

H4918.37

Prepared for:

Rijkswaterstaat/RIKZ

Reducing the computational time of SWAN by dynamic deactivation of grid points

Activity 3.1 of SBW-RVW Waddenzee

Report

August 2007

h4918.37

wl | delft hydraulics

Prepared for:

Rijkswaterstaat/RIKZ

Reducing the computational time of SWAN by dynamic deactivation of grid points

Activity 3.1 of SBW-RVW Waddenzee

A. J. van der Westhuysen

Report

August 2007



| | | | | | | | |
|-------------------------|---|---|----------------|--|--------------------|---|--|
| CLIENT: | DG Rijkswaterstaat – Rijksinstituut voor Kust en Zee (RIKZ) | | | | | | |
| TITLE: | Reducing the computational time of SWAN by dynamic deactivation of grid points | | | | | | |
| ABSTRACT: | <p>The spectral wind wave model SWAN plays a key role in the estimation of the Hydraulic Boundary Conditions (HBC) for the primary sea defences of The Netherlands. Since some uncertainty remains with respect to the reliability of SWAN for application to the geographically complex area of the Wadden Sea, a number of activities have been initiated under project H4918 ‘Uitvoering Plan van Aanpak SBW-RVW Waddenzee’ (Plan of Action on the Boundary Conditions for the Wadden Sea) to devise a strategy for the improvement of the model. This activity is carried out in parallel with a measurement campaign that is being undertaken in the Wadden Sea to assist in the establishment of the boundary conditions (‘SBW-Veldmetingen’). In this context, hindcast and sensitivity studies carried out with SWAN for the Wadden Sea have shown that significant computational times are required to achieve results with the desired levels of numerical accuracy.</p> <p>In response to this finding, in the current project two methods for the reduction of computational time are investigated, namely the improvement of the initial guess used in the iterative solution procedure and the deactivation of converged grid points during the iteration process. The present report describes the latter of these two topics. In this study we investigated whether the numerical effort - and hence the simulation time - per iteration can be reduced by concentrating the iteration activity on those geographical regions that require the most iterations to converge. This involves the identification of geographical regions where the solution has already converged, for which the iteration procedure is subsequently suspended. This approach is therefore termed the <i>Dynamic Deactivation Method (DDM)</i>. The present study aims at demonstrating the potential of this approach in serial processing mode, and explores the settings required for its application, without performing an exhaustive optimisation.</p> <p>It is found that application of this technique to SWAN version 40.51A succeeds in significantly reducing simulation times while retaining a high degree of correspondence with the model outcomes of the original model. It was found, however, that results vary with the iteration behaviour of the model (which depends on the physical situation modelled) and with the method used to identify the convergence of the grid points.</p> | | | | | | |
| REFERENCES: | RIKZ contract RIKZ1797 (dated 9 March 2007) SAP bestelnummer: 45 000 73 341 | | | | | | |
| VER | AUTHOR | DATE | REMARKS | REVIEW | APPROVED BY | | |
| 1.0 | A.J. v/d Westhuysen | June 2007 | Draft | J. Groeneweg | M.R.A. van Gent | | |
| 2.0 | A.J. v/d Westhuysen | August 2007 | Final | W. E. Rogers | M.R.A. van Gent | | |
| | | | | | | | |
| | | | | | | | |
| PROJECT NUMBER: | | H4918.37 | | | | | |
| KEYWORDS: | | SWAN, SBW-RVW Waddenzee, Amelander Zeegat, computational speed-up | | | | | |
| NUMBER OF PAGES: | | 73 (including figures) | | | | | |
| CONFIDENTIAL: | | <input type="checkbox"/> YES | | <input checked="" type="checkbox"/> NO | | | |
| STATUS: | | <input type="checkbox"/> PRELIMINARY | | <input type="checkbox"/> DRAFT | | <input checked="" type="checkbox"/> FINAL | |

Contents

| | | |
|----------|---|------------|
| 1 | Introduction | 1—1 |
| 2 | Geographical distribution of convergence | 2—1 |
| 2.1 | Geographical locality of convergence | 2—1 |
| 2.2 | Differences in convergence speed..... | 2—2 |
| 2.3 | Spatial patterns of the iteration process..... | 2—2 |
| 2.3.1 | High tide with flood current | 2—3 |
| 2.3.2 | Low tide with ebb current | 2—5 |
| 3 | Dynamic deactivation of grid points | 3—1 |
| 3.1 | General approach | 3—1 |
| 3.2 | Deactivation criteria and regime | 3—2 |
| 4 | Evaluation of the DDM..... | 4—1 |
| 4.1 | General | 4—1 |
| 4.2 | Case1: High tide with flood current | 4—2 |
| 4.3 | Case 2: Low tide with ebb current | 4—4 |
| 5 | Discussion..... | 5—1 |
| 6 | Conclusions and Recommendations..... | 6—1 |

Appendices

| | | |
|----------|---|------------|
| A | Appendix..... | A—1 |
| A.1 | Deactivation regime and run control | A—1 |
| A.2 | Deactivation criteria | A—2 |

I Introduction

The spectral wind wave model SWAN (Booij et al. 1999) plays a key role in the estimation of the Hydraulic Boundary Conditions (HBC) for the primary sea defences of the Netherlands. Since some uncertainty remains with respect to the reliability of SWAN for application to the geographically complex area of the Wadden Sea, a number of activities have been initiated under project H4918 ‘Uitvoering Plan van Aanpak SBW-RVW Waddenzee’ (Plan of Action on the Boundary Conditions for the Wadden Sea) to devise a strategy for the improvement of the model. This activity is carried out in parallel with a measurement campaign that is being undertaken in the Wadden Sea to assist in the establishment of the boundary conditions (‘SBW-Veldmetingen’). In this context, hindcast and sensitivity studies carried out with SWAN for the Amelader Zeegat in the Wadden Sea (WL 2006; Royal Haskoning 2006; WL 2007) have shown that significant computational times are required (for the latter study, approximately 2.5 hours on a 3.4 GHz Pentium processor with 1.0 Gb internal memory) to achieve results with the desired levels of numerical accuracy. The computation of the complete HBC with SWAN, which includes a great number of environmental conditions and a model domain of the entire Wadden Sea, would therefore result in a substantial total computational time. This finding has led to a drive towards exploring ways to reduce the computation times of SWAN. In calculating the HBC, simulation times can be reduced either by employing parallel computing and high-performance processors in combination with the standard model code, or by streamlining the computational algorithm of the model itself (or a combination of the two). The current project explores the avenue of adapting the model code, in which two methods for the reduction of computational time are investigated. Firstly, the improvement of the initial guess used in the iterative solution procedure is pursued and, secondly, the deactivation of converged grid points during the iteration process is considered. The present report describes the latter of these two topics.

The wind wave model SWAN (Booij et al. 1999) solves the action balance equation, which describes the evolution of wind wave spectra in two dimensions of geographical space, two dimensions of spectral space and in time. The action balance equation has been discretised using an implicit scheme in geographical space and an explicit schemes in spectral (frequency and directional) spaces. The discretisation of the action balance equation yields a system of linear equations that need to be solved. This system can be solved in one step by using a Gauss-Seidel technique (Press et al. 1993). However, it is well known that adapting the ordering of updates of the unknowns (in this case the action density) in geographical space according to the propagation direction can improve the rate of convergence of the Gauss-Seidel iterative procedure (Wesseling 1992). Hence, the action density is solved by means of four sweeps over the geographical space – one for each quadrant of wave propagation direction.

The final solution of the action balance equation is not found after just one set of four sweeps, however, but needs to be repeated during a number of iterations of these sweeps (henceforth referred to simply as *iterations*). This is for a number of reasons. Firstly, action density can be transferred from one directional quadrant (sweep direction) to a neighbouring quadrant by the processes of refraction and quadruplet nonlinear interaction. This would require the sweep for the neighbouring quadrant to be repeated during a subsequent iteration. Secondly, in order to stabilise the source term integration, SWAN makes use of an

action density limiter (Hersbach and Janssen 1999) that limits the amount of change in action density during each iteration. After each set of four sweeps, the total change in each spectral bin is truncated to a certain percentage (default 10 %) of the Phillips equilibrium spectrum. This implies that the actual change in action density prescribed by the physics may not be realised after only one Gauss-Seidel solution procedure of four sweeps.

Studies have shown that the influence of the action density limiter is the primary reason for requiring multiple iterations (e.g. Zijlema and Van der Westhuysen 2005, Fraza 1998). In non-stationary simulation, the change in action density per time step prescribed by the model physics tends to be of the same order as the amount allowed by the action limiter, so that three iterations per time step appears to be sufficient (Fraza 1998). Stationary simulation, on the other hand, typically requires many more iterations before convergence is reached. During the stationary solution procedure, the time step is infinite, so that the change in action density during a single iteration can be far greater than the amount of change allowed by the action limiter. To alleviate this problem, stationary SWAN simulations are initialised with a so-called ‘first guess’ of the final solution, so that the amount of change required to reach the converged solution is reduced. Yet a number of studies have shown that in stationary mode SWAN still often requires more than 30 iterations to reach full convergence (e.g. Van der Westhuysen et al. 2005; Alkyon 2007). This slow convergence can be seen in wave parameters such as the significant wave height, period measures and the mean wave direction. Since the computational time per iteration can be significant for detailed simulations, the need for such a great number of iterations can require substantial total computational time. For the application of SWAN to the Wadden Sea to derive the HBC, interest is primarily in the stationary mode of simulation, hence the remainder of this study will be limited to this mode of operation.

Efforts to reduce the total computational time can be directed towards either the reduction of the total number of iterations required, or the reduction of computational time per iteration. In the present study we focus on the latter, with the hypothesis, proposed by Alkyon (2005), that the computational effort may be reduced by concentrating the numerical effort on geographical regions where convergence is the poorest. Numerical investigations (e.g. Alkyon 1999, 2007) have suggested that convergence is typically not reached simultaneously over a given geographical domain. This is due to the fact that different combinations of source terms are active over the various regions, each combination having its own characteristic convergence speed. For example, a balance of only deep water source terms (wind input, whitecapping dissipation and quadruplet interaction) may be expected to converge at a different rate than a combination in which depth-induced breaking and current interaction is also active.

In the present study we investigated whether the numerical effort - and hence the simulation time - per iteration can be reduced by concentrating the iteration activity on those regions that require the most iterations to converge. This involves the identification of geographical regions where the solution has already converged, for which the iteration procedure is suspended. Active simulation is continued only over those grid points that have not reached convergence yet. Effectively, the active computational grid gradually diminishes during the iteration procedure, based on the geographical distribution of converged grid points, and hence this approach is termed the *Dynamic Deactivation Method (DDM)*. The present study aims at demonstrating the potential of this approach, and explores the settings required for its application, without performing an exhaustive optimisation. In evaluating the

convergence of grid points, only the significant wave height and mean period T_{m01} are considered in this first evaluation. Furthermore, this study is restricted to investigating serial processor computation, and does not consider additional issues involved in parallel computing (see discussion in Section 5). All computations in this study were performed using a single AMD Athlon64 processor model 4000+, with a 2.4 GHz clock speed and 4.0 Gb of internal memory. It will be shown that application of the DDM technique to SWAN version 40.51A succeeds in significantly reducing simulation times while retaining a high degree of correspondence with the model outcomes of the original model. It was found, however, that results vary with the iteration behaviour of the model (which depends on the physical situation modelled) and with the method used to identify the convergence of the grid points.

This study was carried out by André van der Westhuysen. The internal quality assurance and review was carried out by Jacco Groeneweg, and the external review was done by W. Erick Rogers (Naval Research Laboratory, USA).

In Section 2 of this report, a detailed analysis of the spatial distribution of the convergence behaviour of SWAN is made. This is done for two idealised cases (deep water wave growth and a prismatic beach) and two field cases of the Amelanders Zeegat in the Dutch Wadden Sea. The primary question in this analysis is as to whether there are sufficiently large differences in the convergence speeds of the various sub regions of a typical Wadden Sea computational domain to make the application of a dynamically deactivating grid viable. This is indeed found to be the case. In Section 3 a description is given of the dynamic deactivation method and its implementation in the SWAN source code. Section 4 presents an evaluation of the dynamic deactivation method for the two field cases in the Amelanders Zeegat, highlighting situations in which it is effective and others where further development is required. In Section 5, the results of this study are discussed. Section 6 closes the report with conclusions and some recommendations.

2 Geographical distribution of convergence

In this section, an analysis is made of the geographical distribution of the convergence behaviour of SWAN, to determine how convergence patterns develop during a simulation. In this investigation, answers to the following questions are sought: firstly, does the bulk of the convergence effort during stationary simulation occur locally in geographical space, or do regions of non-convergence propagate through the computational domain? In other words, does it take more iterations for a spectrum at a given geographical location to work away the error due to non-convergence than it takes for this error to propagate through the computational domain? Secondly, given that regions of non-convergence remain stationary in geographical space, are there certain areas where convergence is reached quickly and other areas where many iterations are required?

The answers to these questions are sought by first analysing two idealised situations, in which some important characteristics of the iteration behaviour are considered in isolation. Subsequently, two field cases of the Amelanders Zeegat are considered in order to investigate the spatial patterns of iteration behaviour in this complex geographical region.

2.1 Geographical locality of convergence

To determine whether non-convergence in SWAN simulations is essentially a problem occurring in spectral space (and hence locally in geographical space) or whether it lies mainly in the slow propagation of converged results from upwind locations through the computational domain, we consider a simple situation of fetch-limited wave growth in deep water. In this first test, a wave field is generated by a wind of $U_{10} = 30$ m/s over a fetch of 100 km, which is simulated in one-dimensional mode. At a fetch of 100 km, the waves attain a height of $H_{m0} = 8.7$ m and a mean period of $T_{m01} = 8.6$ s. Figure 2.1 shows the development of these integral parameters during the iteration process at three locations, namely at fetches of 10 km, 50 km and 100 km. Regarding first the upwind point at a fetch of 10 km, it can be seen that the convergence of wave height and mean period is fairly slow and erratic, with full convergence (save for small amplitude oscillations) reached after 50 iterations. The convergence of the two downwind output locations (at 50 km and 100 km) occurs during the same iteration window as for the 10 km location, and even reaches convergence somewhat more quickly.

The finding that there is no phase difference (in iteration time) between the convergence of upwind and downwind locations is significant. This means that the propagation of spectral updates made in geographical points upwind moves through the computational domain quickly, updating points downwind within the same iteration. This result can be explained by considering the fact that in stationary simulation the time step is infinite, and hence information can propagate fully through the computational domain during a single iteration. The large number of iterations is therefore spent not in propagating energy updates through the computational domain, but in reaching convergence within the spectrum, locally in geographical space. This convergence effort is spent mainly to overcome the influence of the action limiter (Zijlema and Van der Westhuysen 2005). The results of this elementary test therefore suggest that areas of poor convergence remain stationary in geographical space, and hence that it may be possible to subdivide the computational domain into distinct areas of fast and slow convergence.

2.2 Differences in convergence speed

The second case considered is that of a bar beach system at Petten, The Netherlands, of which the coast-normal beach profile is shown in Figure 2.2 (bottom panel). In the present analysis, this field situation is idealised by defining a one-dimensional SWAN simulation along this coast-normal transect running from the deep water measuring station MP1 to station MP6 in the surf zone. In this idealised simulation, we can observe the evolution of the computed integral parameters along the MP1-MP6 transect during the iteration process. The field case considered was recorded on 02/01/1995 at 06:00 at which time a swell wave field with $H_{m0} = 5.39$ m and $T_p = 12.5$ from NW was incident nearly perpendicular to the coast, while a wind with $U_{10} = 19.0$ m/s from W generated local wind sea.

Figure 2.2 shows the spatial evolution of the significant wave height and mean wave period, together with the water depth along the MP1-MP6 transect, obtained after 5, 10 and 30 iterations. The deep water model boundary is placed at the station MP1, which is in a water depth of -21 m. Moving toward the shore, a bank (the Pettemer Polder) is found, which dissipates a significant amount of wave energy during storms. The shoreline features a bar beach system over which the remaining wave energy is dissipated. The model domain can therefore be divided into regions based on the physical processes which are active. Offshore of the Pettemer Polder bank, the processes of wind input, whitecapping dissipation, quadruplet interaction and bottom friction are dominant. At the Pettemer Polder and the bar beach system the processes of depth-induced breaking and triad nonlinear interaction become dominant.

Figure 2.2 shows the evolution of wave height and mean wave period at iterations 5, 10 and 30 during the simulation. These results show that after 5 iterations the region from the breaker zone at the Pettemer Polder has effectively converged. However, the region from the breaker zone at the Pettemer Polder to the beach requires more than 30 iterations before convergence is reached. These results show that, for an idealised barred beach situation, the modelling domain can be divided into two regions, based on required computational effort: the region offshore of the bank, where only a few iterations are needed to reach convergence and the region shoreward of the bank, where a large number are required. We can therefore suppose that a solution procedure that would focus only on the surf zone regions after the fifth iteration would require almost half the total computational effort without compromising the final converged solution. In the next section this result is generalised to two dimensions in horizontal space.

2.3 Spatial patterns of the iteration process

The modelling results presented above suggest that, during stationary simulation, the bulk of the iteration effort lies with reducing convergence errors locally in geographical space, and that these regions of non-convergence do not migrate through the computational domain. An example of such a region of slow convergence is the surf zone, in which the combined action of depth-induced breaking and triad nonlinear interaction requires relatively many iterations to converge. In the present section these findings are verified for two complex field cases in the Amelanders Zeegat in the Dutch Wadden Sea (Figure 2.3) that feature a variety of physical processes and bathymetry that varies in two horizontal dimensions.

2.3.1 High tide with flood current

Figure 2.4 shows the first of the two field cases of the Amelander Zeegat considered here, taken from the set of storms hindcasted by Royal Haskoning (2006). This Wadden Sea region is a complex combination of barrier islands (Terschelling to the west and Ameland to the east), through which a deep tidal channel runs, ending in a complex system of tidal channels and flats inshore of the islands. Offshore of the tidal gap a shallow ebb tidal delta is found. The figure also shows a number of small shoals and islands throughout the modelling domain, which show up as small white areas. This first field case, taken on 16/12/2005 at 10:00, features an offshore wave condition of $H_{m0} = 5.4$ m and $T_{m-1,0} = 9.5$ s from NW, with a wind of $U_{10} = 17.5$ m/s from NNW. The lower panel of Figure 2.4 shows the current field for this simulation time, which was computed by the WAQUA flow model on the 'Kuststrook' and more detailed 'Kuststrook-Fijn' model grids. At the time of the observations it was high tide, and the results of the flow model show a flood current of about 0.6 m/s in the main tidal channel. Based on tidal observations along the coasts of Terschelling (Station TERS) and Ameland (Station NES), a spatially uniform water level of +2.0 m NAP is set over the model domain. Also shown on Figure 2.4 are six locations at which model behaviour will be evaluated in detail in the following sections. Figure 2.5 shows spatial plots of the simulated significant wave height and mean period found after 40 iterations, at which time the stationary simulation has generally converged (see below). These results show that high wave energy and mean periods are found on the North Sea side of the barrier islands, which is dissipated for the most part either on the coasts of the barrier islands or on the ebb tidal delta. Due to the NNW wind, waves are regenerated behind the barrier islands. However, since the water depth is limited and the fetch short (about 15 km) this wave growth is restricted. We note that at the western boundary unrealistically large wave heights and periods exit due to sub-optimal boundary conditions. These are, however, removed from the modelling area of interest (the tidal gap and the mainland shore) and are not of importance in the present study.

Next, the spatial distribution of the iteration process that arrives at this converged result is considered. Here we define two types of convergence errors: firstly, the global convergence error is defined as the difference between the value of the solution at the current iteration and its value upon convergence. This gives an indication of the *total distance to convergence* that must be bridged by the iteration process, and the actual error made if the iteration process were to be broken off at the current iteration level. The global convergence error is not known during the simulation, since it is a function of the converged model solution. Secondly, the local convergence error is defined as the difference between the solution at the current iteration and its value in the previous iteration. The local convergence error is a measure of the *rate of convergence*, and can be computed during the iteration procedure. The ratio of the global convergence error to the local convergence error gives a linearised indication of the number of iterations required to reach convergence. If either the global convergence error were to be large or the local convergence error were to be small, many iterations would be needed for convergence. We note that the local convergence error is often taken as a substitute for the (unknown) global convergence error, for example in the convergence criteria of SWAN. This holds the danger that a small value for the local convergence error can be interpreted as an indication that the model is near convergence, which is not necessarily the case.

Figures 2.6 and 2.7 show spatial plots of the global convergence error in significant wave height and mean period after 5, 10, 20 and 30 iterations, with the iteration level indicated by s . (The white parts in the wet areas of these figures indicate that the convergence error is below -10%, i.e. a negative difference in excess of 10%. The solid black lines indicate the approximate position of the coastlines of the islands and the mainland. The dashed black line indicates the boundary of a land reclamation site ('kwelder') along the Frisian coast.) As outlined above, these plots indicate the total error due to non-convergence made at the iteration in question - regions that have small global convergence errors are essentially converged, whereas non-zero global convergence errors indicate non-convergence. Interpreting the results in this way, it can be seen that the region offshore of the barrier island surf zone converged after just 5 iterations. Inside the surf zone of the barrier islands and on the ebb tidal delta, however, convergence is slower. It takes about 30 iterations before the global convergence errors in wave height and mean period have become small. The other significant regions where many iterations are required to reach convergence are immediately behind the barrier islands, where young wind sea is generated. Here too, about 30 iterations are required before global convergence errors have diminished sufficiently, and convergence is reached. As was found in the idealised simulations presented above, distinct areas of fast and slow convergence seem to exist, and the latter do not seem to migrate through the simulation domain during the iteration process.

Next, the spatial distribution of the rate of convergence is considered. Figures 2.8 and 2.9 show the local convergence error at the corresponding iteration levels considered above. For this case, the rate of convergence tends to be high in areas where the global convergence error was found to be high (Figures 2.6 and 2.7). As the iteration process proceeds, the rate of convergence soon drops, although it remains relatively high in two regions, namely for the wave height inshore of the barrier islands and for the mean wave period offshore in the surf zone offshore of the barrier islands.

To summarise the results of the global and local convergence errors, Figure 2.10 shows the values of significant wave height and mean wave period as a function of iteration level for the six locations plotted in Figures 2.4 to 2.9. The results have been classified as areas offshore of the barrier islands (locations 1 to 3), areas behind the barrier islands (locations 4 and 6) and on the salt marshes shorewards of the tidal flats (location 5). The iteration behaviour of the three geographical areas can be clearly distinguished. Wave heights and mean periods offshore of the barrier islands diminish during the iteration procedure, whereas they grow strongly for the locations in the lee of the islands. In both geographical regions, however, convergence is not reached before 40 iterations. By contrast, the location on the tidal flats in front of the mainland reaches convergence after about ten iterations.

Finally, Figure 2.11 shows the corresponding energy density spectra in the relative frequency space for the six output locations. These plots compare the solution of the wave spectra after 5 iterations with the converged results obtained after 40 iterations. For the locations in the surf zone of the barrier islands, the iteration effort goes towards reducing the spectral energy (through depth-induced breaking) and towards producing a first harmonic. In the lee of the barrier islands, and also on the tidal flat, the iteration effort goes into growing the young wind sea.

2.3.2 Low tide with ebb current

Figure 2.12 presents the second of the two Amelanders Zeegat field cases considered here. This second field case, taken on 09/02/2006 at 11:00, features an offshore wave condition of $H_{m0} = 5.0$ m and $T_{m-1,0} = 10.0$ s from NW, with a wind of $U_{10} = 19.5$ m/s, also from NW. Figure 2.12 shows the current field for this simulation time, again computed by the WAQUA flow model. At the time of the observations it was ebb tide, with a maximum computed current in the main tidal channel of about 0.7 m/s, and a weaker current of about 0.2 m/s over the tidal flats. Based on tidal observations, a spatially uniform water level of +0.5 m NAP is set over the model domain. Figure 2.13 shows spatial plots of the simulated significant wave height and mean period, which is found after 40 iterations with a stationary model run. As with the storm of 16/12/2005, the wave field on the North Sea side of the barrier islands is characterised by high wave heights and high periods, both being strongly reduced on the coasts of the barrier islands and on the ebb tidal delta. Wind sea is generated over the region behind the barrier islands, reaching wave heights of up to about 1 m.

Figures 2.14 and 2.15 show the spatial distribution of the global convergence error at the iteration levels 5, 10, 20 and 30, indicating which regions of the computational domain require the most iterations to converge. These results show a large degree of correspondence with those of the high tide case considered above: the region offshore of the barrier islands converges after just a few iterations, whereas significantly more iterations are required on the ebb tidal delta and along the North Sea coastline of the barrier islands. Also, in the region behind the barrier islands, where young wind sea is found, more than 20 iterations are needed to reduce the global convergence error to an acceptable level. A distinct difference between the results of the present case and those considered above is the influence of the ebb current on the convergence characteristics on the tidal flats. Figures 2.14 and 2.15 show that after 20 iterations, the region running from the tidal inlet to the mainland coast still has areas where the global convergence errors in both significant wave height and mean period exceed 4%. Correlating this region with the bathymetry and computed current field (Figure 2.12) shows that it is characterised by a small water depth (tidal flats and salt marches), over which the current is in the opposite direction to wave propagation.

Figures 2.16 and 2.17 show the local convergence error over the computational region. Based on these results, the largest convergence rates are seen to be in the immediate regions along both the North Sea and Wadden Sea sides of the barrier islands, and also on the ebb tidal delta. Comparison with Figures 2.14 and 2.15 shows that these regions match the regions where the global convergence error is large, suggesting good convergence behaviour. However, an important exception is again the region running from the tidal inlet to the mainland coast. Here the global convergence error remains large during most of the simulation, whereas Figures 2.16 and 2.17 show that here the local convergence error - and hence the convergence rate - is small. This suggests that the convergence in this region is poor, requiring many iterations. Figure 2.18 shows the iteration curves of six output locations plotted in Figures 2.12 to 2.17. These include three locations on the North Sea side of the barrier islands (locations 1 to 3) and three locations over the tidal flats behind the tidal gap (locations 4 to 6). The iteration curves show a distinct difference in the iteration behaviour of the locations on the North Sea side of the barrier islands from that in the area behind the tidal gap. The former group of locations - all lying in the surf zone - start with large global convergence errors in wave height and period, which diminish steadily during

the course of the simulation. The latter group of locations starts with relatively small global convergence errors, but the mean wave period converges slowly so that convergence is not reached before 40 iterations.

Finally, Figure 2.19 shows the corresponding energy density spectra in the relative frequency space for the six output locations. As above, these plots compare the solution of the wave spectra after 5 iterations with the converged results obtained after 40 iterations. For the three locations in the surf zone of the barrier islands, the spectra decrease significantly in energy during the iteration process and develop higher harmonics. By contrast, at the three locations behind the tidal inlet, the iteration effort goes towards transporting energy to higher frequencies due to Doppler shifting in the counter current. The changes in the spectra due to this Doppler shifting are evidently relatively small compared to those in the surf zone, but require many iterations to realise.

From the results presented above it can be concluded that the various sub regions in the Ameland Zeegat computational domain vary with respect to the number of iterations that they require for convergence. Regions with strong depth-induced breaking and triad interaction, such as the North Sea coast of the barrier islands and the ebb tidal delta, require many iterations, as do the regions immediately downwind of the barrier islands, where young wind sea is generated. Some areas on the tidal flats under the influence of wave-current interaction also appear to converge slowly. These regions of slow convergence seem to remain stationary in geographical space during the iteration process, and not to migrate through it. By contrast, regions further offshore of the barrier islands, and regions more toward the mainland coast appear to converge quickly. It can be concluded that these results provide sufficient grounds on which to design a numerical approach that is aimed at reducing the computational time per iteration by focusing the iteration effort on those geographically isolated regions where the convergence is the slowest. This method is detailed in Section 3.

3 Dynamic deactivation of grid points

From the analysis presented in Section 2, it was seen that, in stationary simulation mode, the greater part of the iteration effort goes into balancing the source terms at each geographical grid point, and not into propagating convergence errors through the model domain. Hence, the convergence speed of a particular geographical region could be expected to depend on the combination of physical processes that are active there. It was shown in Section 2.3 that for the investigated model domain of Amelanders Zeegat, coherent areas of substantial size exist that require significantly fewer iterations to converge than others. This finding can be used to reduce the computational effort required in the simulation. This section presents a method in which the converged grid cells are deactivated as the iteration process progresses, reducing the computational effort per iteration, and hence the total computational time. Section 3.1 presents the general approach of this method, whereas Section 3.2 discusses the criteria used for the deactivation (and re-activation) of grid points. Full details of the implementation into the SWAN code are given in Appendix A.

