of the tidal curve in between the tidal extremes that determine the velocities. The mean tidal level has only a secondary influence on the velocities, e.g. by determining which part of the bathymetry is subject to wetting and drying. A small bias between measured and modelled tide could for instance be related to a difference in surge. The choice between RMSE and $RMSE_0$ is ultimately up to the modeller.

Since local velocity is influenced by the bathymetry, the approximation that matching tidal conditions mean matching velocities only holds if the bathymetry in the model and the bathymetry during the measurement are not too different. In light of the morphological dynamics of the estuary, this means that the measurement timeframe [$t_{start,meas}$ $t_{stop,meas}$] should not be too far away from the modelled timeframe [$t_{start,model}$ $t_{stop,model}$]. It is up to the modeller to decide which measurements he/she can still take into account in the model calibration. Another case where matching tidal conditions do not necessarily mean matching velocities would be when density currents are known to be important.

V. The VelMaps module

As shown in Figure 8, the output of the CT method (basically a list of $t_{ref,model}$ for each model and for each measurement campaign) can be input for the VelMaps module. This module compares a modelled velocity field (map output, so representing a snapshot at one particular timestep in the model run) with a measured velocity transect. When using CT to determine $t_{ref,model}$, the timestamp of the measured transect and the modelled velocity field do not need to overlap.

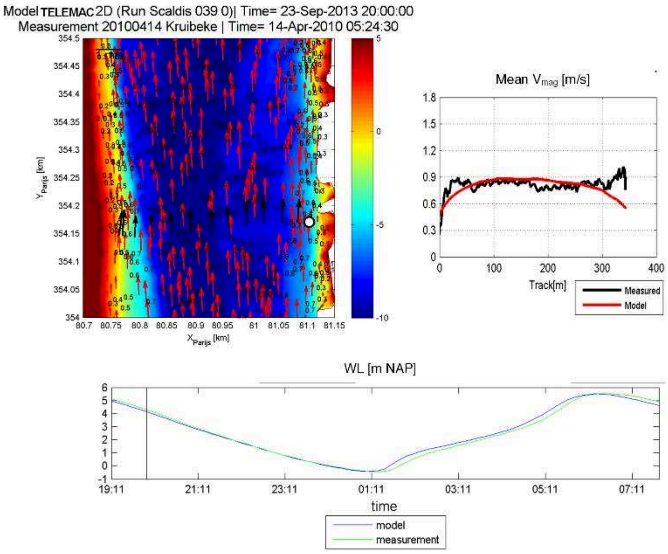

Figure 10 - Example output of the VelMaps module in VIMM

Figure 10 illustrates the different elements of the VelMaps module. The bottom panel illustrates the CT principle. Note in the figure title that we are comparing a model run of 2013 with a transect of an ADCP campaign measured in 2010. Modelled and measured velocities are compared on a local intratidal time-axis.

Once the modelled and measured velocities are superimposed, the modelled velocity field is interpolated on the measured transect. This allows the velocity magnitude to be compared in the top right panel. The following error statistics are subsequently calculated for each measured transect: bias of magnitude and direction, RMSE of magnitude and direction, and the Relative Mean Absolute Error (RMAE).

$$RMAE = \frac{\overline{\|\vec{v}_{mod} - \vec{v}_{meas}\|}}{\overline{\|\vec{v}_{meas}\|}} \qquad (3)$$

With ‖ ‖ denoting the length of a vector and the overbar the average over the transect. $\vec{v}_{mod}$ and $\vec{v}_{meas}$ are depth averaged velocity vectors. The result of equation 3 is one RMAE for every transect.

The error statistics (see also Table 3) are subsequently averaged over all transects to express the model skill in reproducing the flow distribution at that location.

If the measured transects are distributed over the estuary, this set of error statistics quantifies the model skill in reproducing the horizontal tide.

The RMAE can be classified in order to express model skill in a categorical way [6]:

Table 1 – Error classification and categorisation of results from tidal flow models, according to [6]

Model qualification		RMAE [-]
Excellent	🟦	<0.2
Good	🟩	0.2-0.4
Reasonable/fair	🟨	0.4-0.7
Poor	🟧	0.7-1.0
Bad	🟥	>1.0

Note that reference [6] actually uses an "adjusted RMAE", where an estimate of the measurement error is subtracted from the error measure in the numerator of equation 3. In VIMM, this estimate of measurement error is set to zero by default. This means that the classification following Table 1 is actually more strict than the classification in the original reference [6].

Because $\|\vec{v}_{meas}\|$ appears in the denominator of equation 3, the RMAE typically becomes larger during slack tide. We propose to also use the RMSE of velocity magnitude as a secondary error classification and categorisation [4].

Table 2 – Proposed error classification and categorisation based on the RMSE of velocity magnitude from tidal flow models

Model qualification		RMSE [m/s]
Good	🟩	<0.1
Reasonable/fair	🟨	0.1-0.2
Poor	🟧	0.2-0.3
Bad	🟥	>0.3

VI. Dimensionless cost function

When calibrating a model, it is important to start from the intended model use. While we would argue that it is always important to assess the model skill of a hydrodynamic model both for the vertical (water levels) and the horizontal tide (velocities, fluxes), the modeller still has to make a decision on the weight he/she is going to attribute to the different error statistics.

For each model run in a calibration process, the VIMM toolbox produces a broad range of different error statistics, both for the vertical and the horizontal tide. An overview of the different error statistics is given in Table 3.