3.1 General approach

In Section 1 it was stated that SWAN is solved by a four-sweep Gauss-Seidel method, which, due to the nature of the physics modelled, needs to be repeated in a number of iterations. The computational core of SWAN therefore features an overall iteration loop, containing an inner loop for the four directional sweeps. During each of the four directional sweeps, the action balance equation is solved at each geographical grid point, starting at the upwind corner of the computation grid and moving in the downwind direction. This basic computational structure of SWAN is summarised below (see e.g. Campbell et al. 2002).

```
SUBROUTINE SWCOMP
...pre-iteration setup...
DO ITER = 1, ITERMX
  DO SWPDIR = 1, 4
    ...set sweep parameters...
    DO IY = IY1, IY2, -IDY
      DO IX = IX1, IX2, -IDX
        CALL SWOMPU(...,IX,IY,...)
      END DO
    END DO
  END DO
...check convergence...
END DO
END SUBROUTINE SWCOMP
```

Here the iteration level is indicated by ITER, each of the four sweeps indicated by SWPDIR and the geographical grid point by the set (IX,IY). The subroutine SWOMPU contains the subprogrammes that calculate the propagation velocities, the source terms and the solver for the linear system.

An analysis of the total computational time of a simulation (see, for example Figure 4.9 below) reveals that a significant portion of time is spent in subroutine SWOMPU to

compute - for each geographical grid point - the source terms, the local propagation velocities and to solve the resulting linear system of equations. In the default model, these time-consuming activities are carried out for all geographical grid points during each iteration, irrespective of whether the iteration activity is actually required there or not. Considering the analysis presented in Section 2, distinct areas can be identified on which computational effort is spent without gains to the model result.

The basic concept of the method proposed here is that simulation time can be saved by skipping unnecessary and time-consuming numerical updating in those geographical grid points that have converged prior to the end of the simulation. This is accomplished by ascribing a binary attribute to each geographical grid point that registers whether or not it has converged, and hence whether or not it needs to be updated in a subsequent iteration. For those geographical grid points that have not yet converged, the action balance equation is fully computed. However, for points that have converged, simulation time can be saved by omitting any part of the computation of the action balance equation (carried out inside the subroutine SWOMPU). For example, time can be saved by not recalculating the source terms at each iteration, and simply solving for changes in propagation. However, in order to still solve the linear system, this would require the source term results of the previous iteration to be stored, and therefore require additional memory. Furthermore, a reasonable amount of computational time is spent in computing propagation velocities, solving the linear system and the associated administration. Therefore, in this study the calculation of the action balance equation is omitted entirely for converged points. For these points, the subroutine SWOMPU simply takes the results of the previous iteration as the solution of the current iteration.

3.2 Deactivation criteria and regime

Two important considerations come to play in the implementation of the basic method outlined above. Firstly, the criteria on the basis of which grid points are deactivated must be set. Secondly, some general rules governing the deactivation process are required.

Central to the method of grid point deactivation is the choice of deactivation criteria. This method can only work effectively if converged points and points that require further iteration can be well discriminated. Weak criteria that deactivate grid points prematurely (before they are converged) will lead to a large reduction in simulation time, but will result in a non-converged model outcome. Very strict criteria that keep a large percentage of points active will reduce the chance of a non-converged result, but will minimise the benefits of the deactivation technique. The choice of deactivation criteria is therefore a balance between the accuracy of the method on the one hand, and the gain in computational time on the other.

A natural choice for the deactivation criteria are those used for the termination of the iteration process of SWAN. These convergence criteria are used to identify the individual grid points in the computational grid that have converged and, if these points reach a predetermined percentage of the total, end the iteration process. We will use these criteria in a similar way to determine which grid points are sufficiently converged to be deactivated. Three alternative sets of deactivation criteria were investigated in this study. The first of these are the default gradient-based criteria, which deactivate a grid point when

$$\left| \frac{H_{m0}^s - H_{m0}^{s-1}}{H_{m0}^{s-1}} \right| < \varepsilon_H^r \quad \text{or} \quad \left| H_{m0}^s - H_{m0}^{s-1} \right| < \varepsilon_H^a \quad (3.1)$$

and

$$\left| \frac{T_{m01}^s - T_{m01}^{s-1}}{T_{m01}^{s-1}} \right| < \varepsilon_T^r \quad \text{or} \quad \left| T_{m01}^s - T_{m01}^{s-1} \right| < \varepsilon_T^a, \quad (3.2)$$

where H_{m0}^s and T_{m01}^s are the significant wave height and mean period values at iteration s . The parameters ε_H^r and ε_T^r denote the maximum allowable relative change, whereas ε_H^a and ε_T^a denote the maximum allowable absolute change. These parameters give the gradient of the iteration curves of wave height and period and represent the local convergence error. In this study, we used the settings $\varepsilon_H^r = \varepsilon_T^r = 0.01$, $\varepsilon_H^a = 0.01$ m and $\varepsilon_T^a = 0.01$ s, which are stricter than the default values ($\varepsilon_H^r = \varepsilon_T^r = 0.02$, $\varepsilon_H^a = 0.02$ m and $\varepsilon_T^a = 0.02$ s). The second criteria are those proposed by Zijlema and Van der Westhuysen (2005), which are based on the curvature of the iteration curve of the significant wave height. A grid point is deactivated when

$$\left| \frac{H_{m0}^s - (H_{m0}^{s-1} + H_{m0}^{s-2}) + H_{m0}^{s-3}}{H_{m0}^s} \right| < \varepsilon_{CH}, \quad s = 3, 4, 5, \dots \quad (3.3)$$

where ε_{CH} is the maximum allowable curvature of the iteration curve of the significant wave height. During the course of the present study it was found that, especially in cases featuring wave-current interaction, the convergence of the mean period can be slower than that of the wave height. Therefore, the criterion of Zijlema and Van der Westhuysen (2005) has been extended to also include the curvature of the iteration curve of the mean period:

$$\left| \frac{T_{m01}^s - (T_{m01}^{s-1} + T_{m01}^{s-2}) + T_{m01}^{s-3}}{T_{m01}^s} \right| < \varepsilon_{CT}, \quad s = 3, 4, 5, \dots \quad (3.4)$$

where ε_{CT} is the maximum allowable curvature of the iteration curve of the mean period T_{m01} ¹. The criteria (3.3) and (3.4) are applied with two sets of values, namely $\varepsilon_{CH} = \varepsilon_{CT} = 0.005$ and $\varepsilon_{CH} = \varepsilon_{CT} = 0.001$, representing the second and third deactivation criteria respectively.

An overall regime for the activation and deactivation of grid points is required. For example, during the first few iterations it is desirable for all grid points to be active, in order for initial convergence patterns to develop. Also, the application of the technique described above will lead to areas within the computational domain that are inactive for an appreciable time

¹These criteria can be extended even further, to also include wave direction parameters, as suggested by Alkyon (2007). This is not considered in the present study.

during the iteration process. Since these grid points may be neighbouring active points whose solution is still developing, it is to be expected that discontinuities in the solution may develop between active and inactive grid points. Such discontinuities can be prevented by, for instance, reactivating all grid points at regular intervals during the iteration process. In this way all formerly inactive points can be updated in accordance with the continuing changes in the active parts of the computational grid. The result of such a reactivation step can be that a formerly inactive grid point undergoes a sufficient change in its solution to remain activated in the succeeding iterations.

In the early stages of this study a few alternatives for the governing deactivation regime were investigated, including regimes where (a) grid points are deactivated purely on the basis of the deactivation criteria and remain inactive for the remainder of the simulation, and (b) where all grid points are forced to be active during the first five iterations, whereafter they are free to be deactivated based on the deactivation criteria. Due to the reasons given above, these two regimes resulted in model results with unacceptably large errors relative to those of the original full model. These regimes were therefore dropped in favour of the single regime presented in this report, namely one in which all grid points are forced to be active during the first five iterations and are reactivated at regular iteration intervals thereafter. In the present study, this regime was evaluated for an arbitrarily chosen reactivation interval of five iterations. In other words, all grid points are activated when

$$[s \leq 5] \quad \text{or} \quad [\text{mod}(s, 5) = 0] \quad (3.5)$$

In the intermediate iterations, however, all points that meet the deactivation criteria are not updated.

4 Evaluation of the DDM

4.1 General

In this section the dynamic deactivation method described above is evaluated for the two Amelanders Zeegat field cases presented in Section 2. The overall aim of the evaluation is to determine the reduction in computational time relative to the original full computation that can be achieved by applying this method. However, the application of the DDM may impact negatively on model accuracy. This is because, as will be shown, the premature deactivation of grid points can leave certain regions of the computational grid unconverged. In the presentation of the results, reference will therefore be made to the solutions of the original model, in which all grid points remain active.

In Section 3 the importance of the choice of deactivation criteria was discussed. The importance of these criteria is illustrated here by presenting the performance of the dynamic deactivation method for each of the three criteria presented in Section 3.2. This results in three run codes, which are defined in Table 4.1.

| Run code | Deactivation criteria |
|----------|---|
| Run 1 | Gradient-based: $\varepsilon_H^r = \varepsilon_T^r = 0.01$, $\varepsilon_H^a = 0.01$ m, $\varepsilon_T^a = 0.01$ s |
| Run 2 | Curvature-based: $\varepsilon_{CH} = \varepsilon_{CT} = 0.005$ |
| Run 3 | Curvature-based: $\varepsilon_{CH} = \varepsilon_{CT} = 0.001$ |

Table 4.1: Run codes for the three different deactivation criteria.

Using these three model settings, simulations were carried out for the two Amelanders Zeegat field cases presented in Section 2.3, which represent a flood and an ebb situation respectively. Apart from the fact that the DDM is applied, the model setup of these simulations is identical to those presented in Section 2.3. The simulations were therefore set up with observed spectra along the boundaries, spatially uniform wind and water level fields based on observations, and computed WAQUA current fields. In the analysis presented below, the results of these simulations applying the DDM are compared with those of the full simulations using the original SWAN model. In these simulations, the convergence criteria of SWAN were not used, but 40 iterations were enforced instead, at which point the solutions were fairly well converged (Figures 2.10 and 2.18). For reference, when using the curvature-based convergence criteria, and requiring a maximum curvature of $\varepsilon_{CH} = \varepsilon_{CT} = 0.001$ at 99.5% of all geographical points, the simulations of these two Amelanders Zeegat field cases are terminated after 33 and 26 iterations respectively. Forcing both sets of simulations to continue up to the convergence point of the full simulation provides an objective measure of the loss in accuracy (error due to non-convergence) resulting from the speeding up the simulation with the DDM.

The results of the evaluation are presented along the following lines: for both the field situations, the performance of the dynamic deactivation method using each of the three deactivation criteria in Table 4.1 is evaluated. This is done by considering the following for each of the deactivation criteria: (a) the evolution of the geographical distribution (or ‘mask’) of active grid points during the iteration process, (b) the quality of the simulation result (i.e. the degree of its convergence), (c) the iteration curves produced and (d) the speed-up of the simulation that has been realised.

4.2 Case I: High tide with flood current

The first set of results presented is those for the field case recorded on 16/12/2005 at 10:00, which features a high water level and a relatively weak, flood current field. Within this set, the first results considered are those for the DDM using the gradient-based deactivation criteria, indicated as Run 1 in Table 4.1. In Figure 4.1, the grid points that are active during the simulation are indicated by shaded areas, from which it can be seen how the ‘mask’ of active points evolves during the simulation. Snapshots of the active points are shown at iteration levels 9, 19, 29 and 39 - in all cases one iteration before all grid points are re-activated by the governing deactivation regime (at multiples of 5 iterations). From these results it can be seen that, when using the gradient-based deactivation criteria, most grid points are already deactivated after nine iterations. At iteration level 9, only small, isolated clusters of active grid points are still found. These are situated on the ebb tidal delta, the northern coastlines of the barrier islands and on the shallow area in front of the mainland coast. The number of active points is largely diminished after 19 iterations, and virtually all grid points are inactive after 29 iterations.

As will be shown below, the fact that a large number of grid points are deactivated early on in the simulation results in a considerable reduction in simulation time. However, this has a detrimental effect on the accuracy of the simulation results. Figure 4.2 compares the solution obtained from this dynamically deactivated simulation with that of the full simulation (presented in Section 2.3.1), both after 40 iterations. The plots depict the percentage difference in significant wave height and mean period between these two simulations, from which it can be seen where the results of the dynamically deactivated simulation are not converged, or otherwise different from those of the original simulation. In general, it can be seen that with the deactivation criteria used, the results of the dynamic deactivation simulation are at a large variance with the results of the original model. Both the significant wave height and mean period differ by more than 10% relative to the original results in some regions. The largest differences in the significant wave height are found in the lee of the barrier islands, whereas the errors in the mean period are observed both on the northern side of the barrier islands (beaches and ebb tidal delta) and on their lee side.

Next, the performance of the curvature-based deactivation criteria with $\varepsilon_{CH} = \varepsilon_{CT} = 0.005$ (Run 2 in Table 4.1) is considered. Figure 4.3 shows the evolution of active points for this model setting. Compared to the results above, it can be seen that significantly more grid points are active in the earlier stages of the simulation (iteration levels 9 and 19) than above. In particular, at iteration 9 a broad strip of the computational grid is active on the northern side of the barrier islands, including the ebb tidal delta. These grid points remain activated up to iteration 29. In addition, a collection of grid points on the lee side of the barrier islands, where young wind sea is generated, remain active up to iteration 19. Figure 4.4

shows that this increased activity in the computational grid reduces the differences between the dynamically deactivated simulation and the original complete simulation. Compared to Figure 4.2, the differences in the significant wave height in the surf zone of the barrier islands and on the ebb tidal delta have mostly disappeared, and in the lee of the islands the differences have been reduced to less than 10%. Similarly, the large differences in mean wave period on the northern and southern sides of the barrier islands shown in Figure 4.2 are significantly less in the present simulation.

Finally, the performance of the curvature-based deactivation criteria with a stricter parameter of $\varepsilon_{CH} = \varepsilon_{CT} = 0.001$ (Run 3 in Table 4.1) is considered. Figure 4.5 presents the grid point activity for this simulation. Comparison with Figures 4.1 and 4.3 shows that for this deactivation setting a considerably larger number of grid points remain active during the simulation. At iteration 9 grid points are still active over most of the computational domain, albeit with a higher concentration around the barrier islands. By iteration 19 the grid point activity has been reduced to the surf zone and ebb tidal delta of the barrier islands and the general area in the lee of the islands. These regions remain active up to iteration 39, whereas the remaining sub-domains, such as the region offshore of the surf zone and closer toward the mainland coast remain inactive. It is noted that this distribution of active grid cells agrees well with the spatial distribution of the unconverged regions of the computational grid shown in Figures 2.6 and 2.7. Figure 4.6 shows the differences between the results of the dynamically deactivated simulation and that of the original, full simulation. With this setting, the differences between the results of the two approaches have been reduced to less than 2% over virtually the entire computational domain. One notable exception to this result is the shallow region in front of the dike ring on the mainland coast. Here an error in mean period of the order of 5% was found. We note, with reference to Figure 2.4, that this small region corresponds to an isolated region in the current field where the local flow velocities are relatively high compared to those of the surroundings. We will return to this observation below.

To summarise the simulation results of this field case, Figures 4.7 and 4.8 compare the iteration behaviour obtained with the three different deactivation criteria with that of the original, full simulation. These results are presented for the six output locations given in Figure 2.4, with locations 1-3 situated on the northern side of the barrier islands and locations 4-6 to the south of the islands, inside the Wadden Sea. Referring to Figures 2.6 and 2.7, these six locations represent geographical areas where the original, full model requires the most iterations to converge, and hence areas where grid points should, in principle, remain activated. Figures 4.7 and 4.8 show that the iteration curves of Run 1 (gradient-based deactivation criteria) follow those of the full simulation only up to iteration 10 at most. Thereafter, the grid points at all six locations were considered to be sufficiently converged according to the deactivation criteria to be deactivated. These points are reactivated every five iterations to adjust to their surroundings, but mostly return to being deactivated in the iteration thereafter. The result of their early deactivation is that after 40 iterations the solutions at these grid points can be at considerable variance with those of the full simulation, resulting in the errors shown in Figure 4.2 above.

By contrast, the iteration curves of Run 2 and 3 (curvature-based criteria) follow the iteration curves of the full simulation much more closely. For locations 1 to 3 (surf zone of the barrier islands and ebb tidal delta) the iteration curves of Run 2 and 3 follow those of the

full simulation exactly, indicating that these grid points were active during the entire simulation. At locations 4 to 6 (Wadden Sea interior) the agreement between the various iteration curves is not as good, although Run 3, with its stricter deactivation criteria, was better at identifying when convergence had not yet been reached, and hence in tracking the result of the full simulation. For example, at Location 4 in Run 3, the grid point is deactivated at iteration 17. However, the grid point is reactivated at iteration 20 (by the deactivation regime) and remains active up to iteration 26. Another general reactivation follows (iteration 30) after which the grid point remains active up to iteration 37. The effect of this is that at iteration 40 the simulation results of Run 2 and 3 agree much better with the outcome of the original full simulation than do those of Run 1 - as was also seen in Figures 4.2, 4.4 and 4.6 above.

From the results presented above it is clear that Run 3, with the strictest deactivation criteria, yields results that differ the least from those of the original, full simulation. To put these results in perspective, we compare the reduction in simulation time that was obtained by applying the dynamic deactivation method. Figure 4.9 presents a breakdown of the computational time spent on various activities during a simulation, including time spent on the calculation of propagation, source terms and solving the linear system. This breakdown is presented for the original, full simulation and for the reduced simulations of Run 1, 2 and 3. From this comparison it can be seen that Run 1 yields a substantial reduction of 67% in total simulation time relative to the full simulation. However, as was seen above, this large saving in time is offset by a rather large inaccuracy in the simulation result. Run 2 yields an equally impressive reduction of 56% in total simulation time, and this is combined with a much better level of accuracy in the model outcome. Run 3, which was shown above to differ only minimally with the full simulation with respect to the final result, is seen to achieve this result with a significant saving of 43% in total simulation time.

4.3 Case 2: Low tide with ebb current

The second set of results presented is those for the field case recorded on 09/02/2006 at 11:00, which feature a low tide and an ebb current field. Within this set, we again consider the results of the three settings for the deactivation criteria as defined in Table 4.1. As the evaluation will show, the lower water levels and the influence of the ebb current bring out distinctly different aspects of the dynamic deactivation implementation.

Figures 4.10 and 4.11 show the results of Run 1, in which the gradient-based deactivation criteria are applied with the setting given in Tabel 4.1. Figure 4.10 shows that the gradient-based deactivation criteria deactivate almost all grid points early in the simulation, similar to what was found for the high tide case in Section 4.2. By iteration 9 the only significant grid point activity was on the ebb tidal delta, but this too is suspended by iteration 19. Figure 4.11 shows that this early deactivation of grid points leads to large differences between the DDM simulation and the original, full simulation. Differences in significant wave height and mean period exceed 10%, and are found in many regions, including the ebb tidal delta, the surf zones of both the barrier islands and the mainland, and also in the Wadden Sea interior between the tidal gap and the mainland.

Figures 4.12 and 4.13 show the results for Run 2, in which the curvature-based convergence criteria with $\varepsilon_{CH} = \varepsilon_{CT} = 0.005$ are applied. These deactivation criteria allow more grid

point activity, which, at iteration 9, is concentrated in the surf zone of the barrier islands and the ebb tidal delta, and in the lee of the barrier islands. By iteration 19 the grid point activity in the lee of the barrier islands had ceased, and by iteration 29 only the grid points over the ebb tidal delta remained active. The enhanced grid point activity resulting from the stricter deactivation criteria leads to a clear reduction in the differences with the solution of the original, full simulation (Figure 4.13). As can be seen in Figure 4.13, the differences between the results of the reduced and full simulations have been cleared up on both the windward and leeward sides of the barrier islands. However, significant differences remain in the region between the tidal inlet and the mainland, where waves in shallow water encounter adverse currents (compare Figure 2.12).

Figures 4.14 and 4.15 show the simulation results of Run 3, in which the curvature-based deactivation criteria have been applied with the stricter setting of $\varepsilon_{CH} = \varepsilon_{CT} = 0.001$. As was found in Section 4.2 above, this setting results in significantly more grid point activity than the other two alternatives. At iteration 9 most of the computational grid is active, with only the region close to the northern model boundary being deactivated. By iteration 19 the regions of active grid points have been reduced to the barrier island surf zone, the ebb tidal delta and some grid points in the lee of the barrier islands. As before, the gridpoints remain active on the ebb tidal delta up to the end of the simulation at iteration 40. Although this choice of deactivation criteria results in the most grid point activity of the investigated alternatives, it is important to notice that here too the grid points in the region between the tidal gap and the mainland – an important area of non-converge in the result above - are deactivated fairly early on in the simulation. Figure 4.15 shows the difference between the results of this simulation and that of the original, full simulation. Over most regions of the computational grid the difference between the reduced and original simulation results is under 2%. However, in the region between the tidal gap and the mainland, and also in the SW corner of the computational domain, larger differences of up to 5% between the two sets of results remain. As mentioned above, these two regions are characterized by counter current on the wave field. Yet it is interesting to note that the largest differences do not occur inside the main tidal channel where the current is the strongest, but rather over the shoals, where fairly young waves meet the adverse current in very shallow water.

Figures 4.16 and 4.17 give an overview of the iteration behaviour for the three deactivation criteria settings and compare these with the iteration curves of the original, full simulation. The results are given for six output locations defined in Figure 2.12, with locations 1-3 situated on the northern side of the barrier islands and locations 4-6 inside the Wadden Sea. Locations 4-6 are situated in the region where wave-current interaction is the most pronounced (Figure 2.12, lower panel), and specifically on the shoals where the difference between the results of the DDM and full simulations was found to be the greatest (compare Figure 4.15). Figures 4.16 and 4.17 show clear differences between the results of the various deactivation criteria settings, but also between the results for the various output locations. As was seen in Section 4.2, the iteration curves of Run 1 (gradient-based deactivation criteria) depart from the iteration curves of the full simulation in the early stages of the simulation (between iterations 5 and 10). As was seen above, this implies that the deactivation criteria used in Run 1 are not strict enough, since they deactivate grid points prematurely, leading to large errors in comparison to the results of the full simulation. By contrast, the curvature-based deactivation criteria perform well for locations 1-3 situated on the northern side of the barrier islands. This fact is reflected in the small errors in this region

presented in Figure 4.15. However, for locations 4-6 the iteration curves of Run 2 and 3 do not follow those of the full simulation all the way to convergence. Particularly the iteration curves of mean period (Figure 4.17) differ significantly. The convergence rate of the mean period is seen to be quite slow but constant, so that even the stricter of the two curvature-base criteria prematurely consider the grid points converged. The result is that for these locations a discrepancy of about 5% is reached by iteration 40, reflected in the results shown in Figure 4.15.

To clarify these discrepancies further, Figure 4.18 compares the spectra produced by Run 3 to those of the original, full simulation at iteration 40. As was seen in Figure 2.19 above, the spectral shape of the full simulation undergoes significant changes during the iteration process. On the northern side of the barrier islands (locations 1-3) the spectral energy at the peak is reduced and higher harmonics are created. The dynamically deactivated simulation Run 3 reproduces these changes well. By comparison, the spectra at locations 4-6 in the interior of the Wadden Sea undergo a less pronounced change, mostly due to wave-current interaction. In the presence of the counter current, Doppler shifting transports energy towards higher frequencies in the relative frequency domain, resulting in a lower, broader relative frequency spectrum. This change in spectral shape is weakly forced and therefore requires many iterations to realise. As was seen in Figure 4.17, the deactivation criteria applied in Run 3 are not sensitive enough to register the gradual changes at the locations during the iteration process, deactivating the grid points after about 10 iterations. The result is that in the reduced simulation the Doppler shifting has not fully transformed the spectrum yet, causing the differences in the mean frequency observed in Figure 4.15. This finding was confirmed by performing a sensitivity simulation without the current field. Without wave-current interaction over the shoals, convergence is much faster, and the differences between the solutions of the reduced and full simulations are brought back to <2% (results not shown).

Finally, we consider the computational cost of the three investigated dynamically deactivated simulations. Figure 4.19 presents an overview of the computational times of the three model variants and of the original, full simulation. Similar to the results presented in Section 4.2, the largest saving in computational time is achieved in Run 1 (gradient-based criteria), but this is also the least accurate of the three alternatives relative to the original. With stricter deactivation criteria the simulation time increases, but, as shown above, so does the accuracy. Nonetheless, with Run 3, which has the highest accuracy of the three investigated variants, a reduction in total simulation time of 45% relative to the original simulation is achieved.

5 Discussion

This study is the first of two investigations in the context of SBW Waddenzee aimed at reducing the computational time required for stationary SWAN simulations. In the present study it was investigated whether computational time can be saved during the iteration process by omitting some or all of the calculations at grid points that are considered to be converged. To this purpose, the iteration behaviour of SWAN, and in particular the geographical distribution of converged areas during the simulation, was studied for idealised cases and field cases in the Wadden Sea. From this investigation it was found that the bulk of the iteration activity in SWAN occurs in spectral space, and is localised in geographical space. Furthermore, it was found that distinct geographical regions can be identified where many iterations are required for convergence (e.g. surf zones), and others where convergence is reached quickly.

These findings support an approach of reducing simulation time by focusing the iteration activity on those geographical regions where convergence is the slowest. In this investigation it was found that the success of this technique depends strongly on the criteria used for deactivating the individual computational grid points. This choice of deactivation criteria is essentially a balance between the increase in speed achieved by grid point deactivation and the accuracy (level of convergence) retained in the final result. Application of this technique to two field cases recorded in the Wadden Sea shows that with deactivation criteria based on Zijlema and Van der Westhuysen (2005) a reduction in simulation time of about 40% can be achieved, while retaining solutions of significant wave height and mean period T_{m01} that are globally within 2% of those produced by the original model.

For the Wadden Sea situation, an exception to this generally favourable result was found in regions where young wind sea interacts with adverse current over shoals. The young wind sea experiences Doppler shifting to higher relative frequencies, which very gradually decreases the mean period during the iteration process. Since the deactivation criteria applied in this study consider the slowly evolving solution to be converged, grid points are deactivated prematurely here. This was shown to lead to differences (in significant wave height and mean period T_{m01}) in these regions of up to about 5% between the solution of the dynamically deactivated simulation and that of the original, full simulation.

There appears to be two ways in which to improve the performance of the dynamic deactivation method in this situation. Firstly, and most obviously, it may be attempted to apply even stricter deactivation criteria than those used in the present study by using smaller values of ε_{CH} and ε_{CT} . In addition, wave direction could be included as a parameter in the deactivation criteria, based on the findings of Alkyon (2007). Although the accuracy may be improved in the region of the current, the stricter criteria may keep more grid points activated unnecessarily elsewhere, reducing the advantage of the method. A second, more constructive approach is to attempt to improve the iteration behaviour of SWAN in the region of the current field. This could be done by better incorporating Doppler shifting into the second-generation first guess of SWAN. Alternatively, a better starting point for the iteration process – that properly incorporates current influences – can be obtained by applying a third-generation first guess, as is developed in the second part of the present project. A better first estimate of the converged solution will reduce the large number of

iterations required for convergence in this region, and make it easier for the deactivation criteria to identify the point of convergence.

Although in this study the DDM approach has been evaluated for an enforced fixed number of iterations (40) as an objective measure, it can also be applied when simulations are terminated using the convergence criteria of SWAN. When using the convergence criteria to break off the simulation, it appears the most logical to let the DDM deactivate grid points on the basis of the same criteria. Hence, by the stage that, for example, 50% of the grid points are converged, the same percentage of grid points will have been deactivated. The simulation will end when the specified percentage of converged grid points, 98% say, have been reached. Correspondingly, at this point only 2% of the grid points will be active.

When applying the DDM in this way, it can be expected that the difference in model results (e.g. Figures 4.6 and 4.15) and the increase in speed (e.g. Figures 4.9 and 4.19) with respect to the full simulation will change. This would particularly be the case should less strict convergence and deactivation criteria be applied (e.g. the default convergence criteria of SWAN). Full simulations using less strict convergence criteria typically terminate after significantly fewer iterations than the number applied in the present comparison - the default convergence criteria terminate the two Amelanders Zeegat field cases considered here after only 8 and 6 iterations respectively, long before convergence is reached (refer Figures 2.10 and 2.18). The effective increase in speed using the DDM will therefore be less than that demonstrated in this study. Furthermore, since the full simulation will terminate at the same (early) iteration level that most of the grid points in the DDM simulation have been deactivated, the difference between the results of the two sets of simulations will be less. Hence, both the effective difference in model outcomes and the decrease in simulation times are expected to be less than those presented in Section 4. However, this result is misleading since, as pointed out by Zijlema and Van der Westhuysen (2005), the solution of the full simulation can itself contain a significant degree of error due to non-convergence if terminated prematurely. These errors have to be considered in addition to the errors due to the application of the DDM.

When using stricter convergence criteria, such as those proposed by Zijlema and Van der Westhuysen (2005), the full simulation will typically complete more iterations, and reach a higher level of convergence. For example, the curvature-based convergence criteria as defined in this study, using a maximum curvature of $\varepsilon_{CH} = \varepsilon_{CT} = 0.001$ at 99.5% of geographical points, terminate the two Amelanders Zeegat field case simulations after 33 and 26 iterations respectively. The performance of the DDM will therefore resemble the comparison presented in the present study more closely. As shown in Section 2, under these conditions many iterations are performed in which only a few grid points are still active, but still too many to terminate the simulation. It is under these conditions that the DDM is expected to contribute the most to increasing the model speed.

Finally, it is noted that this study was restricted to investigating serial processor computation, and did not consider parallel computing. Parallel computing in SWAN, using either OpenMP or MPI standards, is done by means of stripwise partitioning of the spatial domain (Campbell et al. 2002; Zijlema 2005). When running the standard SWAN model in these parallel modes, wet grid points are distributed equally over the computational nodes, so that the computational load is balanced over the latter. In applying the DDM, the number

of active grid points (a subset of all wet points) is not constant during the simulation, typically being concentrated in certain regions (e.g. surf zones). The load distribution over the parallel nodes could therefore become unequal, leading to low efficiency. It is therefore foreseen that the application of the DDM would require a revision of the implementation of parallel computing in SWAN. One approach could be to treat inactive points as dry points, and to adapt the partition after each (or every few) iteration(s), such that it is balanced again (so-called dynamic load balancing). Such optimisation falls outside the scope of the present study, but should be incorporated in the future development of the DDM.

6 Conclusions and Recommendations

In the present study a method was implemented in SWAN that aims to reduce the computational time of stationary simulations by omitting calculations at converged grid points during the iteration process. This implementation (termed the dynamic deactivation method, or DDM) was successfully evaluated for two field cases recorded at the Amelander Zeegat in the Dutch Wadden Sea that feature current fields (plus one sensitivity run without currents). From the results of this study, and from the discussion presented above, the following conclusions can be drawn:

- The iteration process in SWAN appears to be local in geographical space, since many more iterations are spent in overcoming convergence errors within the wave spectrum at a particular grid point than for these errors to propagate through the computational domain.
- Distinct, coherent regions can be identified in the computational domain where many iterations are required for convergence, and others where convergence is reached quickly. For NW storm conditions at the Amelander Zeegat, the following areas of slow convergence were identified: the surf zones and leeward sides of the barrier islands, the ebb tidal delta and over shoals in the inner Wadden Sea where young wave fields are influenced by moderate currents.
- The success of the dynamic deactivation method was found to depend strongly on the choice of criteria for the deactivation of grid points, and on the overall regime for controlling the activation and deactivation of grid cells.
- For the field cases considered, the best results were obtained with deactivation criteria based on the convergence criteria of Zijlema and Van der Westhuysen (2005), and for a deactivation regime that reactivates all grid points every five iterations. Using these settings, a 40% reduction in simulation time is achieved, with differences in the final solutions of significant wave height and mean period T_{m01} of at most 5% relative to those of the original, full model.
- The largest discrepancies between the results of the dynamically deactivated model version and those of the original are found in areas where wave-current interaction occurs over shallow areas in the Wadden Sea interior. The slow convergence of the Doppler shifting process in these areas causes the deactivation criteria to deactivate grid points prematurely, resulting in non-convergence errors of up to 5%.

Based on the results of the study and the conclusions drawn above, the following recommendations can be made:

- It is recommended that the DDM implementation be evaluated for a wider range of cases in the Wadden Sea, featuring a wider range of conditions, and for other field situations, such as the closed Dutch coast and for deep water.
- Using the larger database of test cases recommended above, the applied deactivation criteria and deactivation regime should be optimized and possibly extended. For

example, based on the findings of Alkyon (2007), wave direction parameters could be included in the deactivation criteria.

- The poor convergence behaviour of grid points in which wave-current interaction occurs should be improved. It is recommended that this be done by improving the first guess used to initialise the iteration process. This may either be done by improving the implementation of the second-generation first-guess currently used in SWAN, or by using the third-generation first guess that is being developed in the context of the current project (reported separately).
- It is recommended to investigate the application of the DDM in combination with the parallel computing facilities of SWAN.

References

- Alkyon (1999). Iteration behaviour of SWAN 40.11 in the Markermeer, Alkyon Report A637, Alkyon Hydraulic Consultancy and Research, October.
- Alkyon (2005). New-processor SWAN, Alkyon Memo A1356ME01, 25 March 2005 (G. Ph. van Vledder).
- Alkyon (2007). Analysis SWAN hindcast tidal inlet of Ameland: Storms of 17 December 2005 and 9 February 2006. Alkyon Report A1725, January 2007.
- Booij, N., R.C. Ris and L.H. Holthuijsen (1999). A third generation wave model for coastal regions, Part I, Model description and validation. *J. Geophys. Res.*, 104, C4, 7649-7666.
- Campbell, T., J.E. Cazes, and W.E. Rogers (2002). Implementation of an Important Wave Model on Parallel Architectures, *Proceedings of MTS/IEEE Oceans 2002 Conference*, 1509.
- Fraza, L.A.J. (1998). Testing the non-stationary option of the SWAN wave model, M.Sc. thesis, Delft University of Technology, Delft, The Netherlands.
- Hersbach, H. and P.A.E.M. Janssen (1999). Improvement of the shortfetch behavior in the wave ocean model (WAM). *J. Atmos. Ocean. Technol.* 16, 884– 892.
- Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vetterling (1993). *Numerical recipes in Fortran 77. The art of scientific computing*, 2nd edition. Cambridge University Press, New York.
- Royal Haskoning (2006). Hindcast Tidal inlet of Ameland. Storms of 17 December 2005 and 9 February 2006. Report Royal Haskoning 9S2639.A0. December 2006.
- Van der Westhuysen, A. J. M. Zijlema, and J. A. Battjes (2005). Improvement of the numerics and deep-water physics in an academic version of SWAN. *Proc. 29th Int. Conf. Coastal Engineering, ASCE*, 855-867.
- Wesseling, P. (1992). *An Introduction to Multigrid Methods*. John Wiley and Sons, Chichester.
- WL (2006). Storm Hindcasts Norderneyer Seegat and Amelander Zeegat. WL | Delft Hydraulics Report H4803.11, August 2006.
- WL (2007). Sensitivity analysis of SWAN for the Amelander Zeegat. WL | Delft Hydraulics Report H4918.41, August 2007.
- Zijlema, M. (2005). Parallelization of a nearshore wind wave model for distributed memory architectures. In: *Parallel Computational Fluid Dynamics|Multidisciplinary Applications*, G. Winter, A. Ecer, J. Periaux, N. Satofuka and P. Fox, Eds. Elsevier.
- Zijlema, M. and A.J. van der Westhuysen (2005). On convergence behaviour and numerical accuracy in stationary SWAN simulations of nearshore wind wave spectra. *Coastal Engng.*, vol. 52, 237-256.

A Appendix

This section details the implementation of the dynamical deactivation method into the SWAN code. These additions to the code are all localized in the program file 'swancom1.ftn', and have been identified with the version number 40.XX.

A.1 Deactivation regime and run control

In the subprogram SWCOMP the following additional code is implemented:

a) The following variable declarations are made:

```

!   4. Argument variables
!
!   SACC0      mean wave frequency at iter-2                40.XX
!   TMDIFC     difference of Tmean(i) - Tmean(i-2) meant for 40.XX
!               computation of curvature of Tmean           40.XX
!   SLEEP      Array to register which geo grid points on the 40.XX
!               computational grid are deactivated (sleeping) 40.XX
!   NUMSLP     Total number of deactivated (sleeping) geo grid 40.XX
!               points during a given iteration              40.XX
!
REAL, DIMENSION(:), ALLOCATABLE :: SACC0, TMDIFC           40.XX
LOGICAL :: SLEEP(MCGRD)                                    40.XX
REAL :: NUMSLP                                             40.XX

```

b) The following allocations are made:

```

ALLOCATE(SACC0(MCGRD))                                     40.XX
ALLOCATE(TMDIFC(MCGRD))                                   40.XX

```

c) The following initializations are made:

```

SACC0 = 0.                                                40.XX
TMDIFC= 0.                                                40.XX
SLEEP(:) = .FALSE.                                       40.XX

```

d) The global deactivation regime is implemented at the start of the main iteration loop in SWCOMP:

```

DO 450 ITER = 1, ITERMX                                    30.00

!--- Control the number of inactive geo points for this iteration 40.XX
    IF ((ITER.LE.5).OR.(MOD(ITER,5).EQ.0)) SLEEP(:)=.FALSE. 40.XX

!--- Spatial plots of sleeping points                          40.XX
    WRITE(PRTEST,'(A9,I4)') 'Iteration',ITER                 40.XX
    DO IX = 1, MXC                                           40.XX
        DO IY = 1, MYC                                       40.XX
            IF (KGRPNT(IX,IY).LE.1 .OR.                      40.XX
&             COMPDA(KGRPNT(IX,IY),JDP2).LE.DEPMIN) THEN 40.XX
                SLEEP(KGRPNT(IX,IY))=.TRUE.                 40.XX
            ENDIF                                             40.XX
            IF ( .NOT.SLEEP(KGRPNT(IX,IY)) ) THEN            40.XX
                WRITE(PRTEST,'(2F10.2,I5)') XCGRID(IX,IY),YCGRID(IX,IY),1 40.XX
            ENDIF                                             40.XX
        END DO                                               40.XX
    END DO                                                    40.XX

```

```
!--- Statistics of sleeping points                                40.XX
    NUMSLP = 0.                                                40.XX
    DO IP = 2, MCGRD                                           40.XX
        IF (SLEEP(IP)) NUMSLP = NUMSLP+1.                    40.XX
    END DO                                                       40.XX
    WRITE(PRINTF,*) '% sleeping points:',NUMSLP/(MCGRD-1)*100. 40.XX
```

e) During the looping through the geographical grid points in SWCOMP, the variable SLEEP is passed through to the subroutine SWOMPU as an argument variable:

```
CALL SWOMPU (SWPDIR, KSX, KSY), [...omitted variables...], SLEEP) 40.XX

SUBROUTINE SWOMPU (SWPDIR, KSX, KSY), [...omitted variables...], SLEEP) 40.XX
! 4. Argument variables
LOGICAL :: SLEEP(MCGRD)                                          40.XX
```

f) Then, at the start of the computational section of SWOMPU, all calculations inside SWOMPU (calculation of propagation, source terms and matrix solving, etc.) are discontinued for the current iteration if a grid point is deactivated. The program control is returned to the subroutine SWCOMP. This is done in analogy to the method followed for dry points².

```
IF (SLEEP(KCGRD(1))) RETURN                                     40.XX
```

The decision whether a grid point is deactivated is taken at the end of the iteration (after completion of four sweeps), and is taken in the same subroutines where the convergence of the simulation is tested. The required changes to these subroutines are presented in the next section.

A.2 Deactivation criteria

The deactivation of grid points occurs in the same subroutines where the degree of convergence of the simulation is computed. Based on the choice of the user, either the gradient-based criteria (PNUMS(21)=0) or the curvature-based criteria (PNUMS(21)=1) are used. In both cases the variable SLEEP is passed through in the argument list:

```
IF (PNUMS(21).EQ.0.) THEN                                     40.41
    CALL SACCUR (COMPDA(1,JDP2), [...omitted variables...], SLEEP) 40.XX
ELSE IF (PNUMS(21).EQ.1.) THEN                               40.41
    CALL SWSTPC ( HSAC0, HSAC1, HSAC2 ), [...omitted variables...], 40.41
&    SACC0, TMDIFC, SLEEP)                                  40.XX
END IF                                                        40.41
```

a) Inside the routine SACCUR the following additions are made:

```
SUBROUTINE SACCUR (DEP2, [...omitted variables...], SLEEP) 40.XX

! 4. Argument variables
LOGICAL :: SLEEP(MCGRD)                                          40.XX
```

2. Alternatively, the call to the subroutine SWOMPU can be omitted entirely, with similar effect.

```

!           *** gridpoint in which mean period and wave height ***
!           *** have reached required accuracy ***

IF ( (TMREL .LE. PNUMS(1) .OR. TMOVAL .LE. PNUMS(16)) .AND.           30.82
&    (HSREL .LE. PNUMS(1) .OR. HSOVAL .LE. PNUMS(15)) ) THEN       30.82
    IACCURt = IACCURt + 1                                           40.31
    SLEEP(INDX) = .TRUE.                                           40.XX
END IF                                                                30.82

```

b) Inside the subroutine SWSTPC, a number of changes are made, pertaining to the addition of the curvature of the mean period and the deactivation of grid points. The complete, adapted subroutine SWSTPC is given below.

```

!*****
!
!       SUBROUTINE SWSTPC ( HSACC0, HSACC1, HSACC2, SACC1 , SACC2,
&                          HSDIFC, DELHS , DELTM , DEP2 , ACCUR ,
&                          I1MYC , I2MYC ,
&                          SACC0 , TMDIFC, SLEEP)                    40.XX
!*****
!
!       USE OCPCOMM4                                                40.41
!       USE SWCOMM3                                                40.41
!       USE SWCOMM4                                                40.41
!       USE M_GENARR
!       USE M_PARALL
!
!       IMPLICIT NONE
!
!       0. Authors
!
!       40.41: Andre van der Westhuysen
!       40.41: Marcel Zijlema
!       40.XX: Andre van der Westhuysen
!
!       1. Updates
!
!       40.41, Jun. 04: New subroutine
!       40.41, Oct. 04: common blocks replaced by modules, include files removed
!
!       2. Purpose
!
!       Check convergence based on the relative, absolute
!       and curvature values of significant wave height
!
!       4. Argument variables
!
!       ACCUR           indicates percentage of grid points in
!                       which accuracy is reached
!       DELHS           difference in Hs between last 2 iterations
!       DELTM           difference in Tm01 between last 2 iterations
!       DEP2            depth
!       HSACC0          significant wave height at iter-2
!       HSACC1          significant wave height at iter-1
!       HSACC2          significant wave height at iter
!       HSDIFC          difference of Hs(i) - Hs(i-2) meant for
!                       computation of curvature of Hs
!       I1MYC           lower index for thread loop over y-grid row
!       I2MYC           upper index for thread loop over y-grid row
!       SACC1           mean wave frequency at iter-1
!       SACC2           mean wave frequency at iter
!
!       INTEGER I1MYC, I2MYC
!       REAL ACCUR
!       REAL DEP2(MCGRD) ,
&       HSACC0(MCGRD) ,
&       HSACC1(MCGRD) ,
&       HSACC2(MCGRD) ,
&       SACC1(MCGRD) ,
&       SACC2(MCGRD) ,

```

```

&      DELHS(MCGRD)      ,
&      DELTM(MCGRD)     ,
&      HSDIFC(MCGRD)    ,
REAL   SACC0(MCGRD)     ,      40.XX
&      TMDIFC(MCGRD)    ,      40.XX
LOGICAL SLEEP(MCGRD)   ,      40.XX
!
! 6. Local variables
!
!   ACS2 : auxiliary variable
!   ACS3 : auxiliary variable
!   HSABS : absolute value of Hs
!   HSCURV: curvature value of Hs
!   HSDIFO: previous value of HSDIFC
!   HSREL : relative value of Hs
!   IACCUR: indicates number of grid points in which
!           accuracy is reached
!   IARR : auxiliary array meant for global reduction
!   ID : counter of direction
!   IENT : number of entries
!   II : loop variable
!   INDX : index for indirect address
!   IS : counter of frequency
!   IX : loop counter
!   IX1 : lower index in x-direction
!   IX2 : upper index in x-direction
!   IY : loop counter
!   IY1 : lower index in y-direction
!   IY2 : upper index in y-direction
!   LHEAD : logical indicating to write header
!   TMABS : absolute value of Tm01
!   TSTFL : indicates whether grid point is a test point
!   WETGRD : number of wet grid points
!   XMOM0 : zeroth moment
!   XMOM1 : first moment
!   SACC0 : mean wave frequency at iter-2      40.XX
!   TMDIFC: difference of Tmean(i) - Tmean(i-2) meant for
!           computation of curvature of Tmean  40.XX
!   TMCURV: curvature value of TM             40.XX
!   TMDIFO: previous value of TMDIFC         40.XX
!   TMREL : relative value of TM             40.XX
!
INTEGER ID, IS, IENT, II, INDX, IX, IY, IX1, IX2, IY1, IY2
INTEGER IACCUR, WETGRD, IACCURt, WETGRDt, IARR(2)
REAL   ACS2, ACS3, HSREL, HSABS, HSCURV, HSDIFO, TMABS,
&      XMOM0, XMOM1
LOGICAL LHEAD, TSTFL
REAL   TMREL, TMCURV, TMDIFO      40.XX
!
! 7. Common blocks used
!
COMMON/SWSTPC_MT_COM/WETGRD,IACCUR
!
SWSTPC_MT_COM place local summed variables WETGRD and IACCUR
!           in common block so they will be scoped as shared
!
! 8. Subroutines used
!
EQREAL      Boolean function which compares two REAL values
STRACE      Tracing routine for debugging
STPNOW      Logical indicating whether program must
!           terminated or not
SWREDUCE    Performs a global reduction
!
LOGICAL EQREAL, STPNOW
!
! 9. Subroutines calling
!
SWCOMP (in SWANCOM1)
!
! 12. Structure
!
master thread initialize the shared variables
! store Hs and Tm as old values and count number of wet grid points
! compute new values of Hs and Tm

```

```

!      calculate a set of accuracy parameters based on relative,
!      absolute and curvature values of Hs and check accuracy
!      global sum of IACCUR and WETGRD
!      carry out reductions across all nodes
!
! 13. Source text

      SAVE IENT
      DATA IENT/0/
      IF (LTRACE) CALL STRACE (IENT,'SWSTPC')

!      --- master thread initialize the shared variables
!$OMP MASTER
      WETGRD = 0
      IACCUR = 0
!$OMP END MASTER
!$OMP BARRIER

      IF ( LMXF ) THEN
          IX1 = 1
      ELSE
          IX1 = 1+IHALOX
      END IF
      IF ( LMXL ) THEN
          IX2 = MXC
      ELSE
          IX2 = MXC-IHALOX
      END IF
      IF ( LMYF ) THEN
          IY1 = I1MYC
      ELSE
          IY1 = 1+IHALOY
      END IF
      IF ( LMYL ) THEN
          IY2 = I2MYC
      ELSE
          IY2 = MYC-IHALOY
      END IF

!      --- store Hs and Tm as old values and count number of wet grid points

      WETGRDt = 0
      DO IX = IX1, IX2
          DO IY = IY1, IY2
              INDX = KGRPNT(IX,IY)
              IF ( DEP2(INDX).GT.DEPMIN ) THEN
                  HSACC0(INDX) = MAX( 1.E-20 , HSACC1(INDX) )
                  HSACC1(INDX) = MAX( 1.E-20 , HSACC2(INDX) )
                  SACC0 (INDX) = MAX( 1.E-20 , SACC1 (INDX) )
                  SACC1 (INDX) = MAX( 1.E-20 , SACC2 (INDX) )
                  WETGRDt = WETGRDt + 1
              ELSE
                  HSACC0(INDX) = 0.
                  HSACC1(INDX) = 0.
                  SACC0 (INDX) = 0.
                  SACC1 (INDX) = 0.
              END IF
          END DO
      END DO

!      --- compute new values of Hs and Tm

      DO IX = IX1, IX2
          DO IY = IY1, IY2
              INDX = KGRPNT(IX,IY)

              IF ( DEP2(INDX).GT.DEPMIN ) THEN

                  XMOM0 = 0.
                  XMOM1 = 0.
                  DO IS = 1, MSC
                      DO ID = 1, MDC
                          ACS2 = SPCSIG(IS)**2 * AC2(ID,IS,INDX)
                          ACS3 = SPCSIG(IS) * ACS2
                          XMOM0 = XMOM0 + ACS2
                      END DO
                  END DO
              END IF
          END DO
      END DO

```

```

        XMOM1 = XMOM1 + ACS3
    END DO
END DO
XMOM0 = XMOM0 * FRINTF * DDIR
XMOM1 = XMOM1 * FRINTF * DDIR

IF ( XMOM0.GT.0. ) THEN
    HSACC2(INDX) = MAX ( 1.E-20 , 4.*SQRT(XMOM0) )
    SACC2 (INDX) = MAX ( 1.E-20 , (XMOM1/XMOM0) )
ELSE
    HSACC2(INDX) = 1.E-20
    SACC2 (INDX) = 1.E-20
END IF

END IF

END DO
END DO

IACCURt = 0

! --- calculate a set of accuracy parameters based on relative,
! absolute and curvature values of Hs and check accuracy

LHEAD=.TRUE.
DO IX = IX1, IX2
    DO IY = IY1, IY2
        INDX = KGRPNT(IX,IY)

! --- determine whether the point is a test point

        TSTFL = .FALSE.
        IF (NPTST.GT.0) THEN
            DO 20 II = 1, NPTST
                IF (IX.NE.XYTST(2*II-1)) GOTO 20
                IF (IY.NE.XYTST(2*II )) GOTO 20
                TSTFL = .TRUE.
            20 CONTINUE
        END IF

        DELHS(INDX) = 0.0
        DELTM(INDX) = 0.0
        IF ( DEP2(INDX).GT.DEPMIN ) THEN

            HSABS = ABS ( HSACC2(INDX) - HSACC1(INDX) )
            HSREL = HSABS / HSACC2(INDX)
            TMABS = ABS ( (PI2/SACC2(INDX)) - (PI2/SACC1(INDX)) )
            TMREL = TMABS / SACC2(INDX)                                40.XX

            HSDIFO = HSDIFC(INDX)
            HSDIFC(INDX) = 0.5*( HSACC2(INDX) - HSACC0(INDX) )
            HSCURV = ABS(HSDIFC(INDX) - HSDIFO)/HSACC2(INDX)

            TMDIFO = TMDIFC(INDX)                                40.XX
            TMDIFC(INDX) = 0.5*( SACC2(INDX) - SACC0(INDX) )    40.XX
            TMCURV = ABS(TMDIFC(INDX) - TMDIFO)/SACC2(INDX)    40.XX

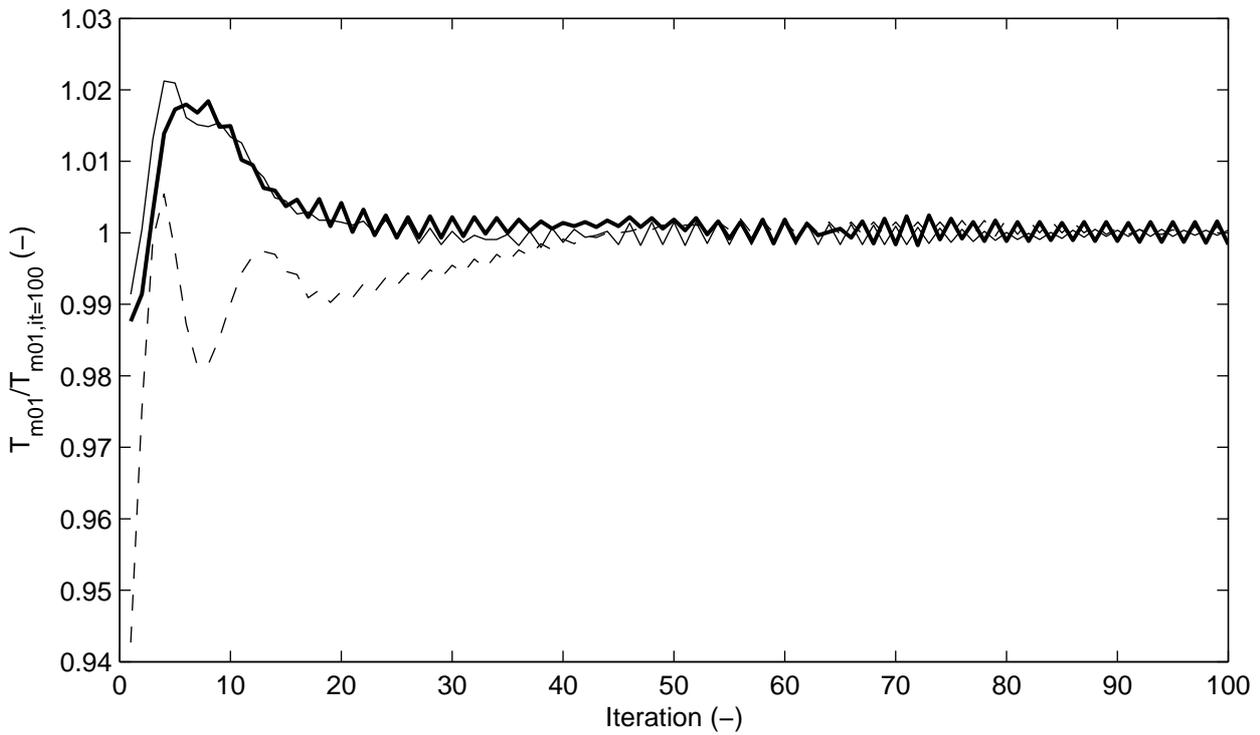
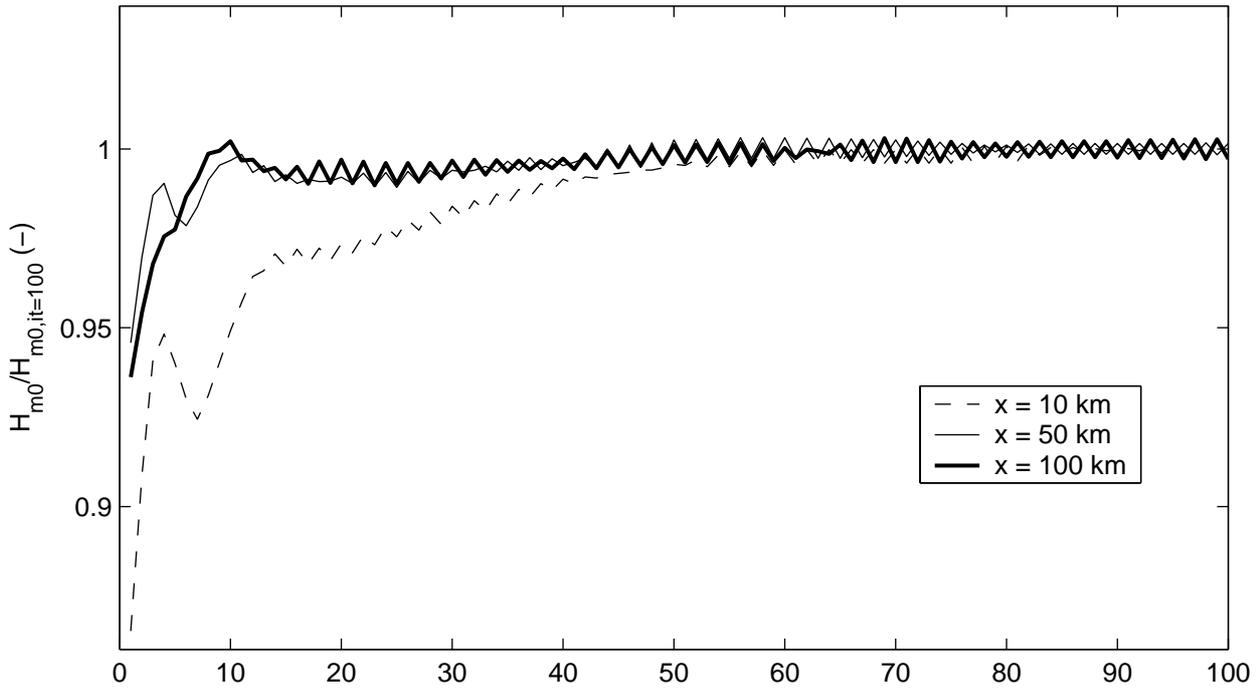
            DELHS(INDX) = HSABS
            IF (EQREAL(SACC1(INDX),1.E-20) .OR.
                & EQREAL(SACC2(INDX),1.E-20) ) THEN
                DELTM(INDX) = 0.
            ELSE
                DELTM(INDX) = TMABS
            END IF

! --- add gridpoint in which wave height has reached
! required accuracy

            IF ( (HSCURV.LE.PNUMS(15) .AND.                                40.XX
                & (HSREL.LE.PNUMS(1) .OR. HSABS.LE.PNUMS(2))) .AND.    40.XX
                & (TMCURV.LE.PNUMS(15) .AND.                                40.XX
                & (TMREL.LE.PNUMS(1) .OR. TMABS.LE.PNUMS(2))) ) THEN    40.XX
                IACCURt = IACCURt + 1
                SLEEP(INDX) = .TRUE.                                    40.XX
            END IF
        END IF
    END DO
END DO

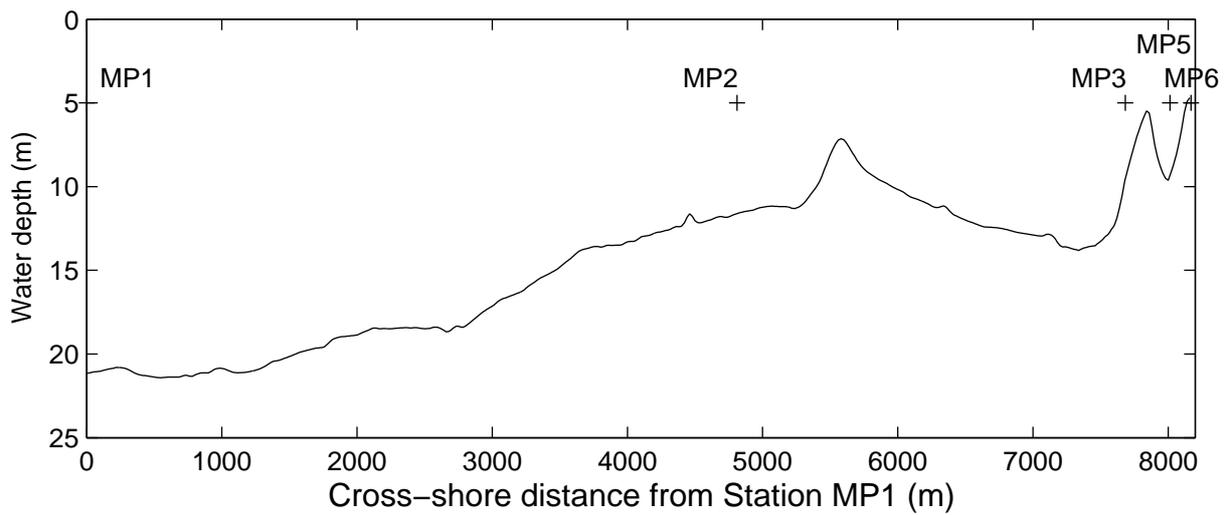
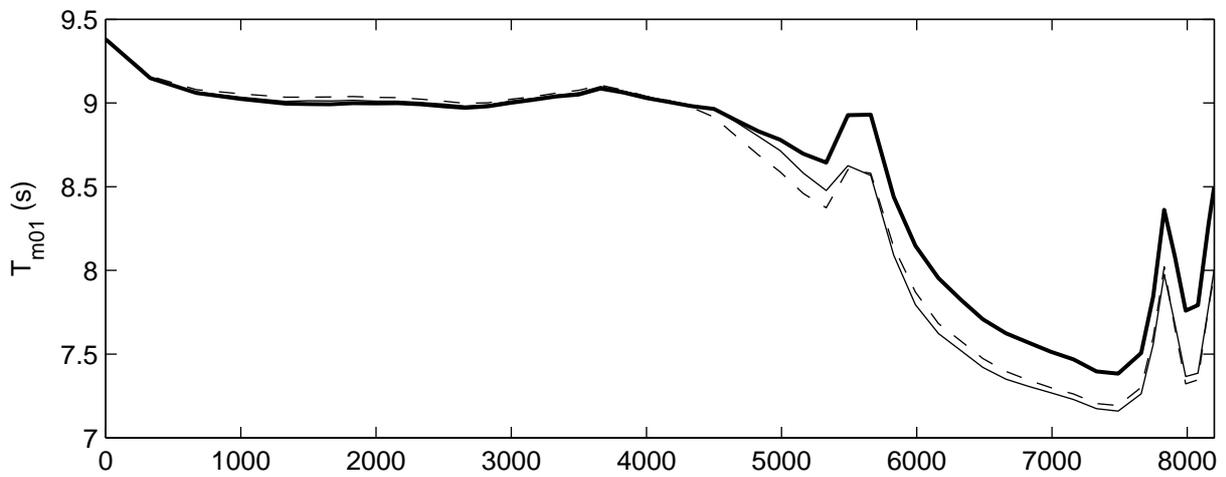
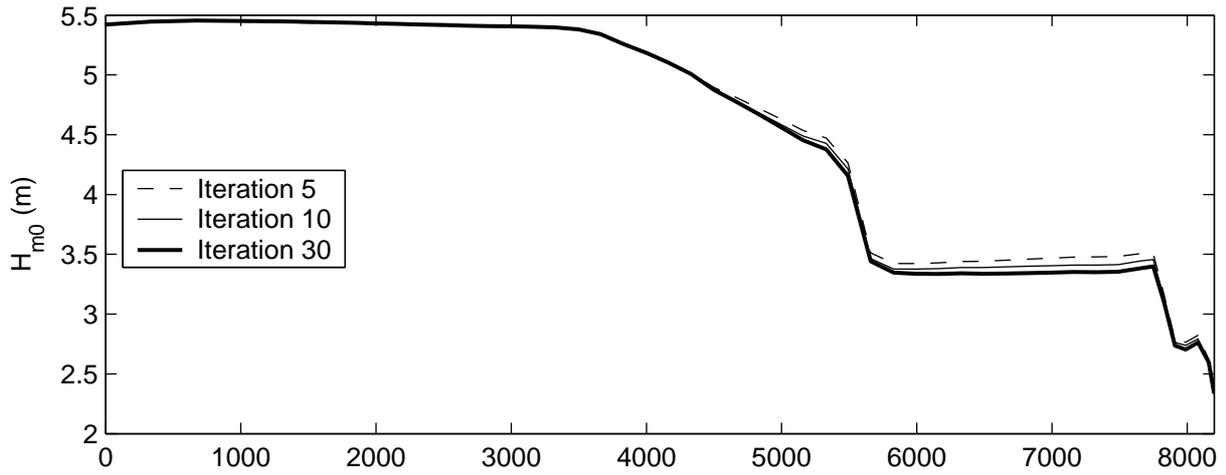
```