Note that the different measurement data types that are listed for the vertical tide, are all derived from a regular timeseries, either by harmonic analysis, or by a local extrema analysis to derive high water (HW) and low water (LW). On the local extrema, the error in timing and level are quantified separately. That may be important, as in some applications the correct prediction of the level of HW can be more important than getting the timing correct.

A dimensionless cost function expresses model skill in one objective factor which represents improvements (decrease) or deterioration (increase) of the model performance, when compared to a reference run. A value lower than 1 indicates an improvement [7] [8].

A dimensionless cost function can be easily derived from the statistical output of the VIMM toolbox.

$$C = \sum \frac{max(Factor_i, Threshold_i)}{max(Factor_{i,ref}, Threshold_i)} \times Weight_i \qquad (4)$$

Each factor ‘i’ in the cost function is a different error statistic, with its own weight and threshold value. An expected observation error can be taken into account to assess the accuracy of the model reference in relation to the predefined modelling objective [9]. If an error statistic in the cost function drops below the threshold, the overall cost remains the same. This methodology helps to avoid giving too much weight to a very small improvement or deterioration of a parameter.

Table 3 – Overview of different error statistics that can be calculated by the VIMM toolbox.

	Measurement data type	Error statistic
Vertical Tide (water levels)	HW(LW)	RMSE of HW(LW), both of level and time
		Bias of HW(LW), both of level and time
		$RMSE_0$ of HW(LW) , both of level and time
	Timeseries	Correlation coefficient
		Bias
		RMSE
		$RMSE_0$
	Harmonic Analysis	For every component: amplitude and phase + uncertainty
		Vector difference
Horizontal Tide (velocities and fluxes)	Velocity at a fixed location	Velocity vectors: MAE and RMAE
		Magnitude: bias and RMSE
		Direction: bias and RMSE
	Fluxes	Bias
		RMSE
		Relative RMSE (to magnitude of measurement)
	Sailed ADCP	Magnitude: bias and RMSE
		Direction: bias and RMSE
		Velocity vectors: RMAE

VII. Discussion and Conclusions

The VIMM toolbox helps the modeller to efficiently perform a quantitative skill assessment of a hydrodynamic model.

This kind of standardised method to evaluate model performance can be included in a more general modelling framework that considers performance criteria as part of an iterative modelling process (e.g. automatic calibration). The broad range of error statistics that is calculated automatically by the VIMM toolbox can easily be fed into a weighted, dimensionless cost function that can serve as a guide during model calibration. The choice of which factors to include, with which weight and which threshold provides an objective and quantified way to communicate (and discuss) on a calibration strategy and should always be viewed in light of the intended model use.

The toolbox is coded in a generic way, so it can be easily set up for any model application, regardless of the modelling suite and the types of data sources that are used. This way the toolbox effectively becomes a knowledge platform that enables new insights to be shared more easily. The different forms of standard output of the toolbox (tables and figures) extend the vocabulary that modellers can use to communicate on model skill in a more objective way.

References

[1] J. Vanlede, S. Smolders, T. Maximova and M. Teles, “The unstructured SCALDIS model: a new 3D high resolution model for hydrodynamics and sediment transport in the tidal Scheldt,” in *E-proceedings of the 36th IAHR World Congress*, The Hague, the Netherlands, 2015.

[2] A. Van Braeckel, J. Speybroeck, J. Vanoverbeke, G. Van Ryckegem and E. Van den Bergh, "Habitatmapping Sea Scheldt subtidal - relation between benthic fauna and hydro- and morphodynamics (in Dutch)," INBO, Brussels - Belgium, 2018.

[3] S. Smolders, T. Maximova, J. Vanlede, T. Verwaest and F. Mostaert, “Integraal Plan Bovenzeeschelde: Subreport 1 – SCALDIS: a 3D Hydrodynamic Model for the Scheldt Estuary. Version 5.0.,” Flanders Hydraulics Research, Antwerp, Belgium., 2016.

[4] K. Chu, J. Vanlede, B. Decrop, J. Verwilligen and F. Mostaert, “Update snelheidsvelden Zeeschelde en Sluistoegangen,” Flanders Hydraulics Research, Antwerp, 2017.

[5] K. Chu, J. Vanlede, B. Decrop and F. Mostaert, “Validation of North Sea models: Sub report 1 – Validation and,” Flanders Hydraulics Research, Antwerp, 2020.

[6] J. Sutherland, D. Walstra, T. Chesher, L. van Rijn and H. Southgate, “Evaluation of coastal area modelling systems at an estuary mouth.,” *Coast. Eng.,* vol. 51, no. 2, p. 119–142, 2004.

[7] J. Vanlede, K. Delecluyse, B. Primo, B. Verheyen, G. Leyssen, Y. Plancke, T. Verwaest and F. Mostaert, “Verbetering randvoorwaardenmodel: Subreport 7 - Calibration of NEVLA 3D,” Flanders Hydraulics Research, Antwerp, Belgium, 2015.

[8] T. Maximova, S. Smolders and J. Vanlede, “Model calibration against different types of velocity data with a dimensionless cost function: application to the Scaldis model of the Scheldt estuary,” in *Proceedings of the 22st TELEMAC-MASCARET User Conference,15th-16th October 2015*, Daresbury Laboratory – UK, 2015.

[9] R. Vos and P. ten Brummelhuis, “Integrated data-modelling approach for suspended sediment transport on a regional scale,” *Coast. Eng.,* vol. 41, p. 177–200, 2000.