Figures

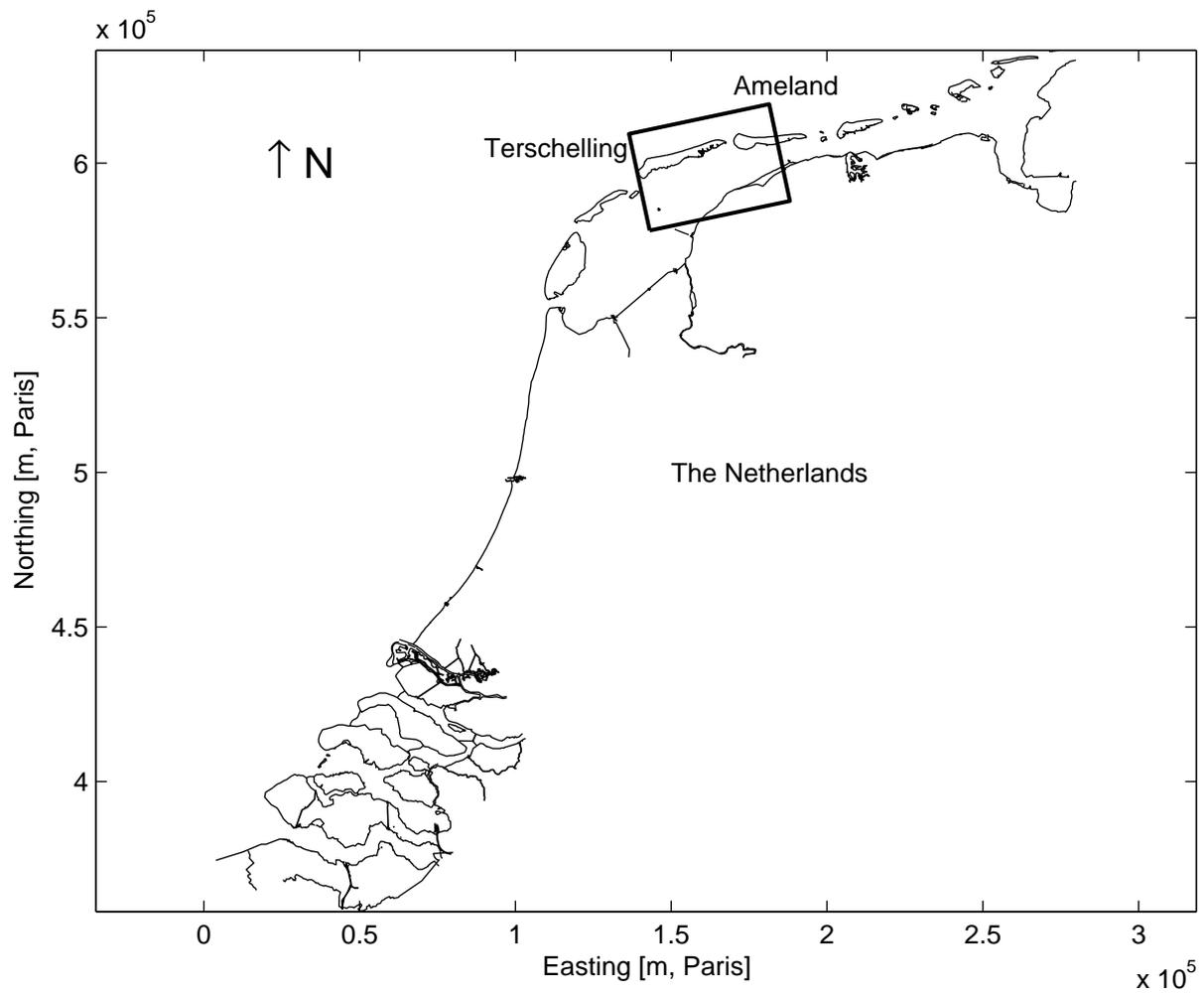


Deep water fetch-limited wave growth over a 100 km fetch
 Normalised iteration curves at various locations along the fetch

Delft3D-WAVE



| | | |
|---|--------------|----------|
| Petten Sea defense: Storm of 02/01/1995 at 06:00 Evolution of significant wave height and mean period during iteration process | | |
| | Delft3D-WAVE | |
| WL DELFT HYDRAULICS | H4918.37 | Fig. 2.2 |



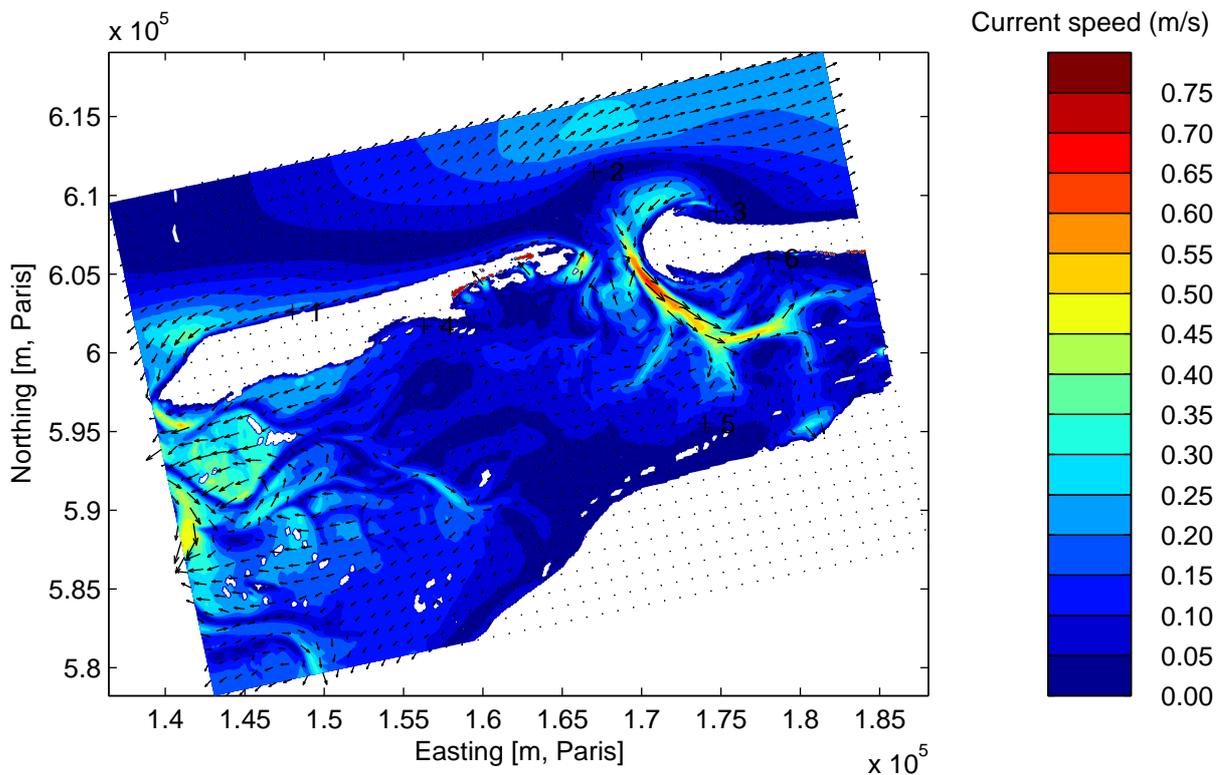
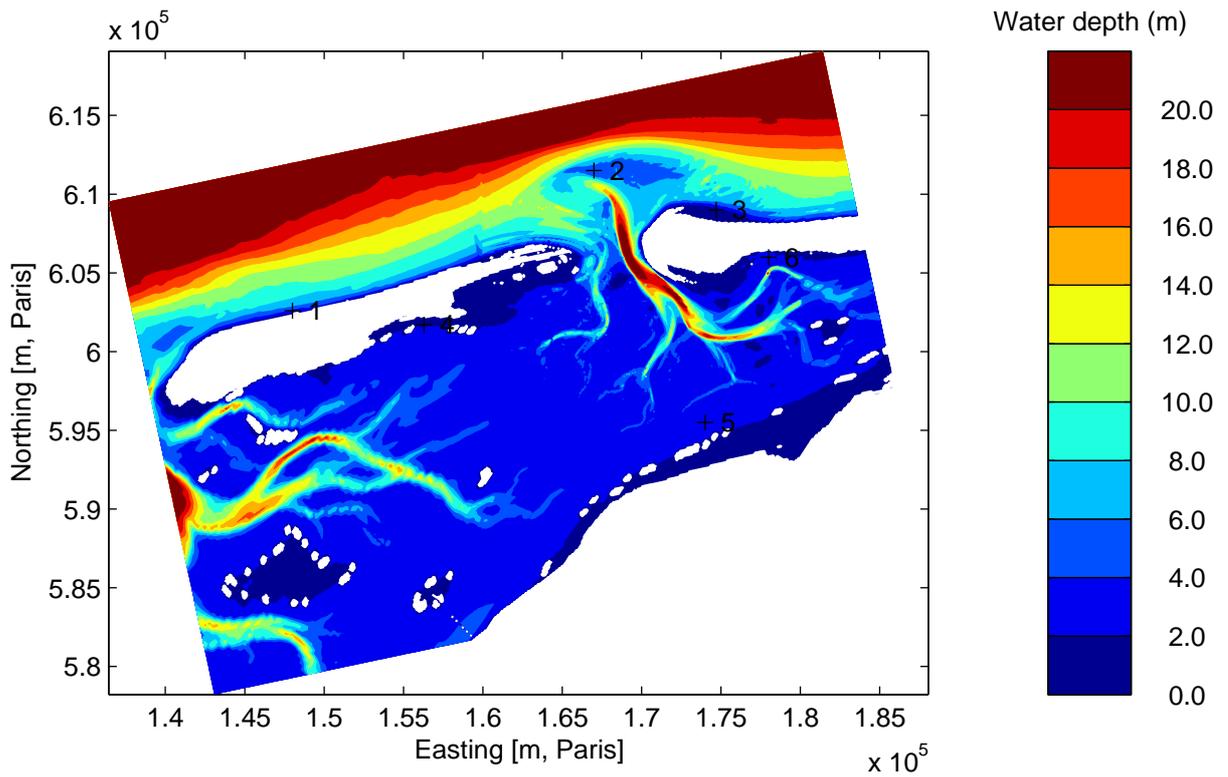
Location map of the Amelander Zeegat model area in the Dutch Wadden Sea

Delft3D-WAVE

WL | DELFT HYDRAULICS

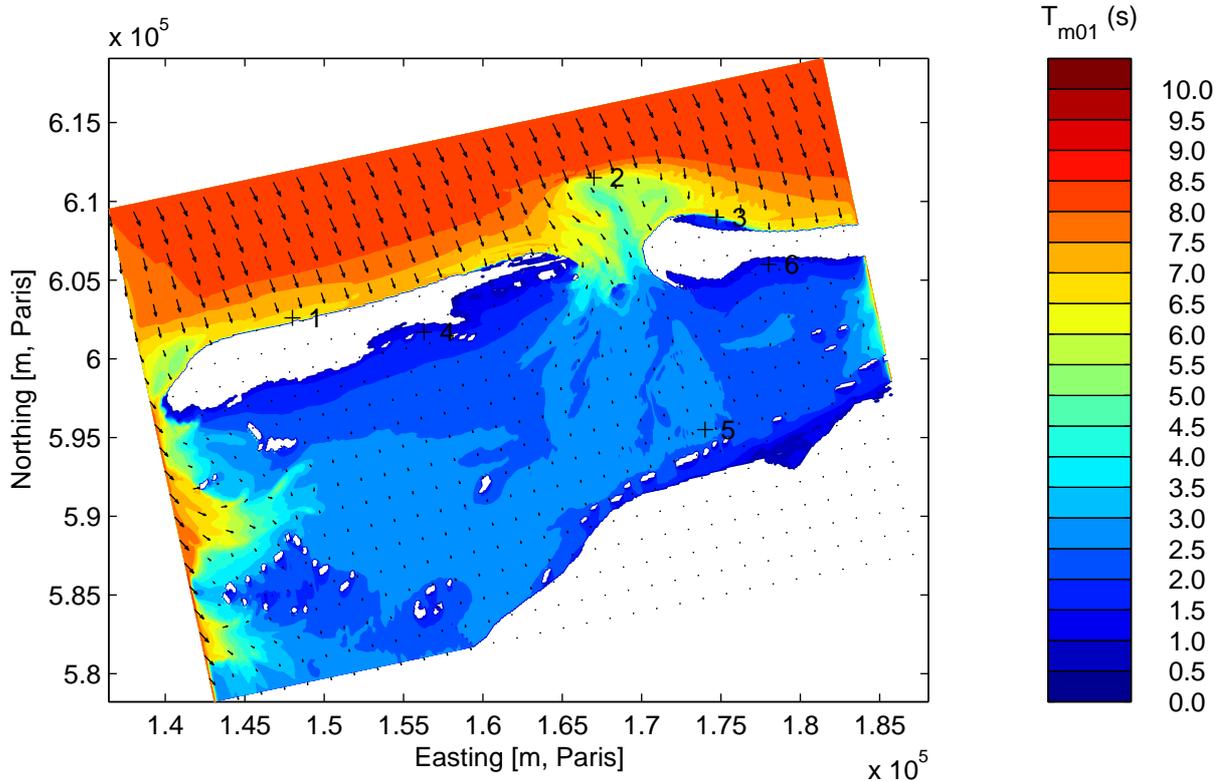
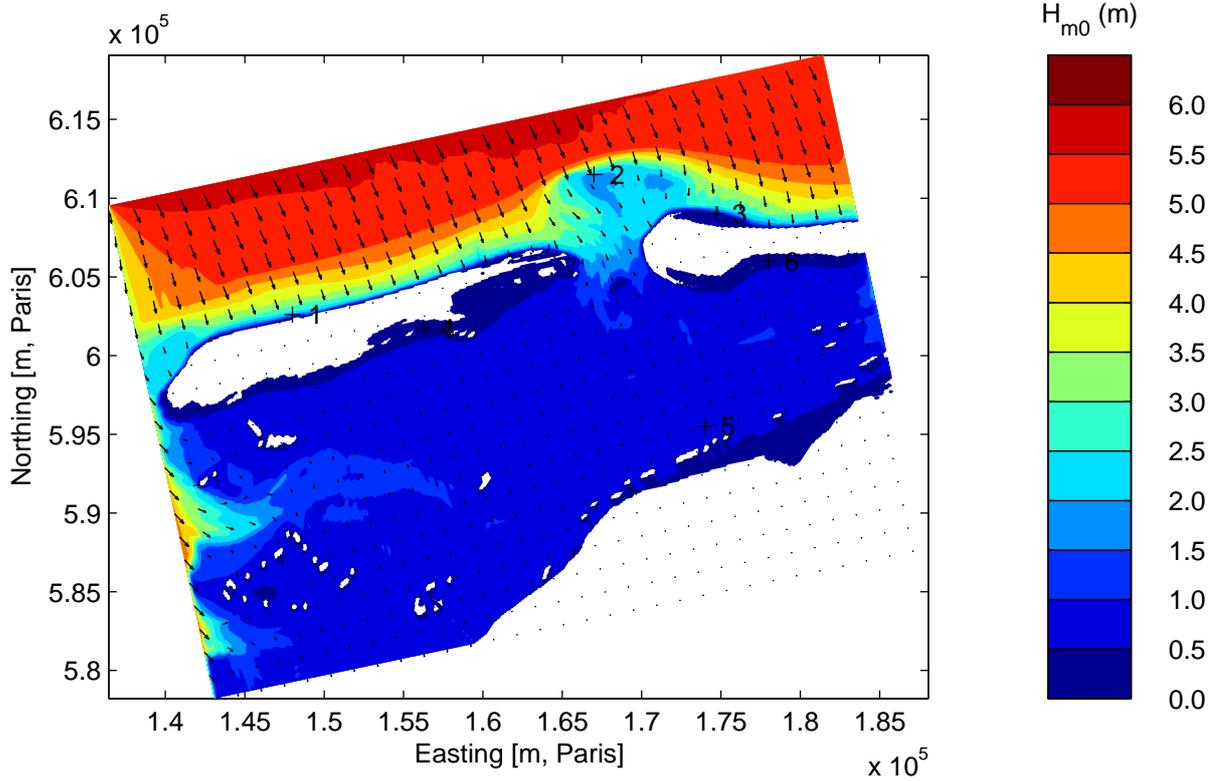
H4918.37

Fig. 2.3



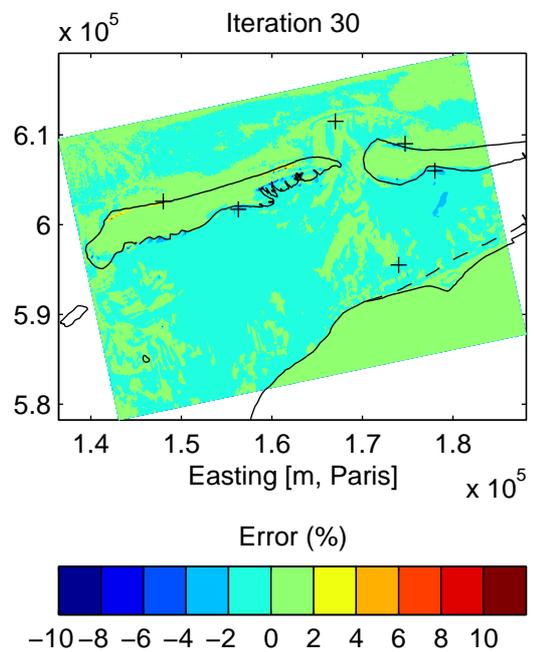
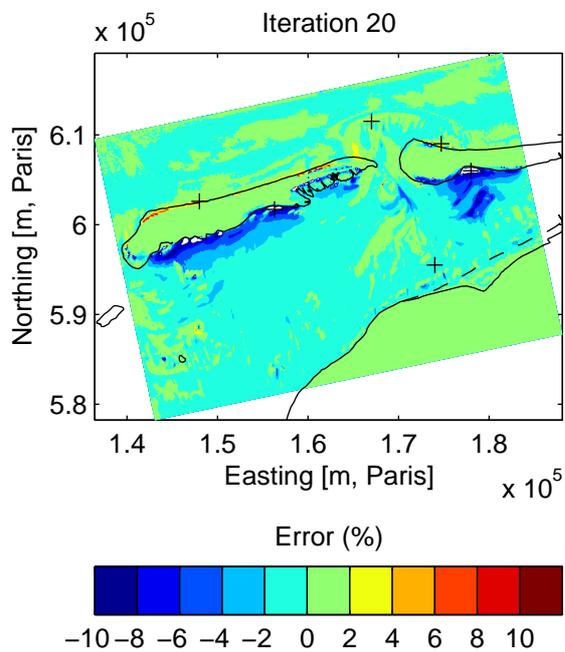
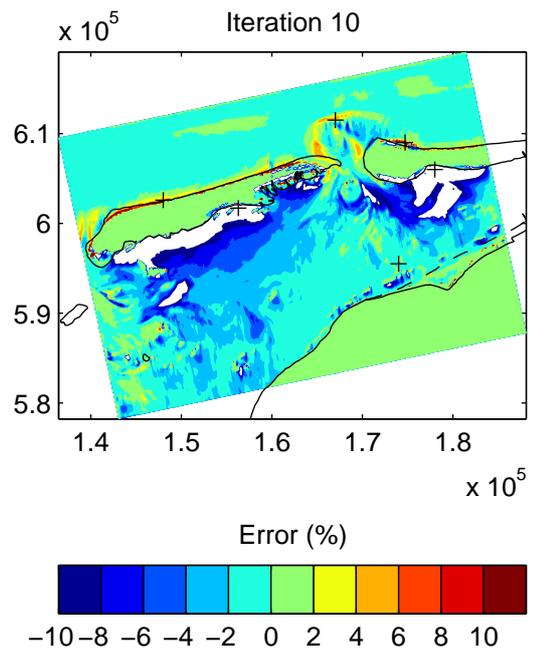
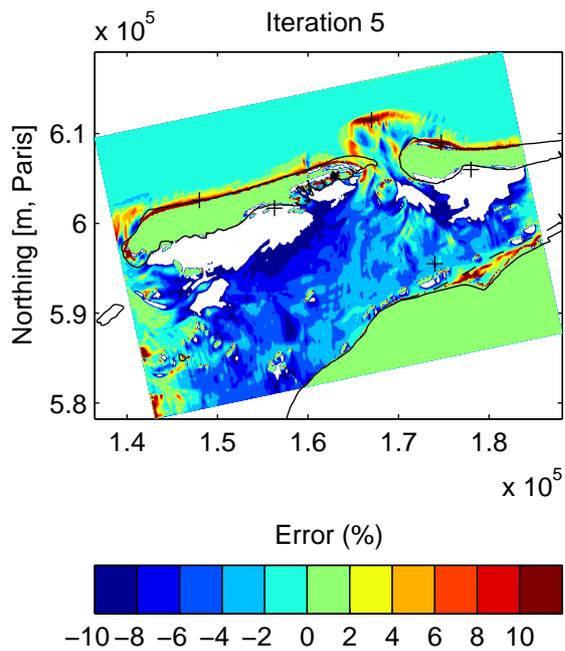
Amelander Zeegat storm of 16/12/2005 at 10:00
Water depth (above) and current field (below).

Delft3D-WAVE



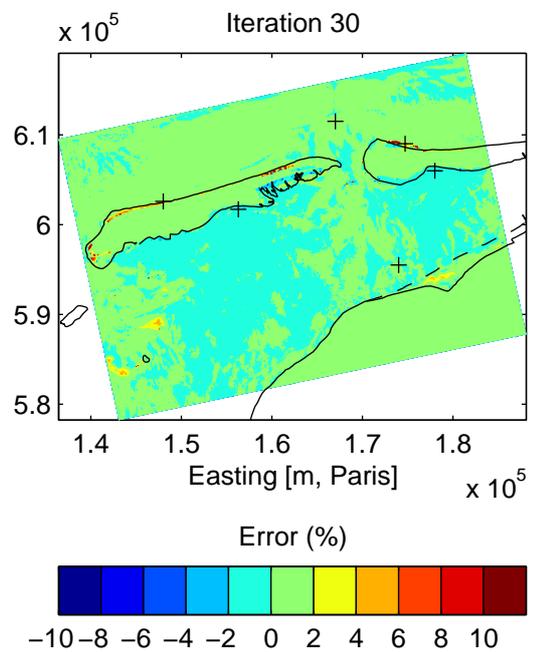
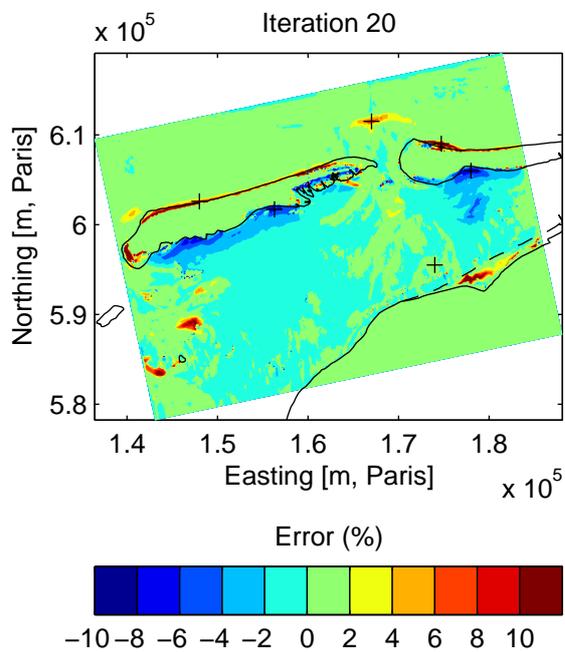
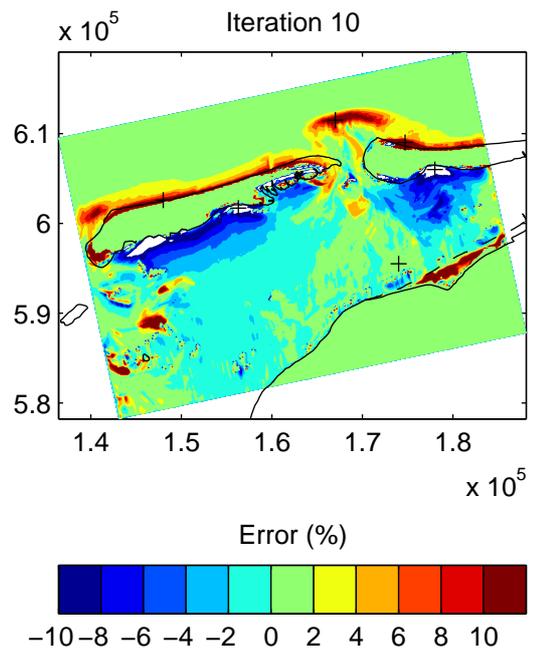
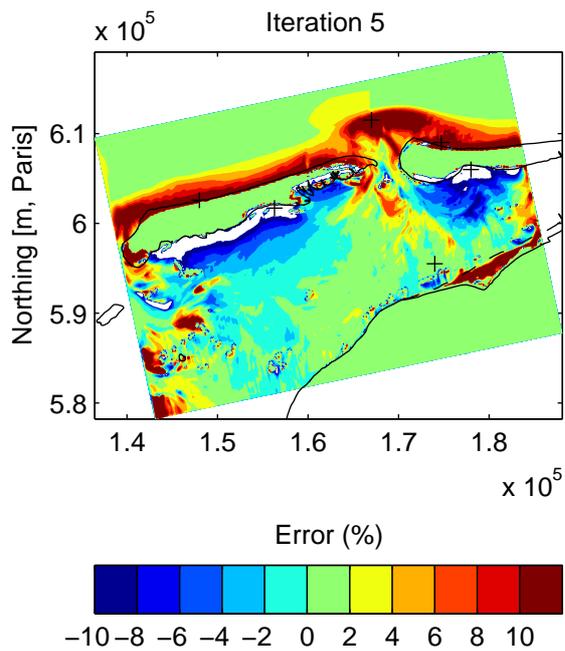
Amelande Zeegat storm of 16/12/2005 at 10:00
 Significant wave height (above) and mean period (below).

Delft3D-WAVE



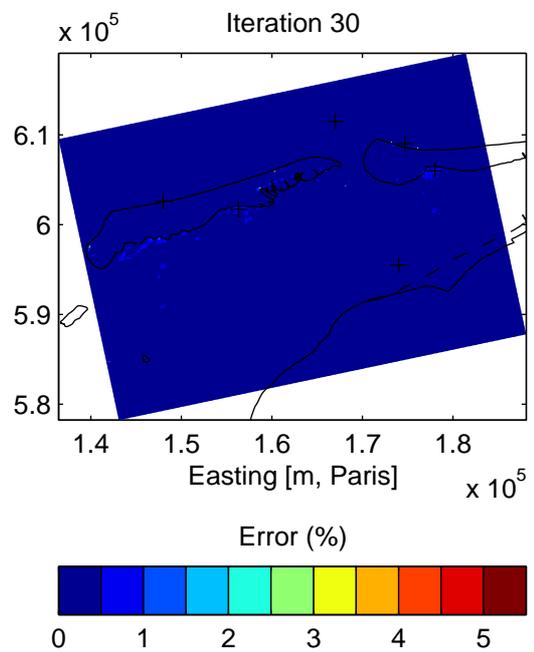
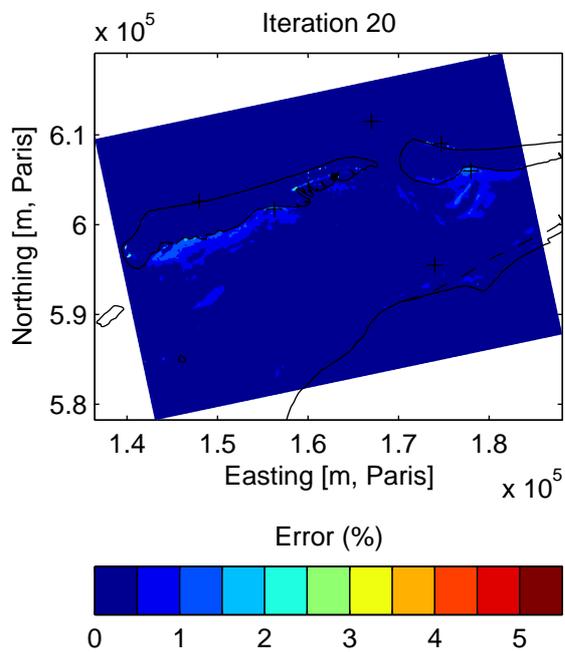
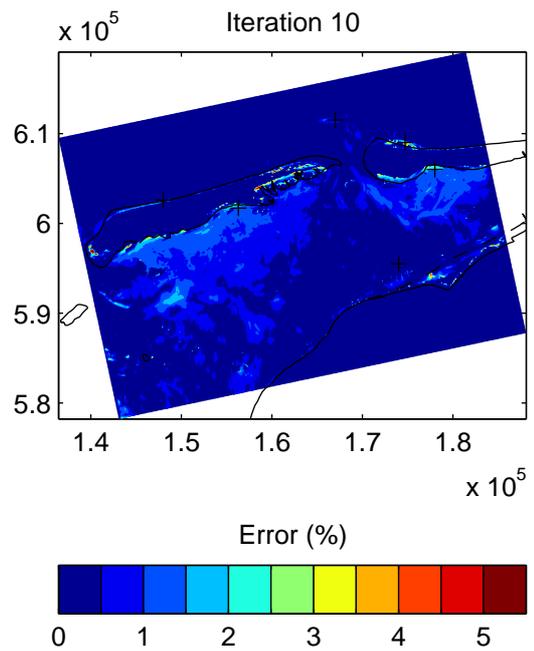
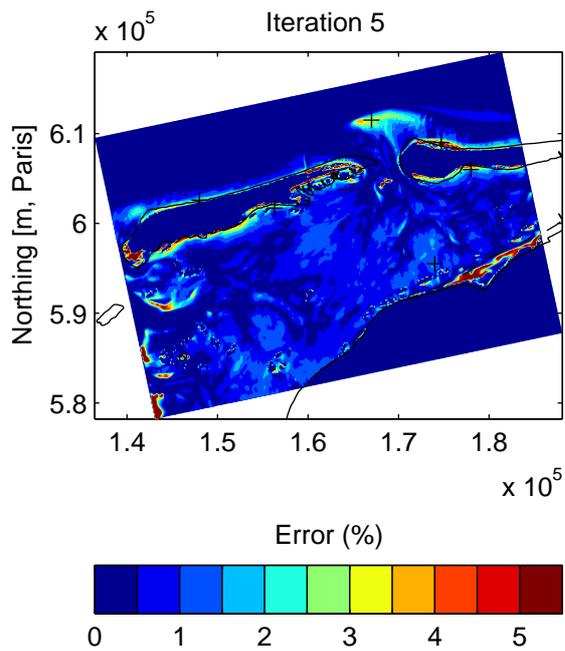
Amelander Zeegat storm of 16/12/2005 at 10:00
 Global convergence error in wave height: $(H_{m0,s} - H_{m0,s=40})/H_{m0,s=40}$

Delft3D-WAVE



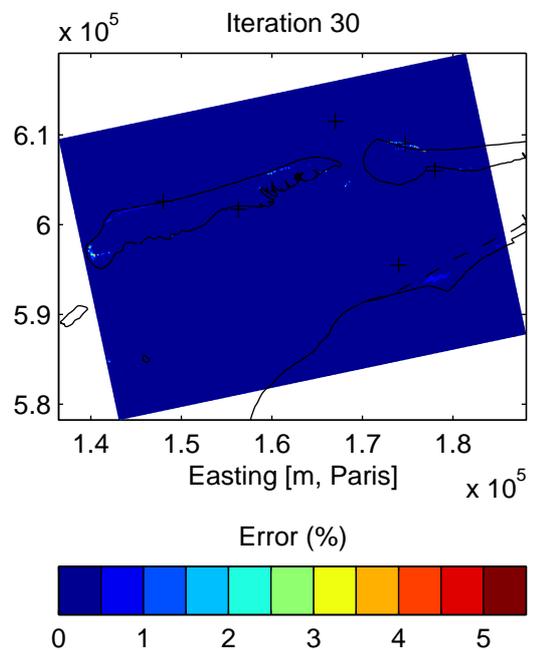
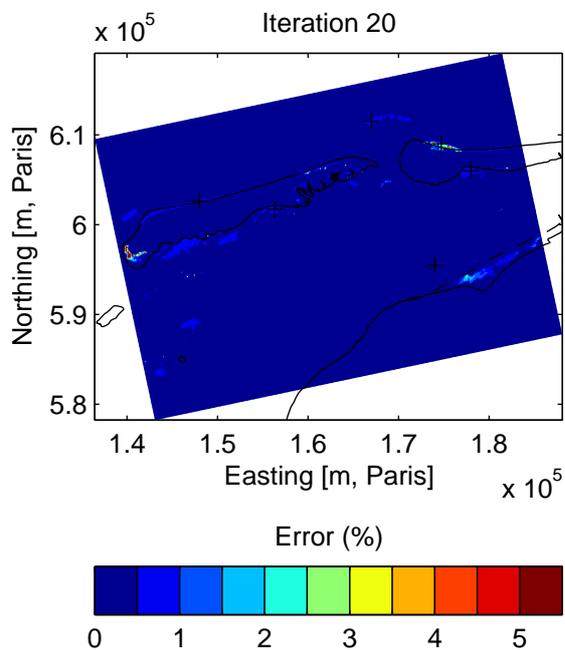
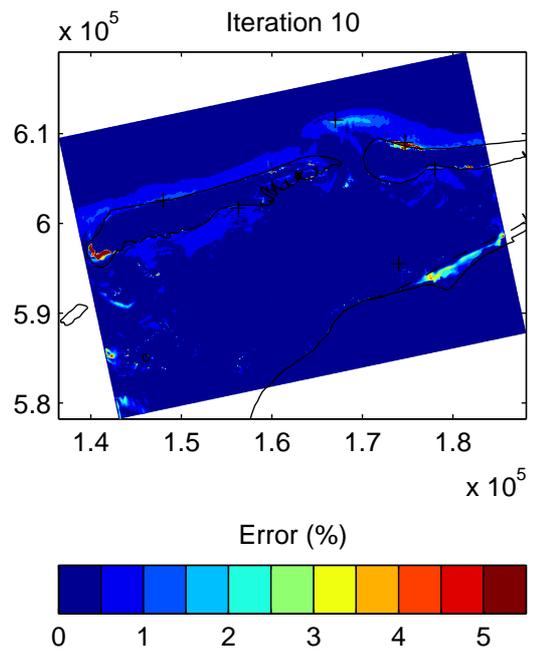
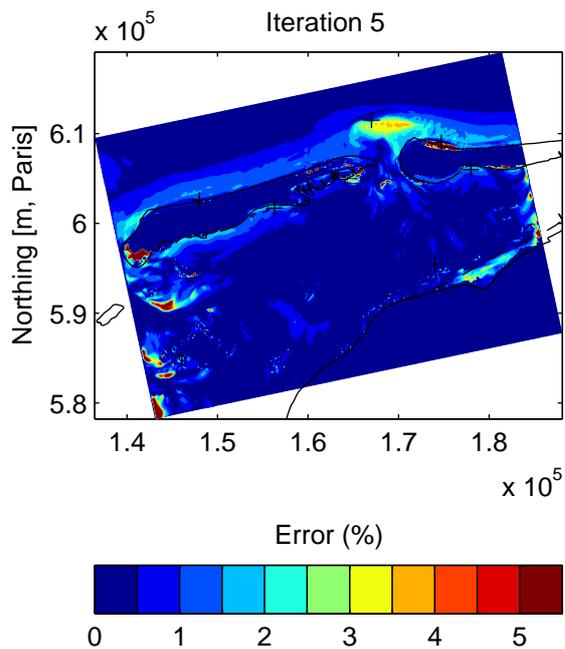
Amelander Zeegat storm of 16/12/2005 at 10:00
 Global convergence error in mean period: $(T_{m01,s} - T_{m01,s=40})/T_{m01,s=40}$

Delft3D-WAVE



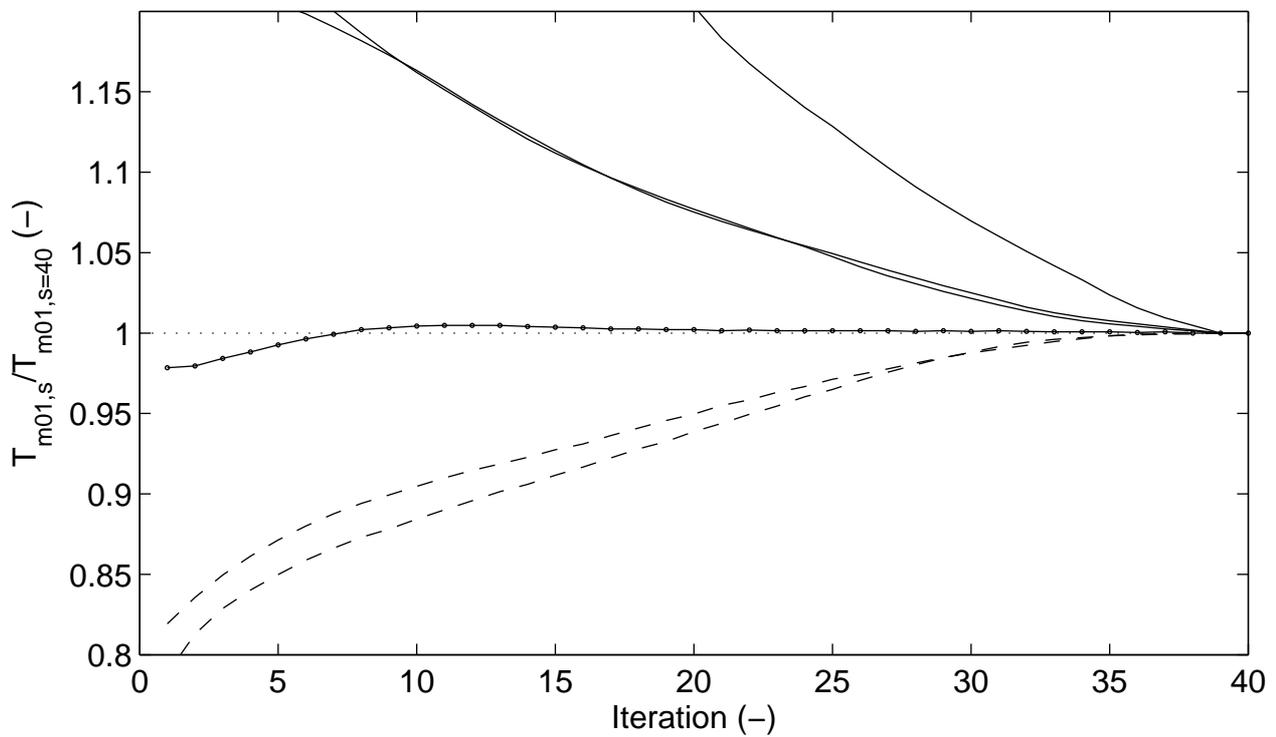
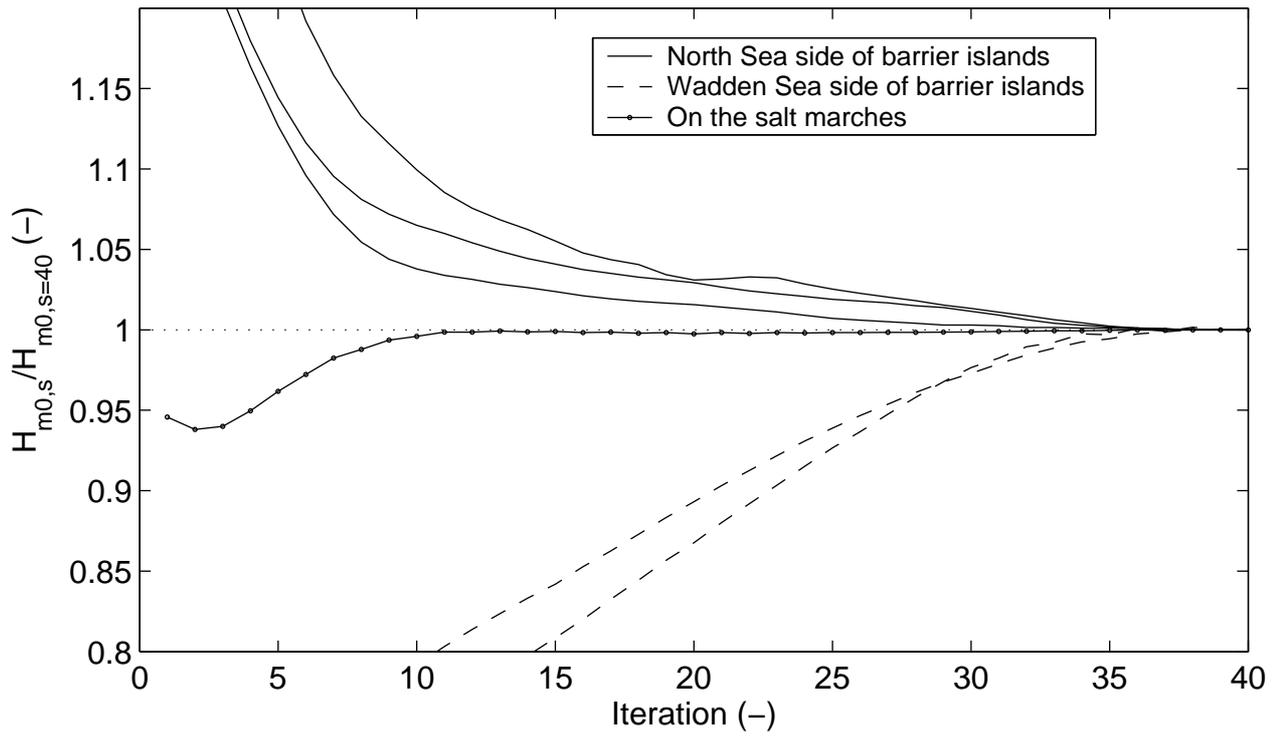
Amelander Zeegat storm of 16/12/2005 at 10:00
 Local convergence error in wave height: $|H_{m0,s} - H_{m0,s-1}|/H_{m0,s}$

Delft3D-WAVE



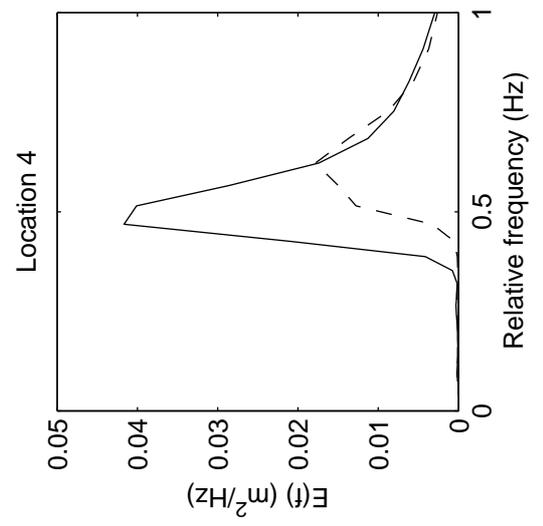
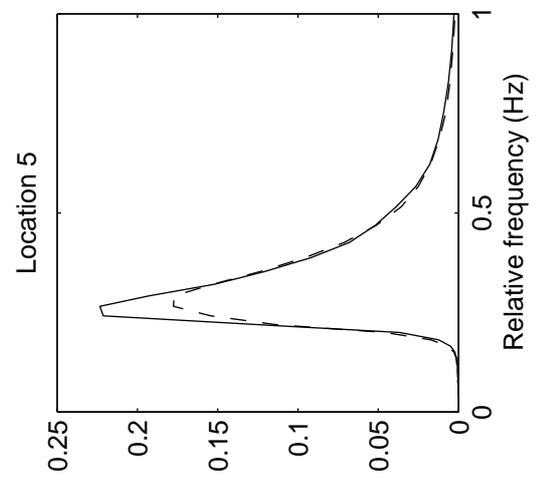
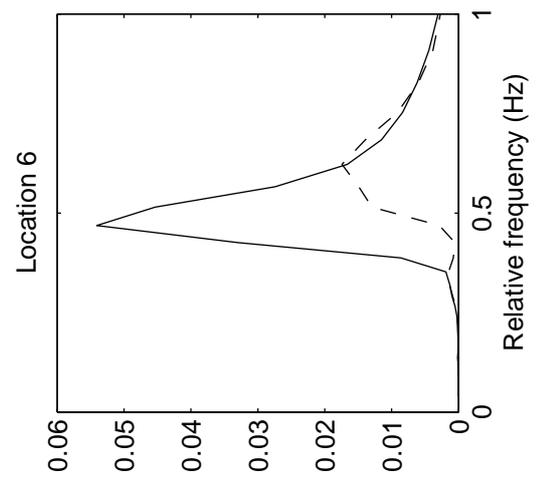
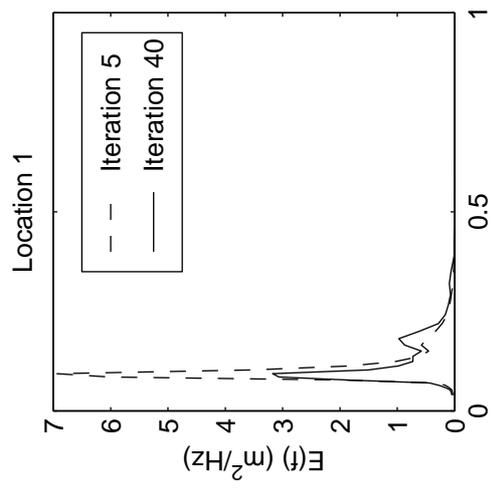
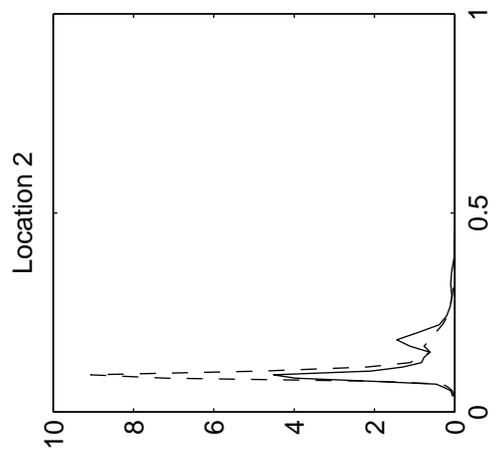
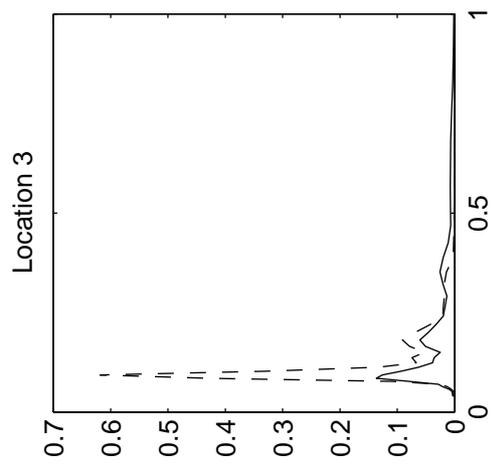
Amelander Zeegat storm of 16/12/2005 at 10:00
 Local convergence error in mean period: $|T_{m01,s} - T_{m01,s-1}| / T_{m01,s}$

Delft3D-WAVE



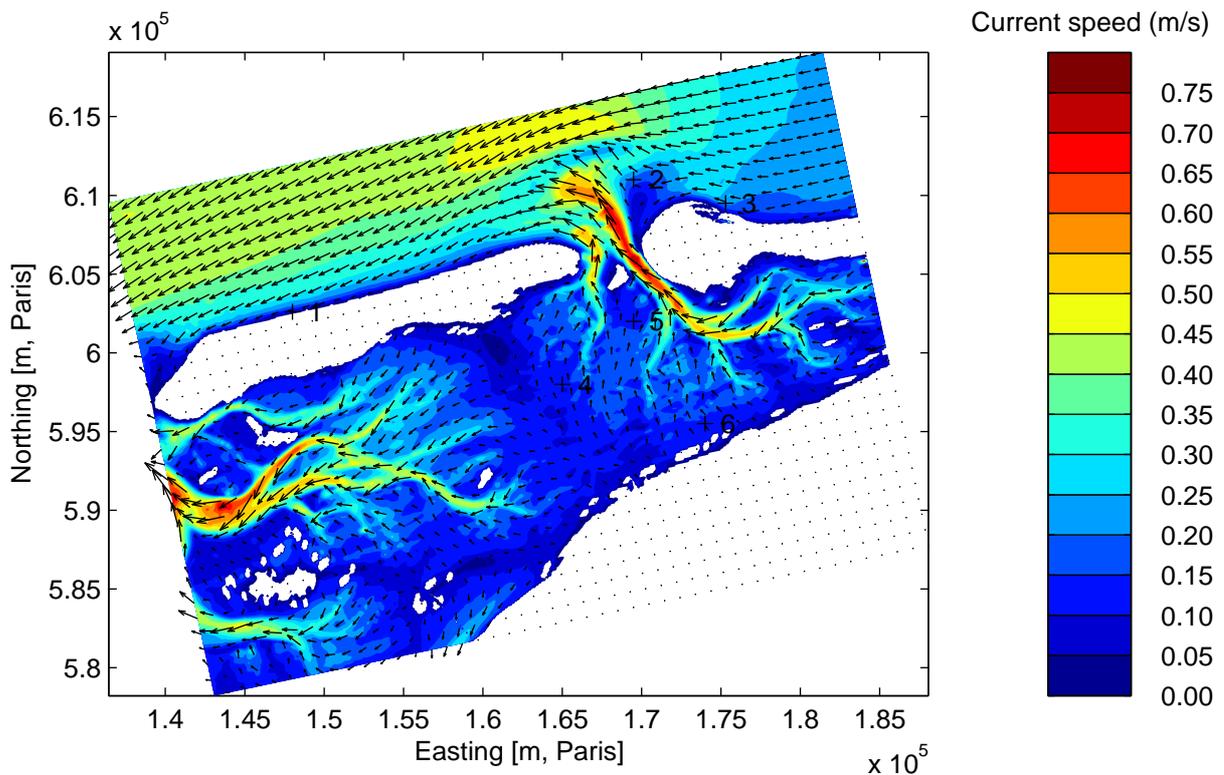
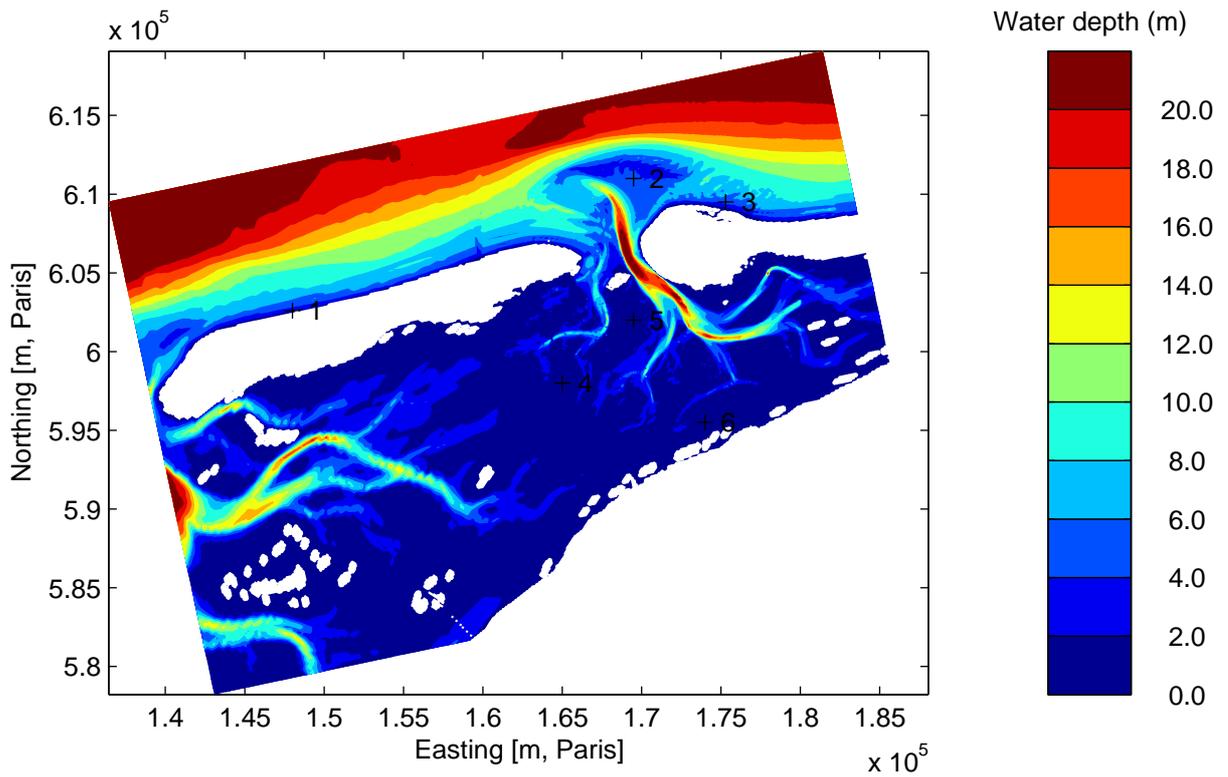
Amelander Zeegat storm of 16/12/2005 at 10:00
 Iteration curves of wave height and period, normalised with values at iteration 40

Delft3D-WAVE



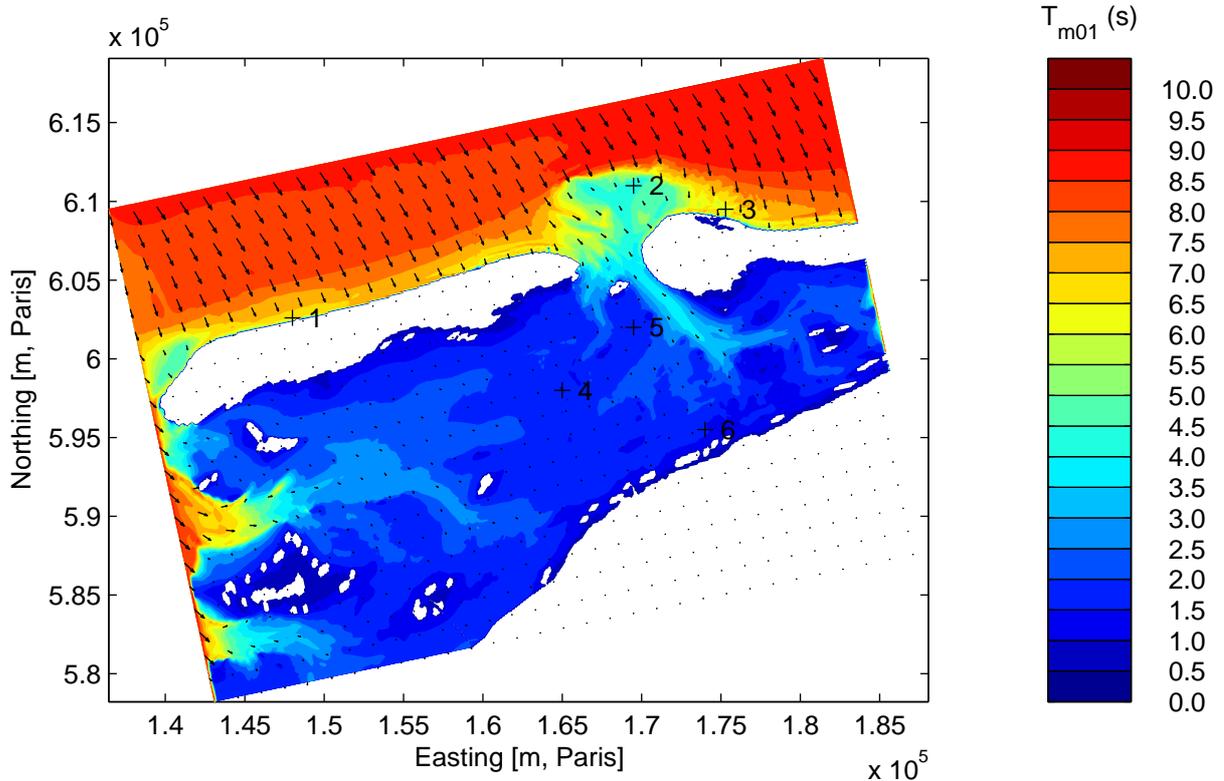
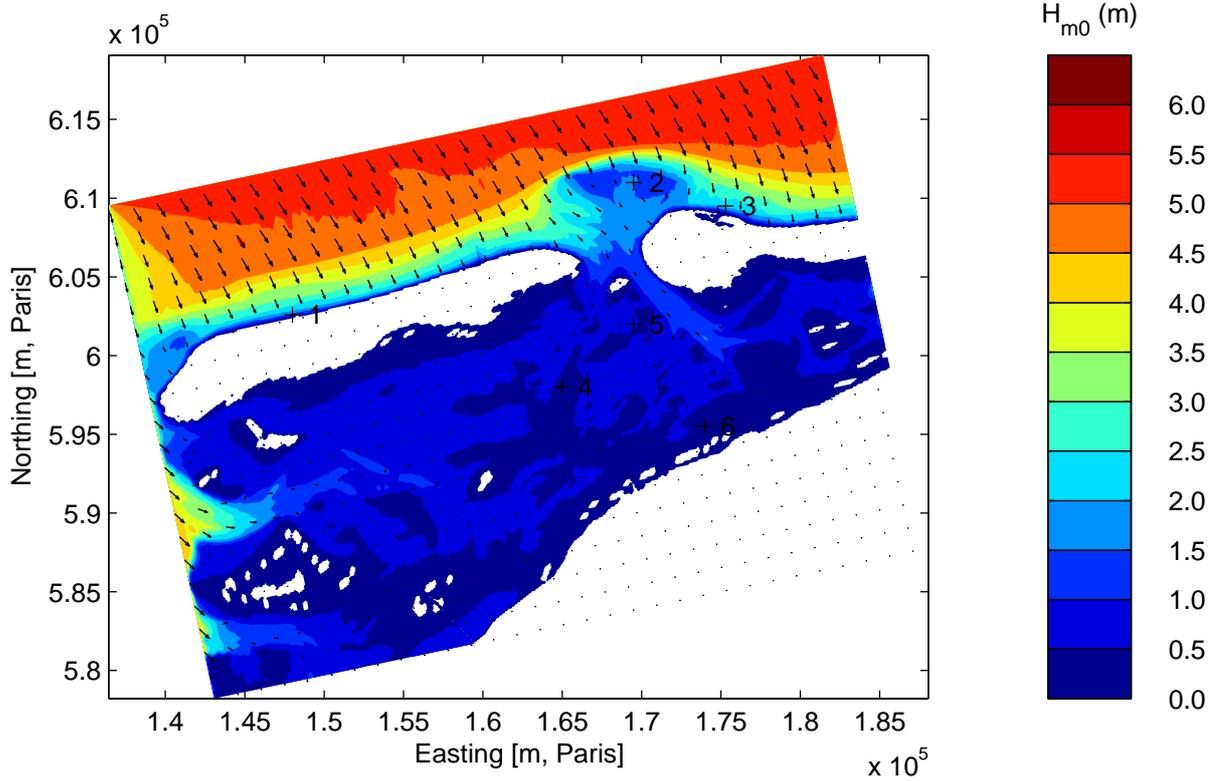
Amelander Zeegat storm of 16/12/2005 at 10:00
 Convergence behaviour of energy density spectra

Delft3D-WAVE



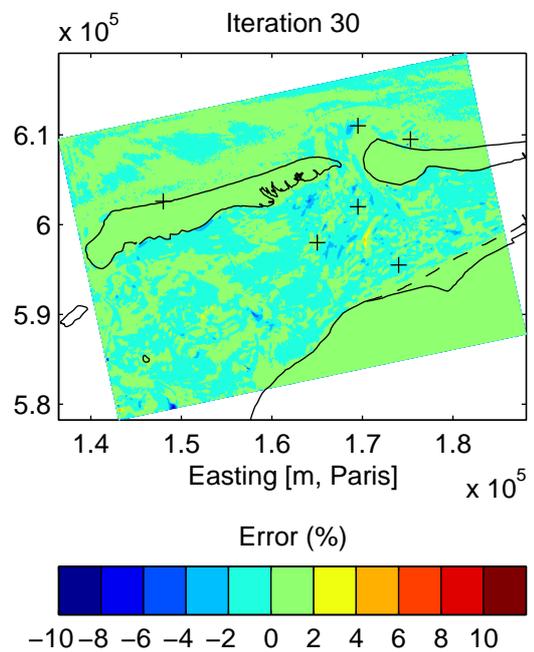
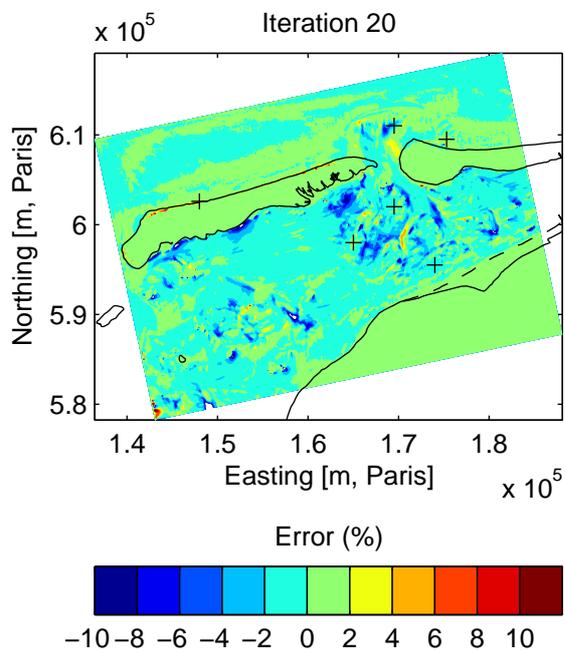
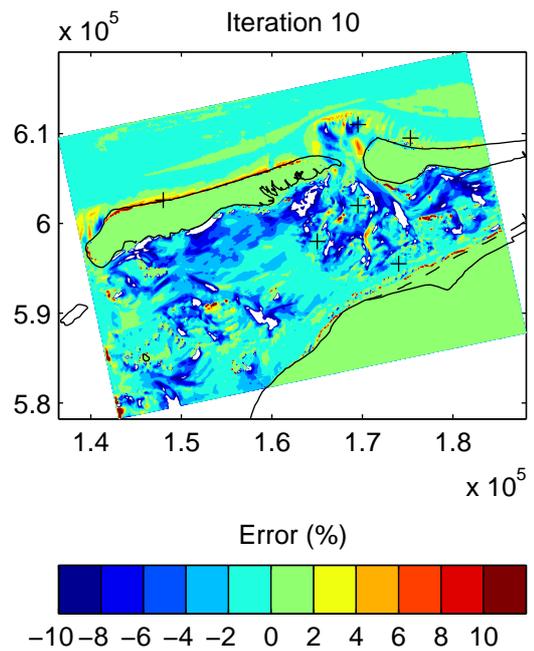
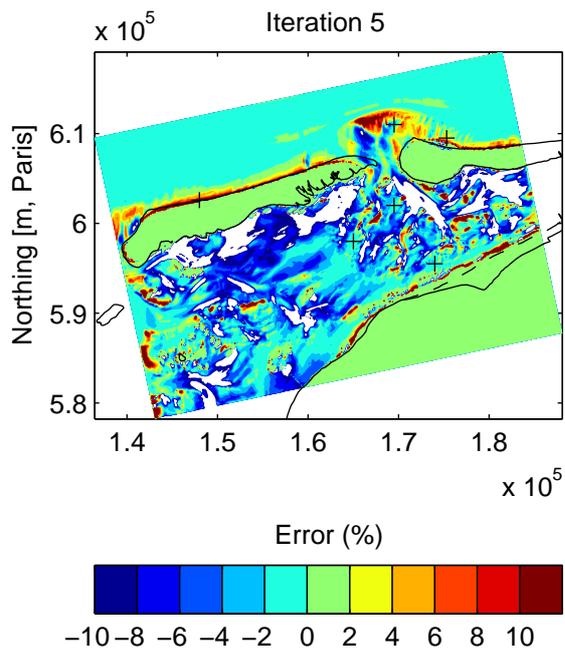
Amelander Zeegat storm of 09/02/2006 at 11:00
Water depth (above) and current field (below).

Delft3D-WAVE



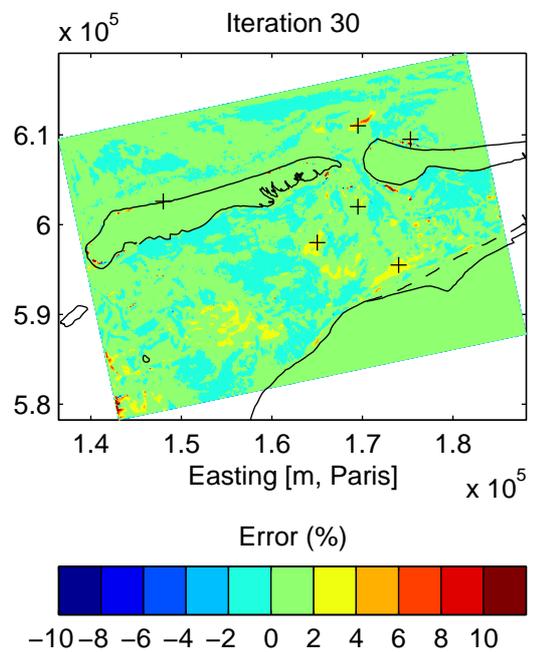
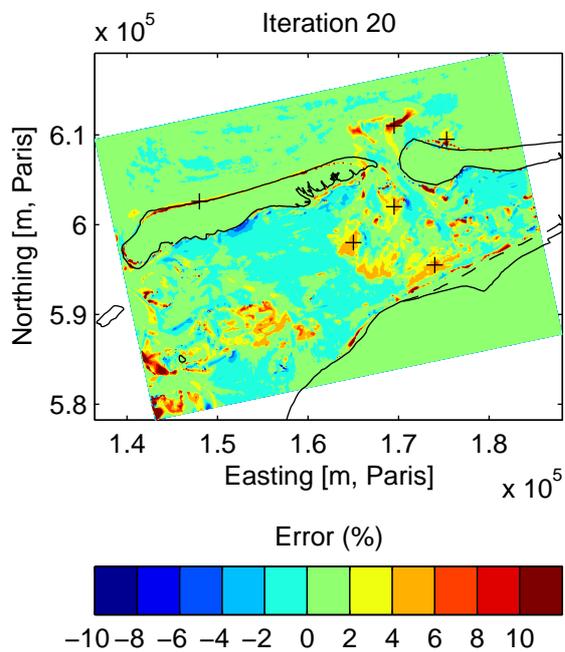
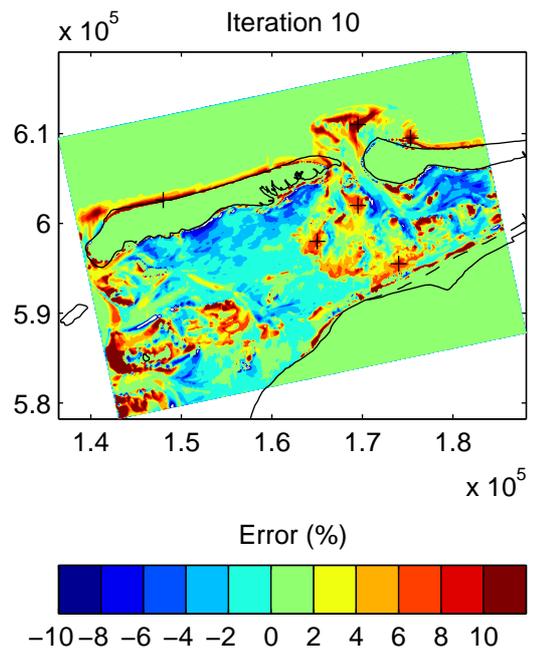
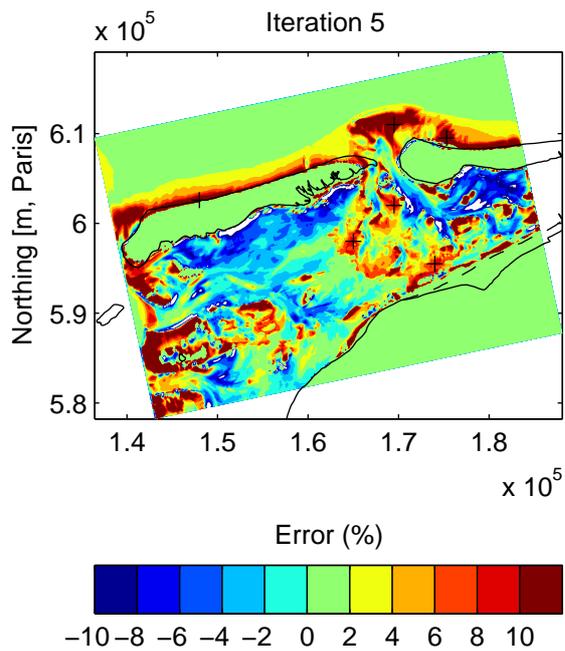
Amelander Zeegat storm of 09/02/2006 at 11:00
 Significant wave height (above) and mean period (below).

Delft3D-WAVE



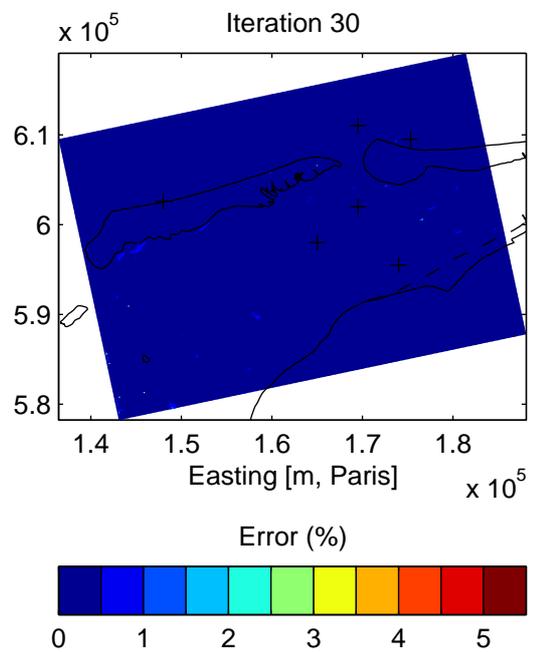
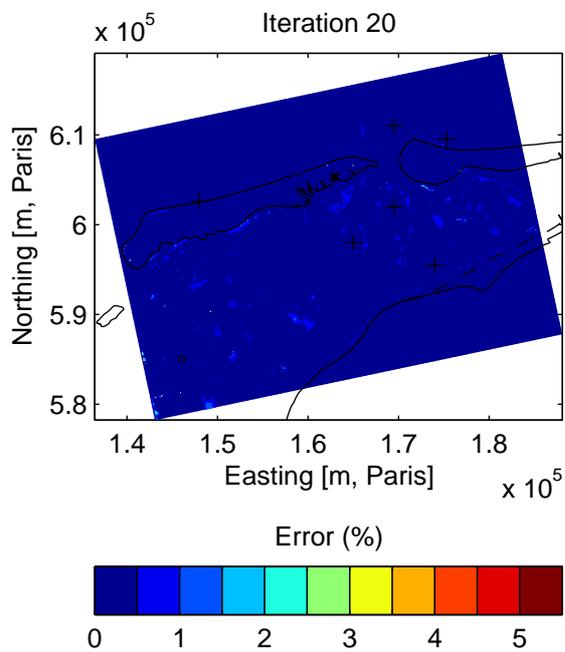
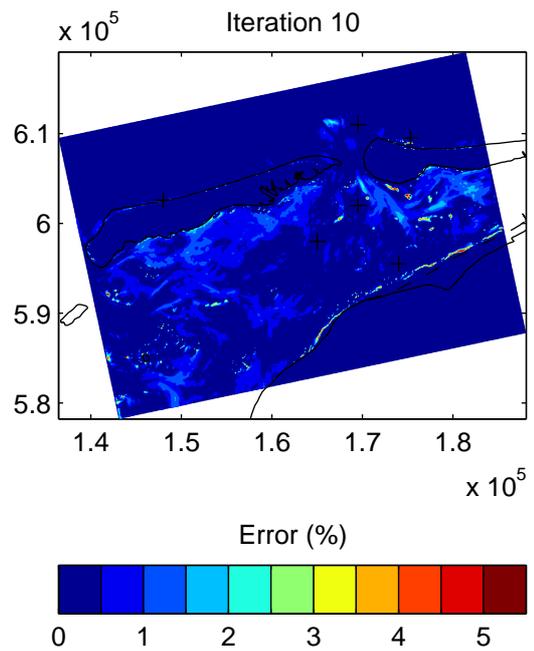
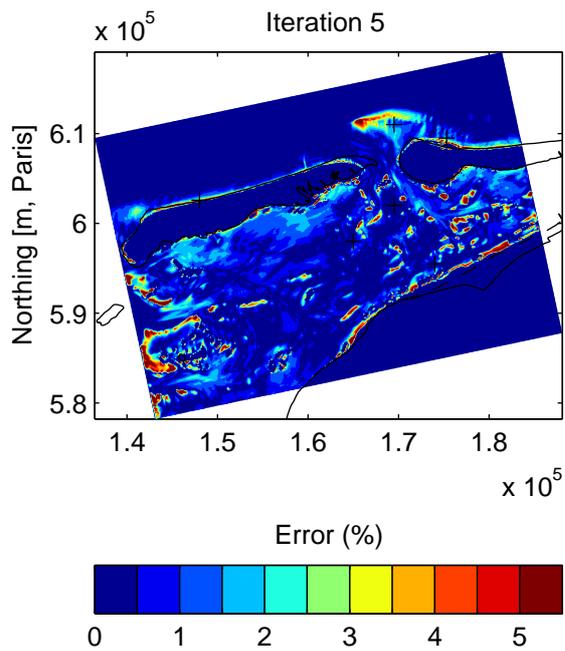
Amelander Zeegat storm of 09/02/2006 at 11:00
 Global convergence error in wave height: $(H_{m0,s} - H_{m0,s=40})/H_{m0,s=40}$

Delft3D-WAVE



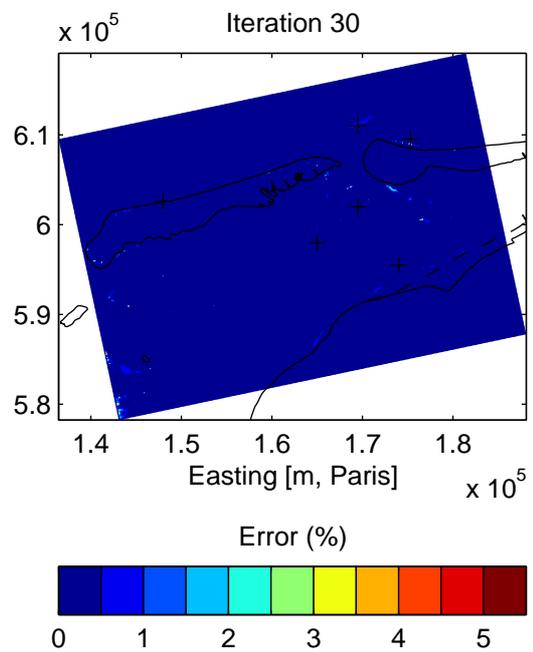
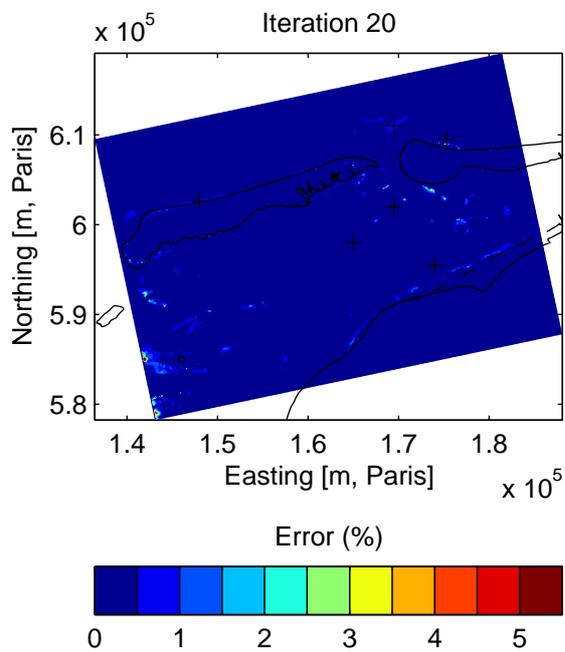
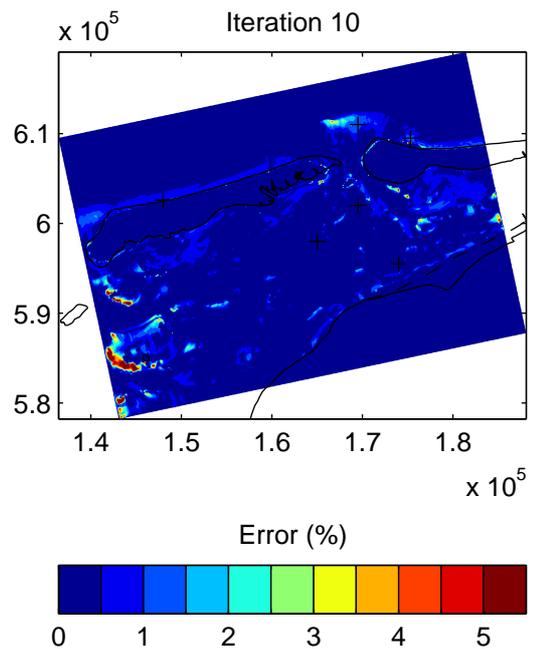
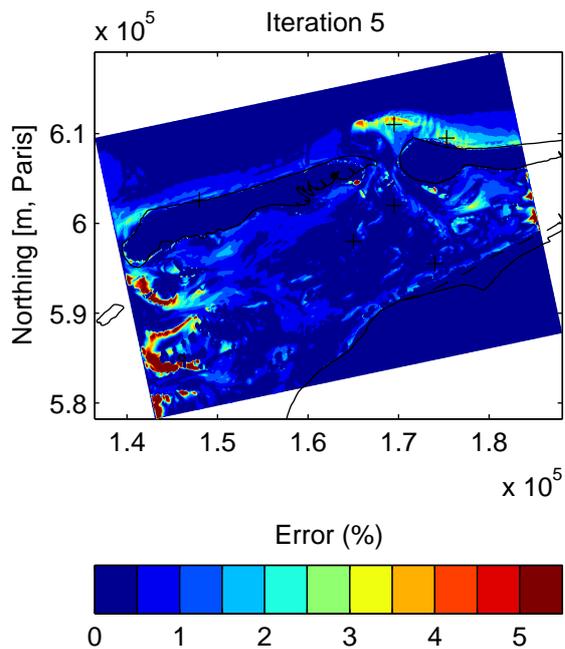
Amelander Zeegat storm of 09/02/2006 at 11:00
 Global convergence error in mean period: $(T_{m01,s} - T_{m01,s=40})/T_{m01,s=40}$

Delft3D-WAVE



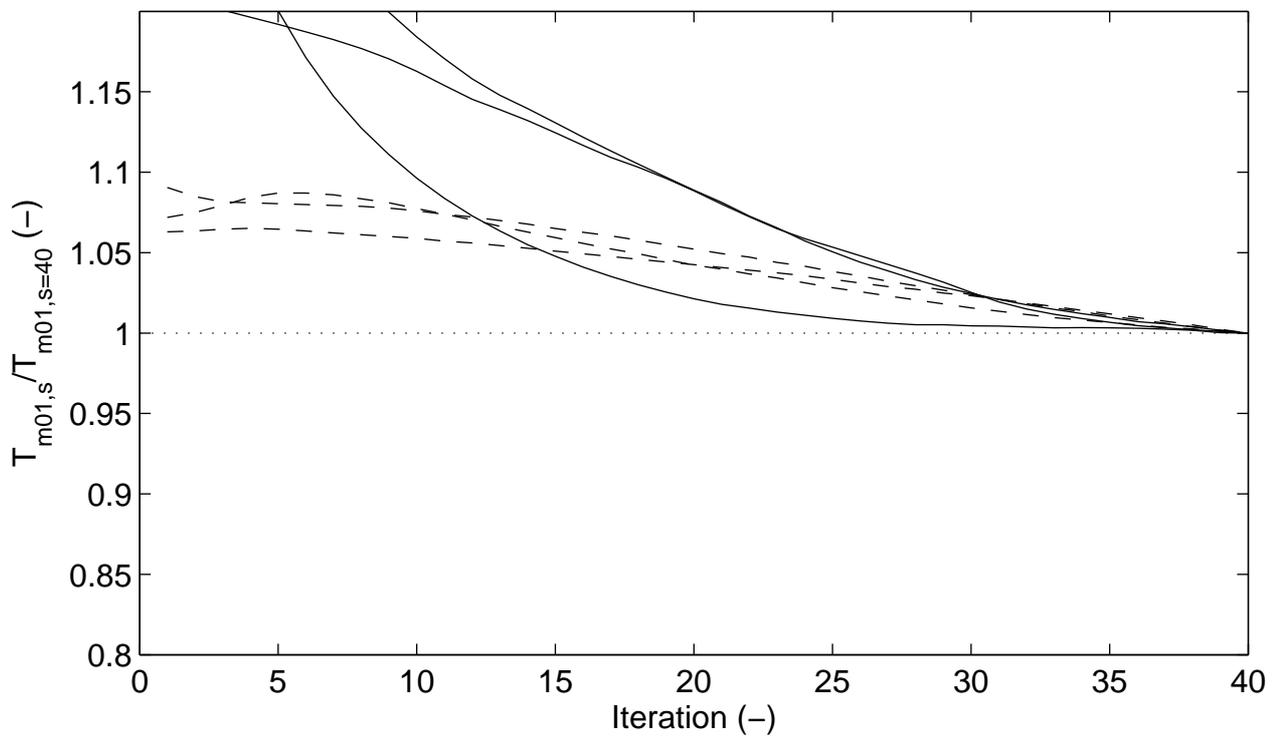
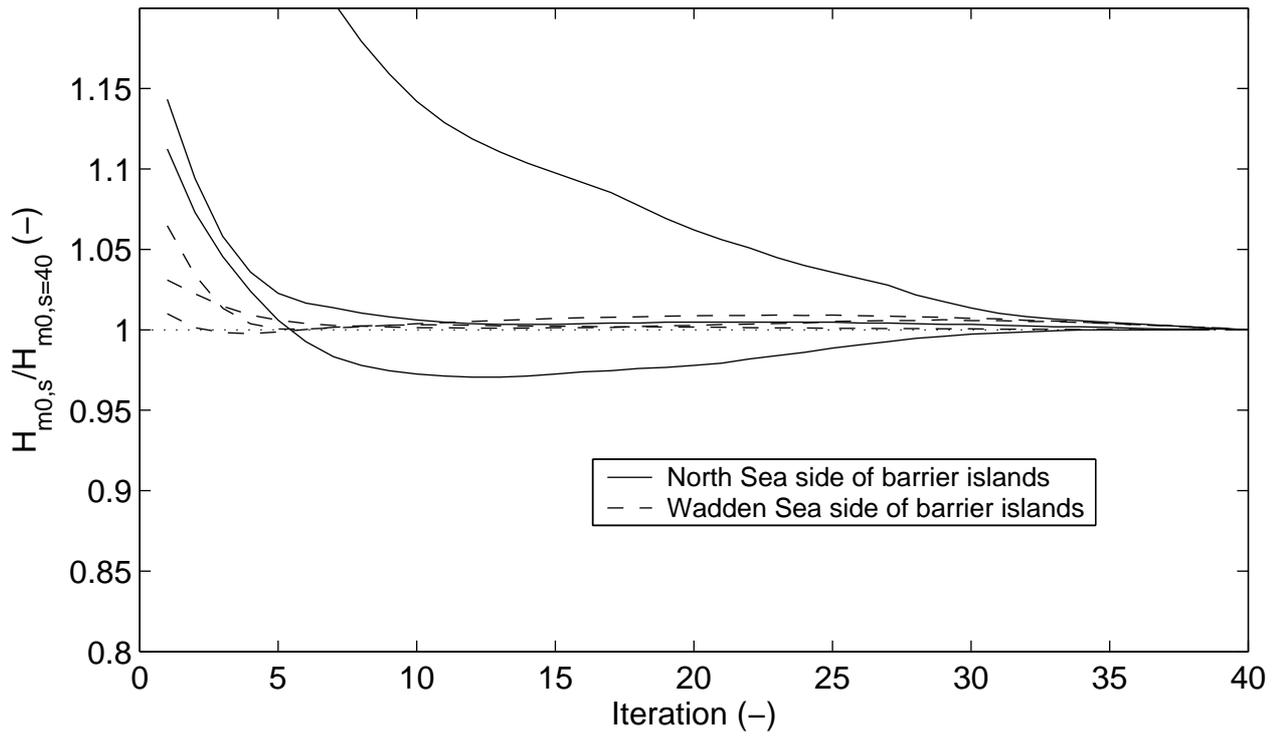
Amelander Zeegat storm of 09/02/2006 at 11:00
 Local convergence error in wave height: $|H_{m0,s} - H_{m0,s-1}|/H_{m0,s}$

Delft3D-WAVE



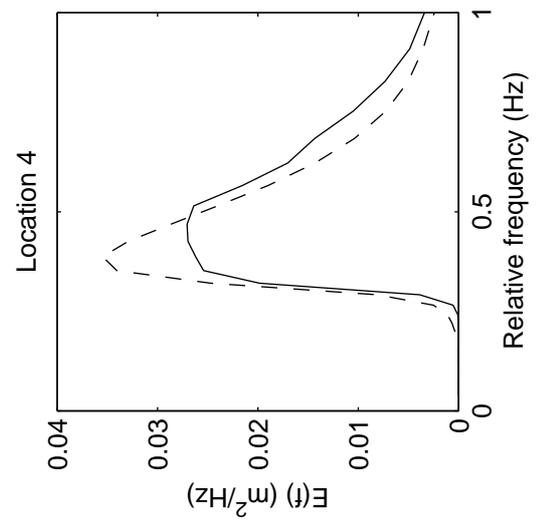
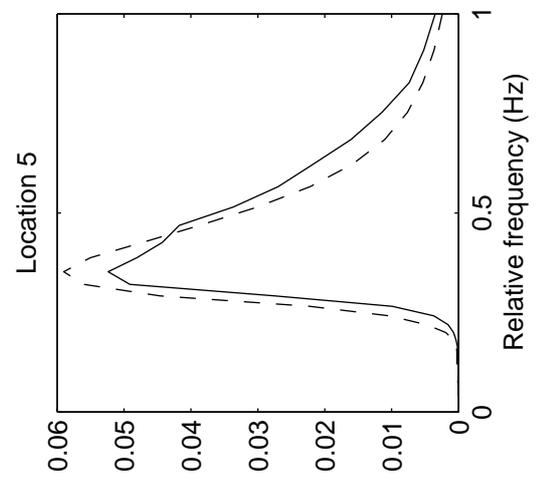
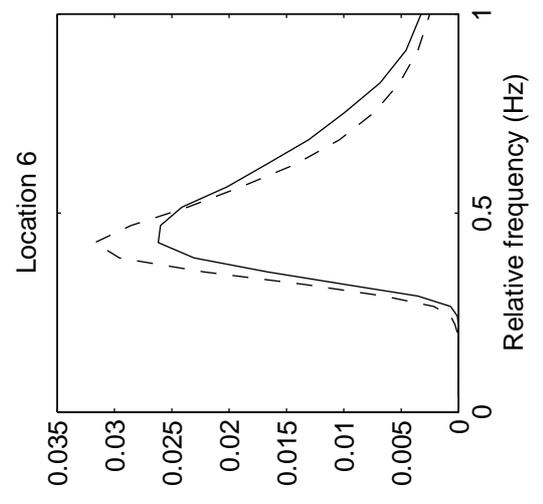
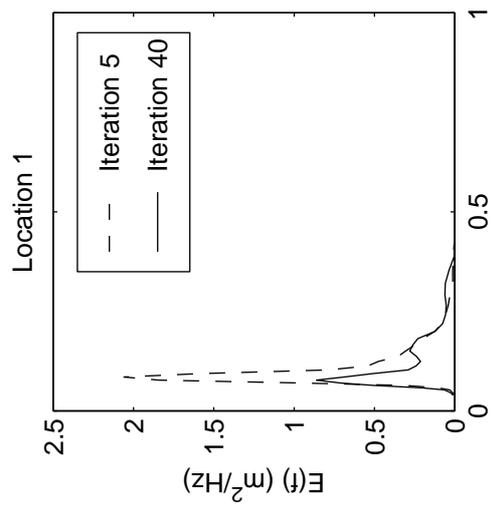
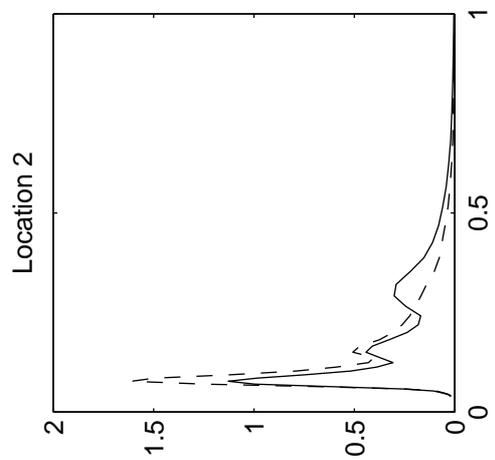
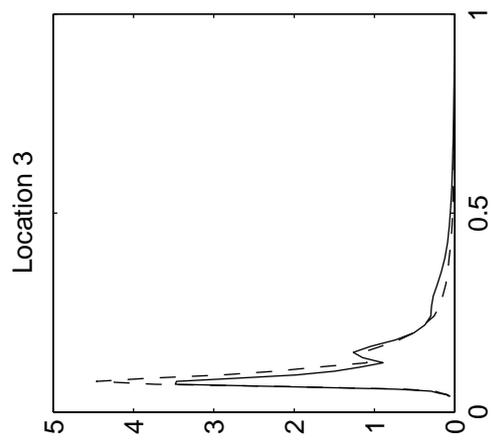
Amelanders Zeegat storm of 09/02/2006 at 11:00
 Local convergence error in mean period: $|T_{m01,s} - T_{m01,s-1}| / T_{m01,s}$

Delft3D-WAVE



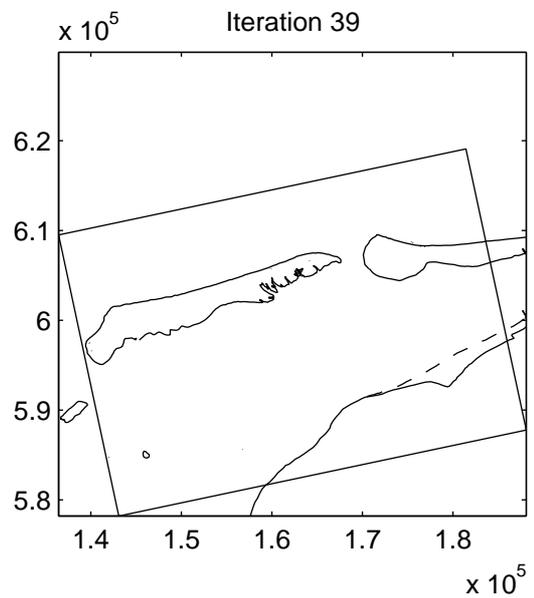
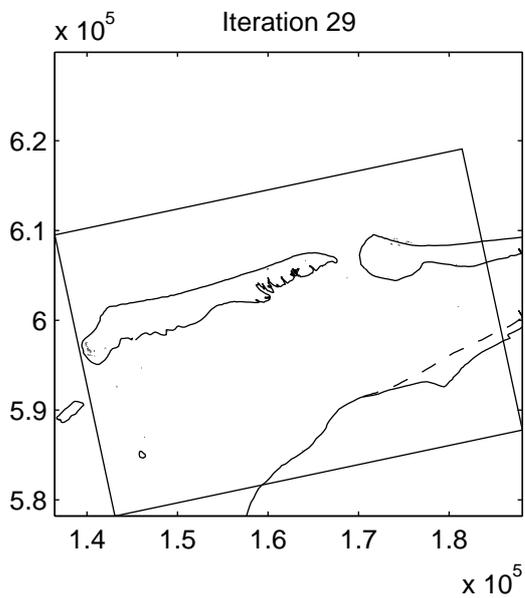
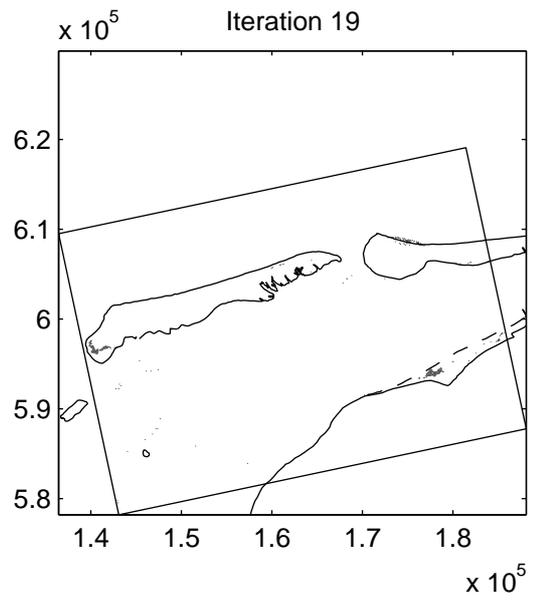
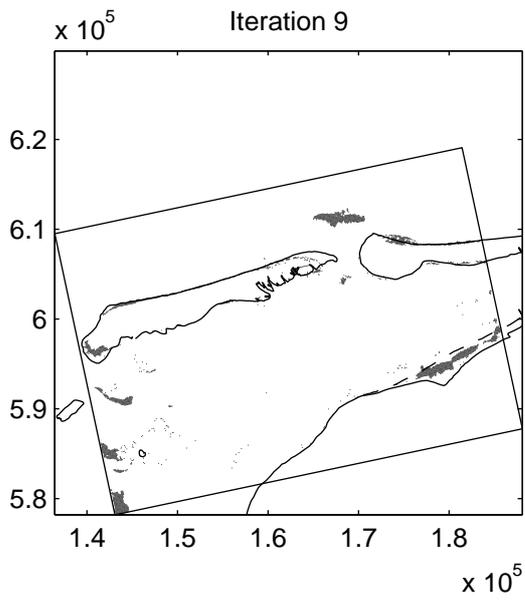
Amelander Zeegat storm of 09/02/2006 at 11:00
 Iteration curves of wave height and period, normalised with values at iteration 40

Delft3D-WAVE



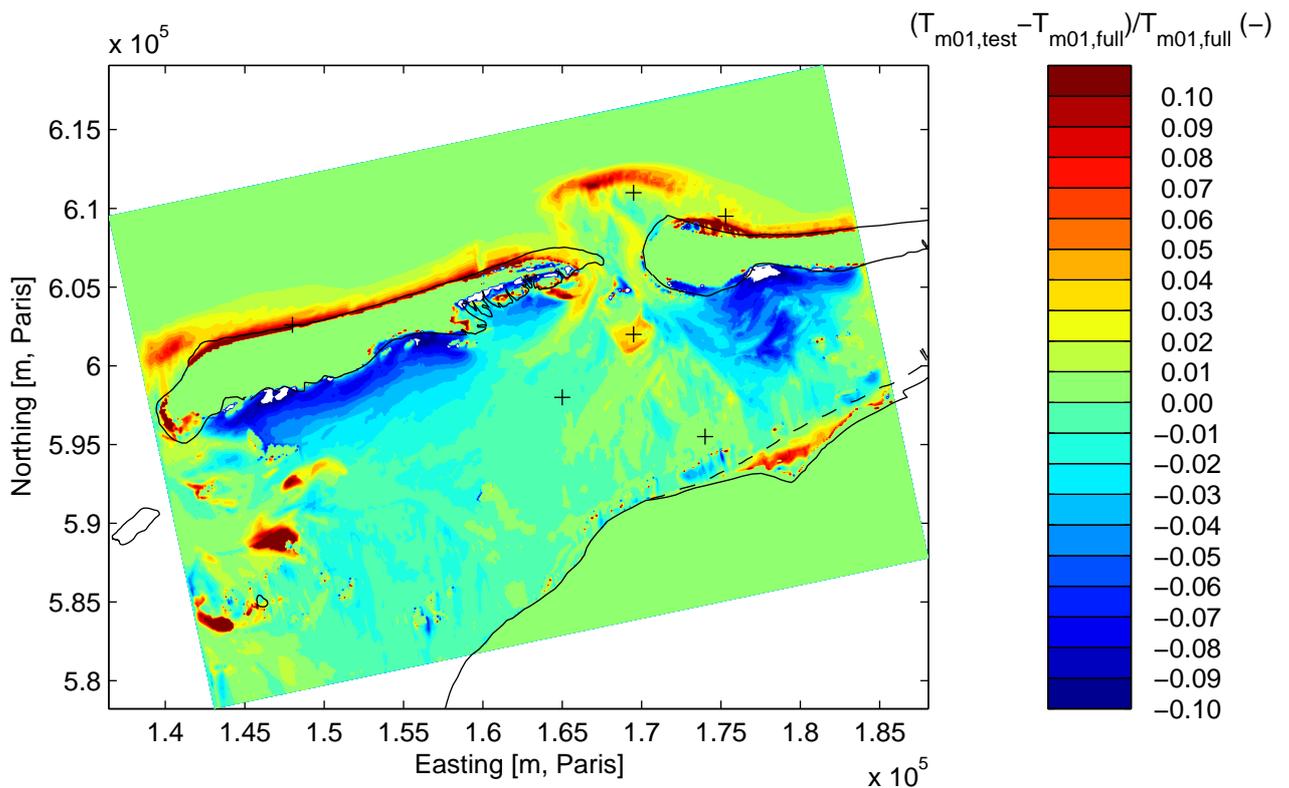
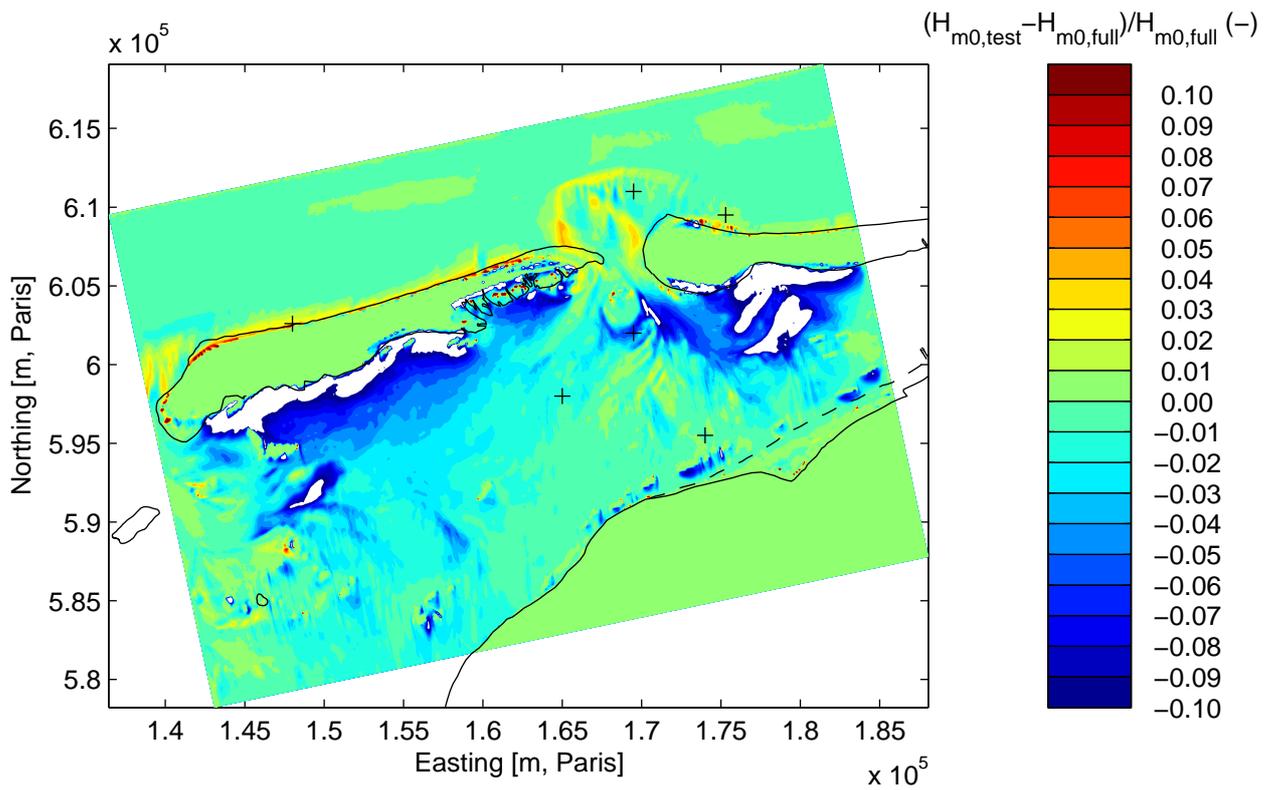
Amelander Zeegat storm of 09/02/2006 at 11:00
 Convergence behaviour of energy density spectra

Delft3D-WAVE



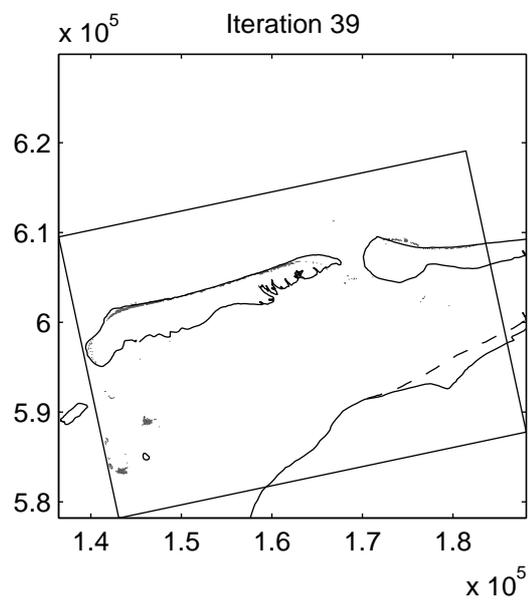
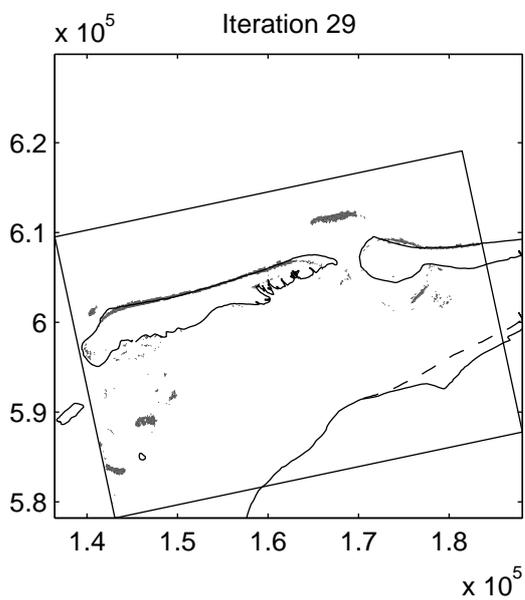
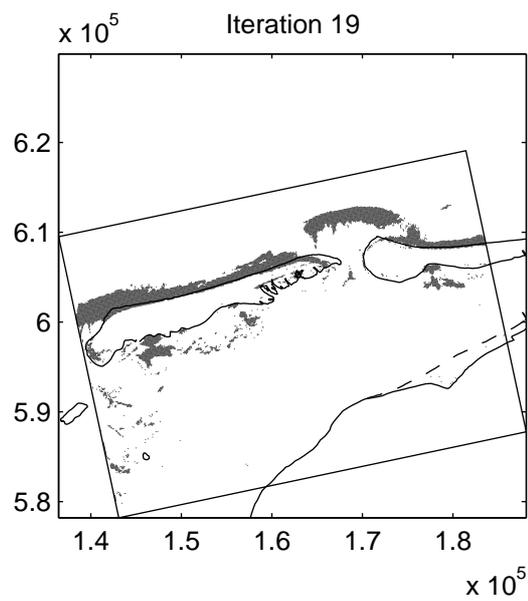
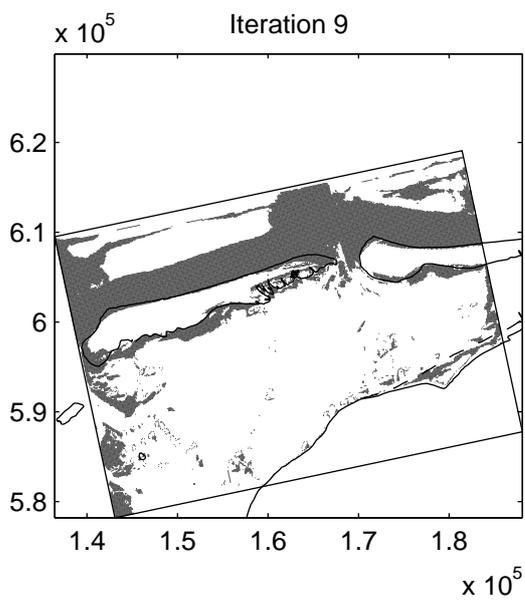
Amelanders Zeegat storm of 2005/12/16 at 10:00
 Calculation masks at various iteration levels, Run 1

Delft3D-WAVE



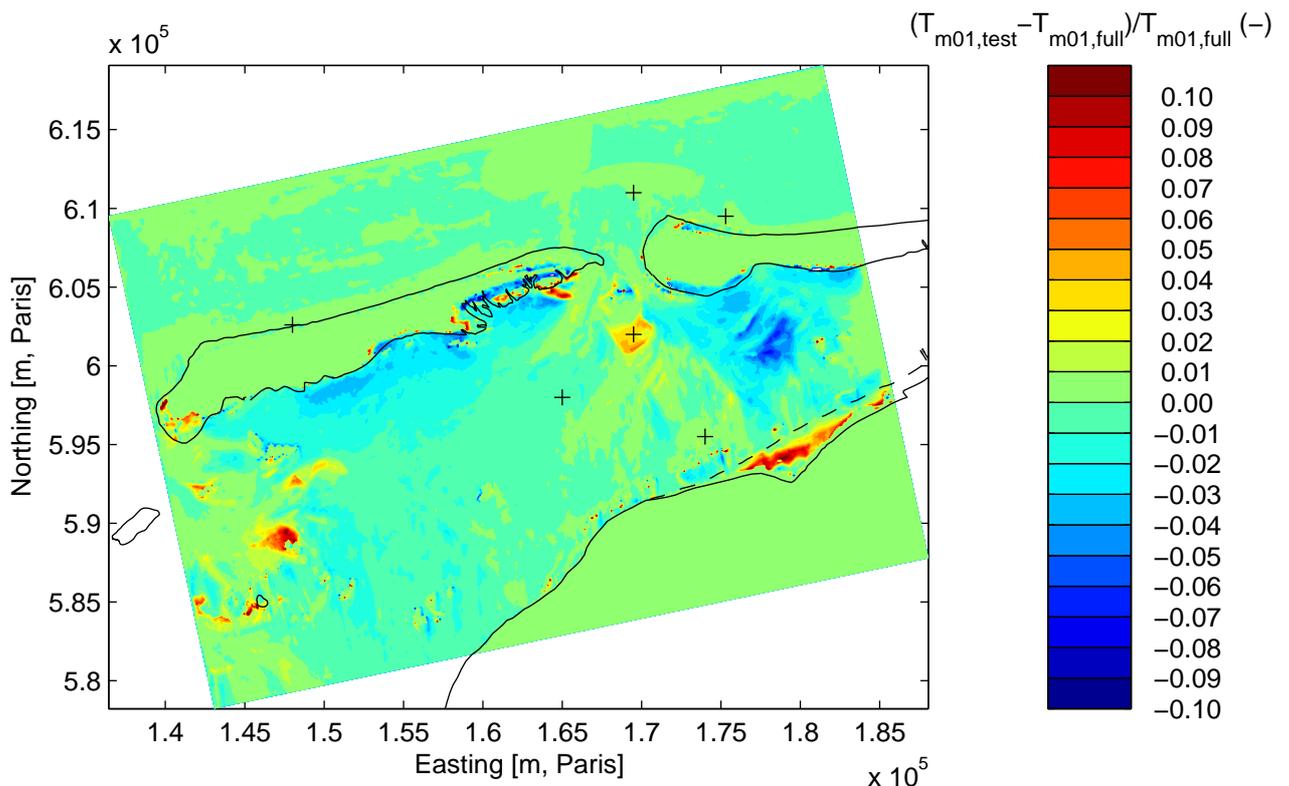
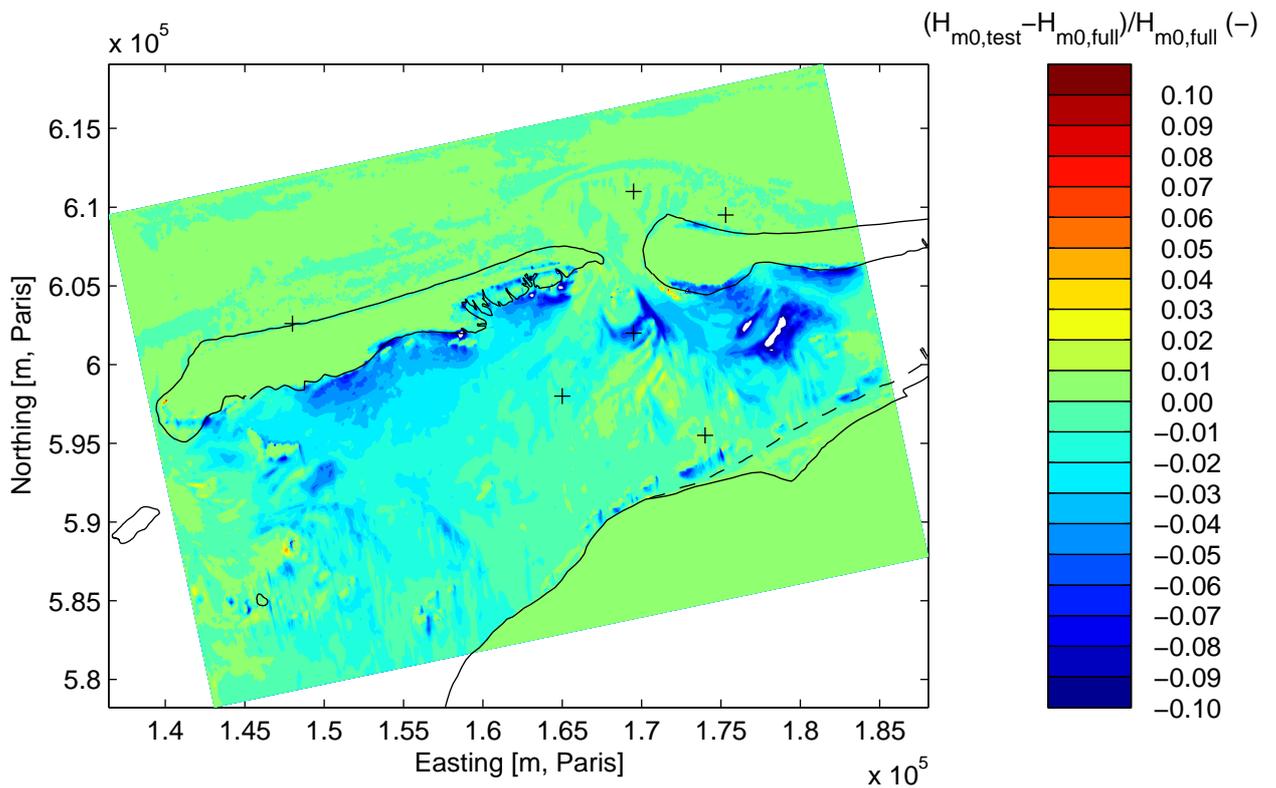
Amelanders Zeegat storm of 16/12/2005 at 10:00
 Error relative to full computation after 40 iterations, Run 1

Delft3D-WAVE



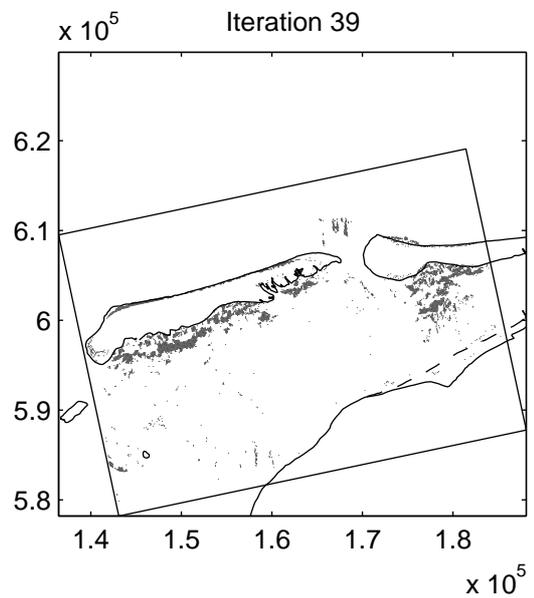
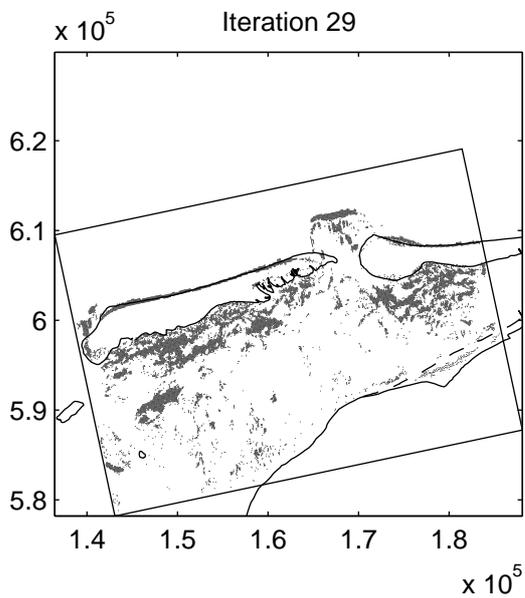
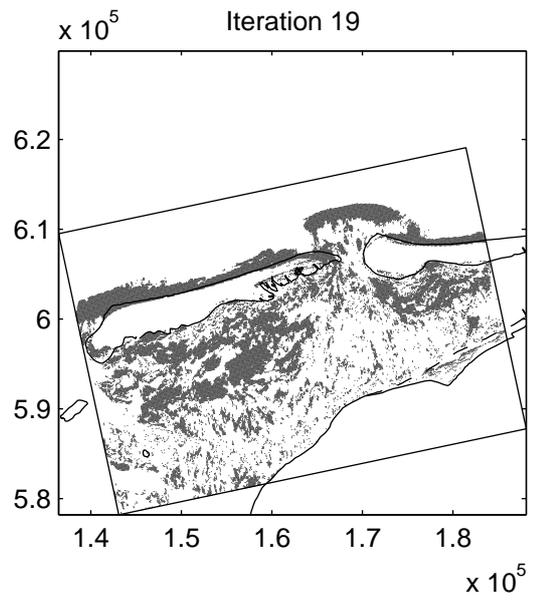
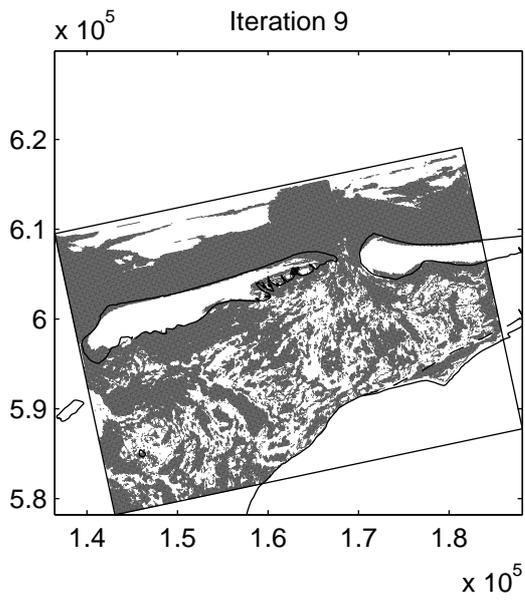
Amelanders Zeegat storm of 2005/12/16 at 10:00
 Calculation masks at various iteration levels, Run 2

Delft3D-WAVE



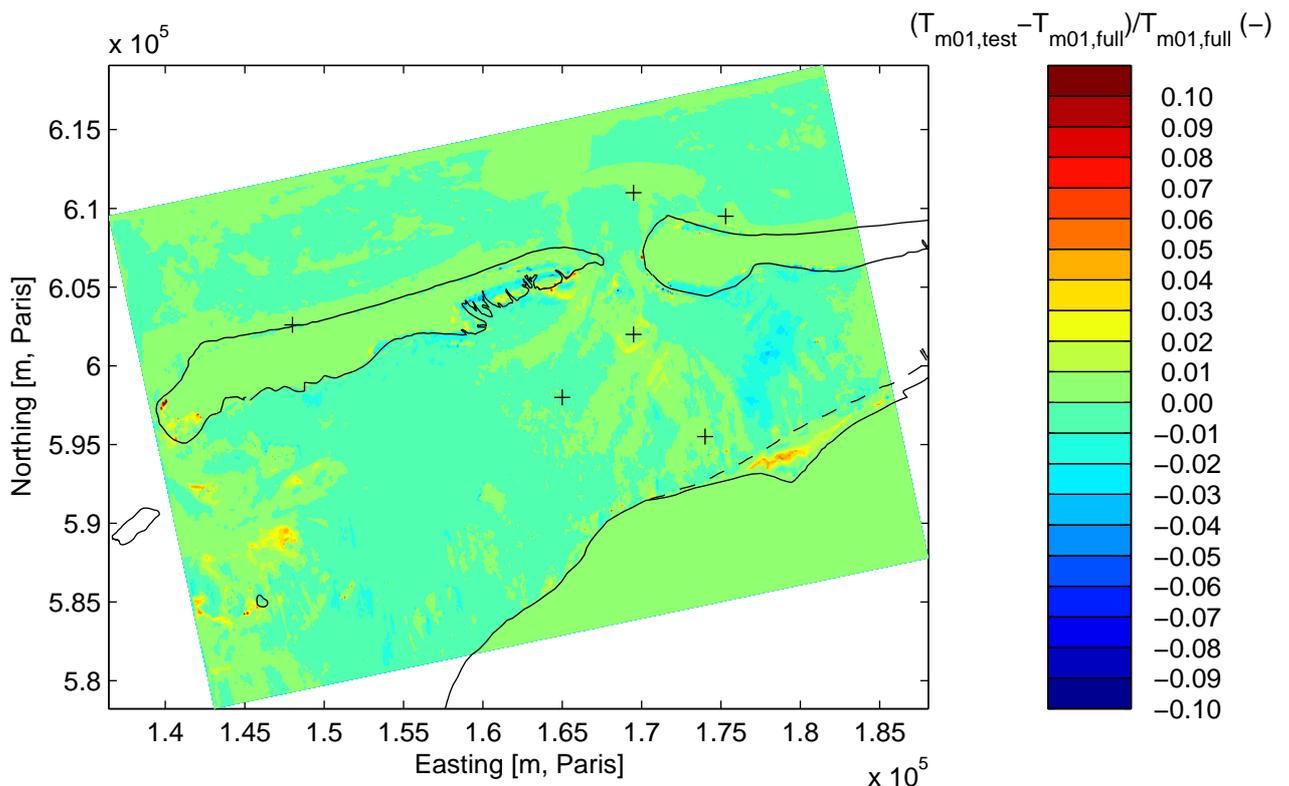
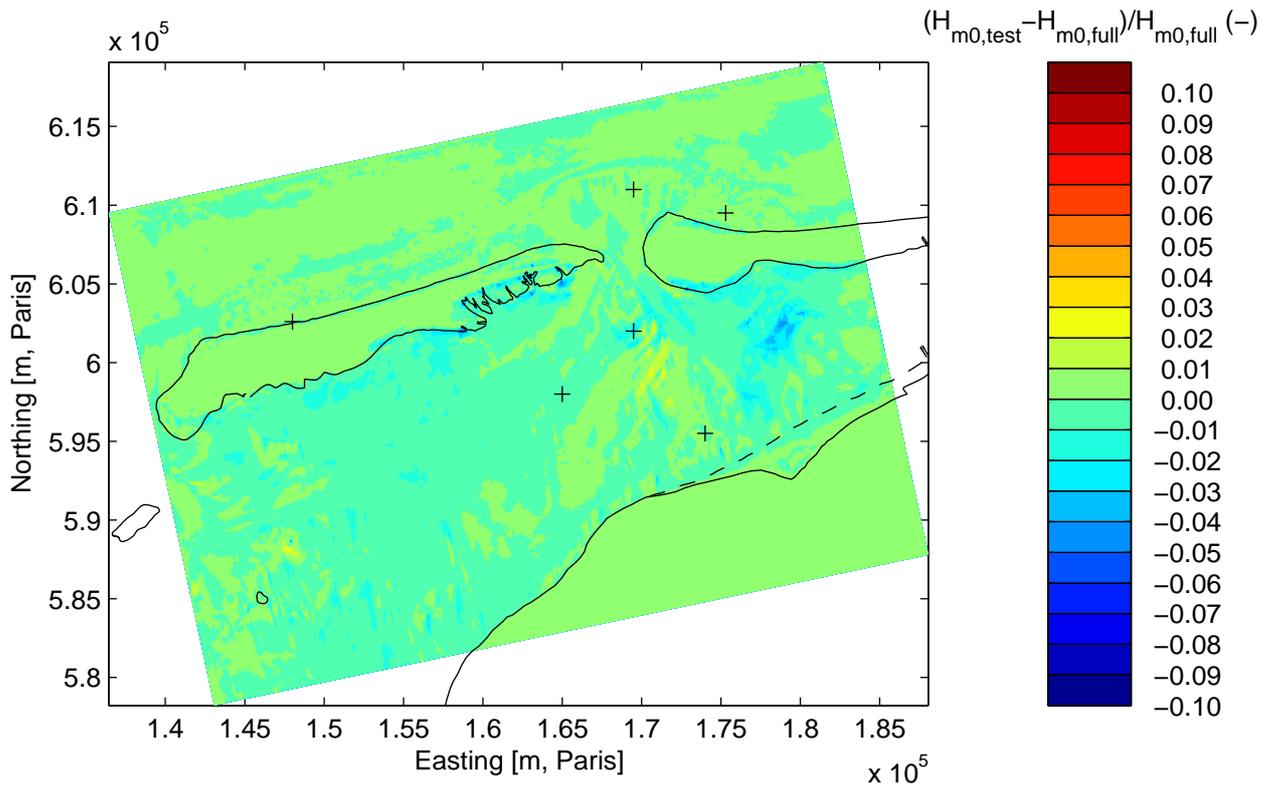
Ameland Zeegat storm of 16/12/2005 at 10:00
 Error relative to full computation after 40 iterations, Run 2

Delft3D-WAVE



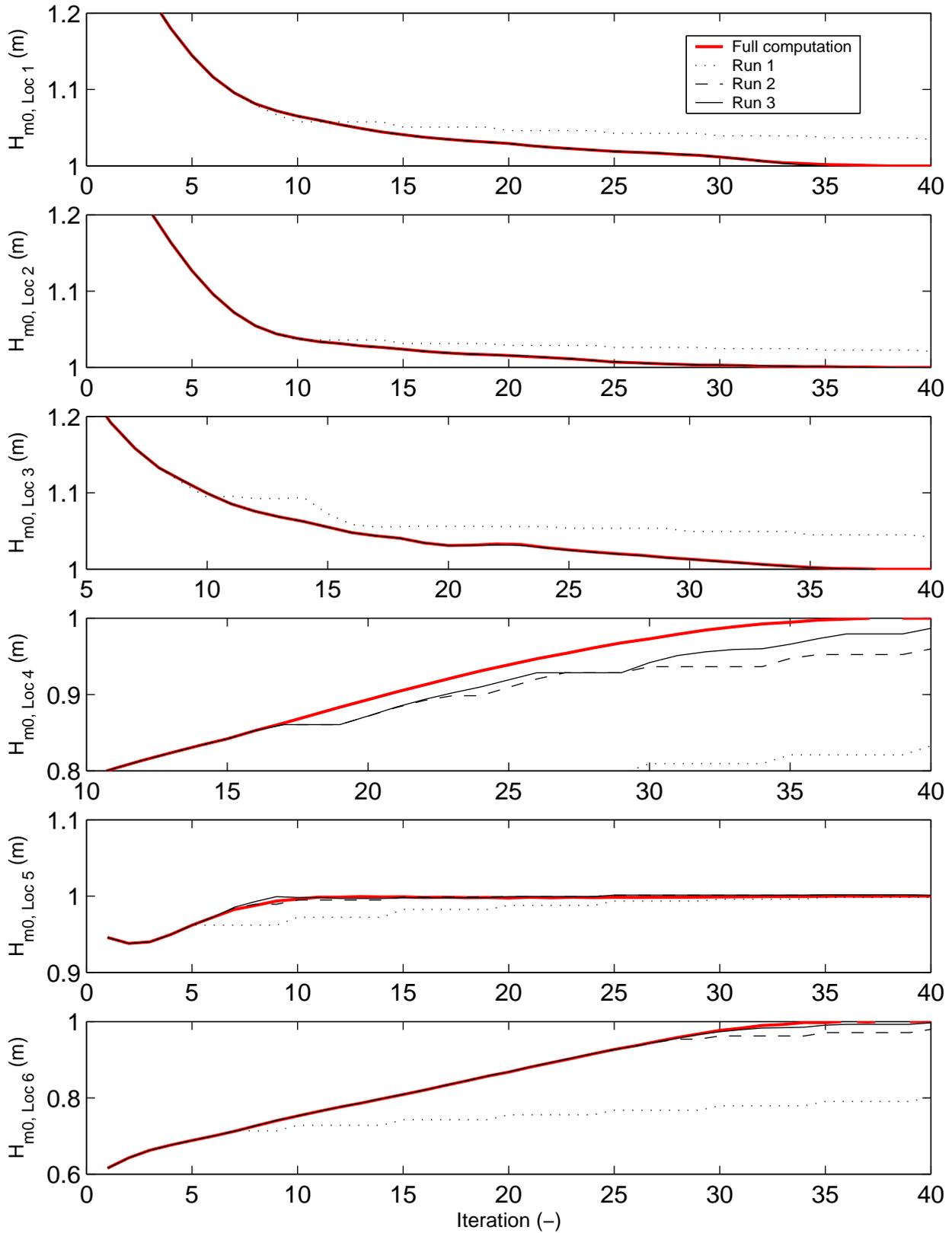
Amelanders Zeegat storm of 2005/12/16 at 10:00
 Calculation masks at various iteration levels, Run 3

Delft3D-WAVE



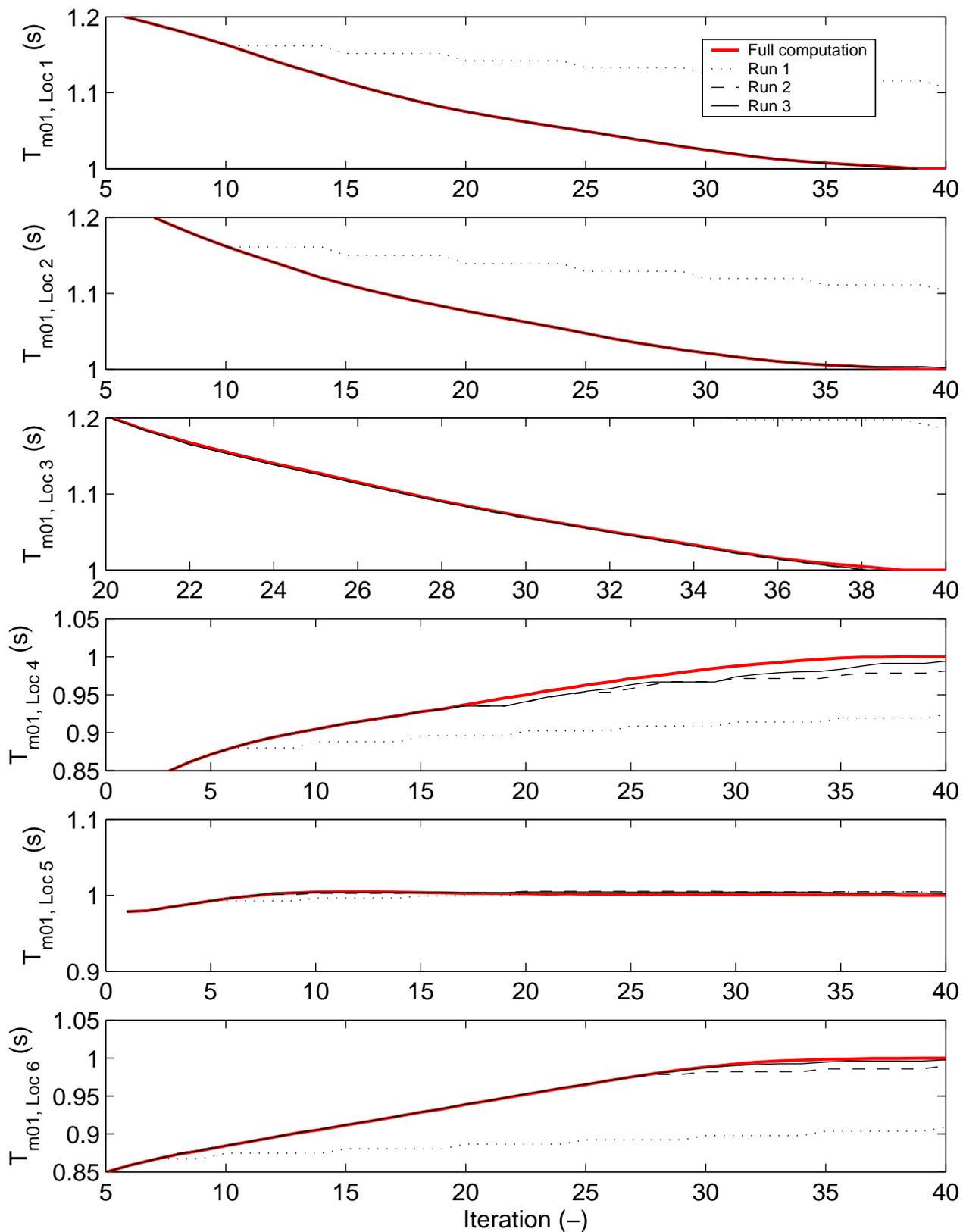
Ameland Zeegat storm of 16/12/2005 at 10:00
 Error relative to full computation after 40 iterations, Run 3

Delft3D-WAVE



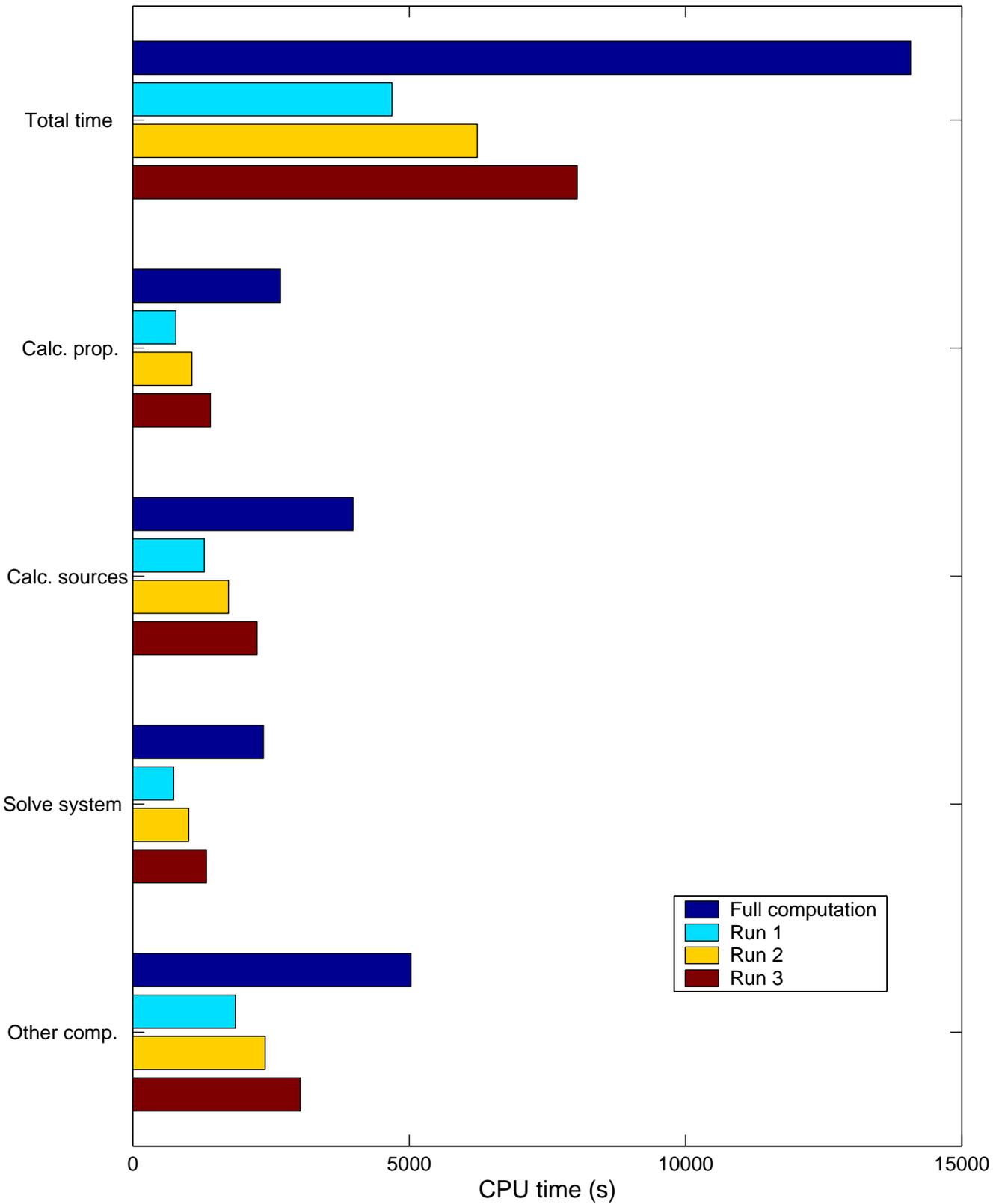
Amelander Zeegat storm of 16/12/2005 at 10:00
 Iteration curves of H_{m0} for Run 1 to 3, normalised with values at iteration 40

Delft3D-WAVE



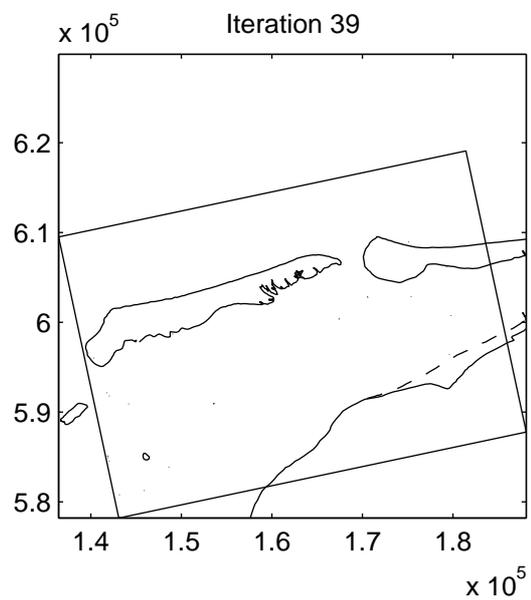
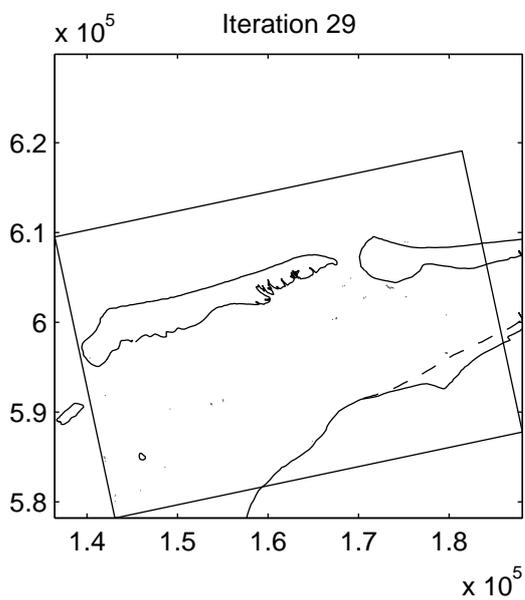
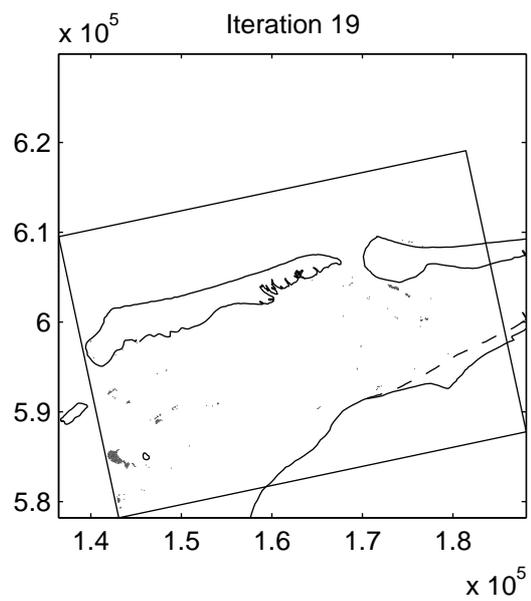
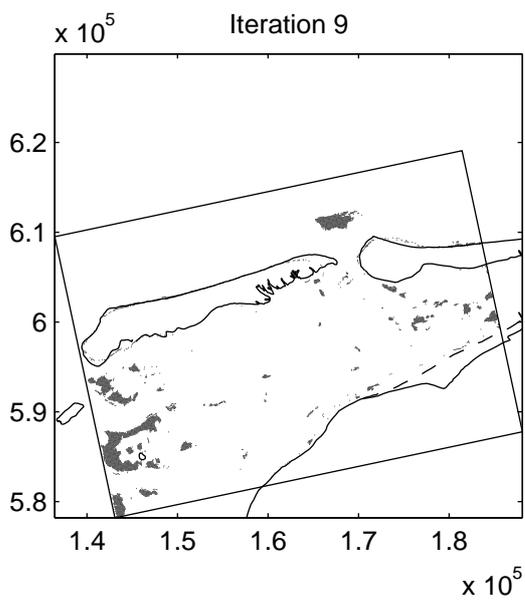
Amelander Zeegat storm of 16/12/2005 at 10:00
 Iteration curves of T_{m01} for Run 1 to 3, normalised with values at iteration 40

Delft3D-WAVE



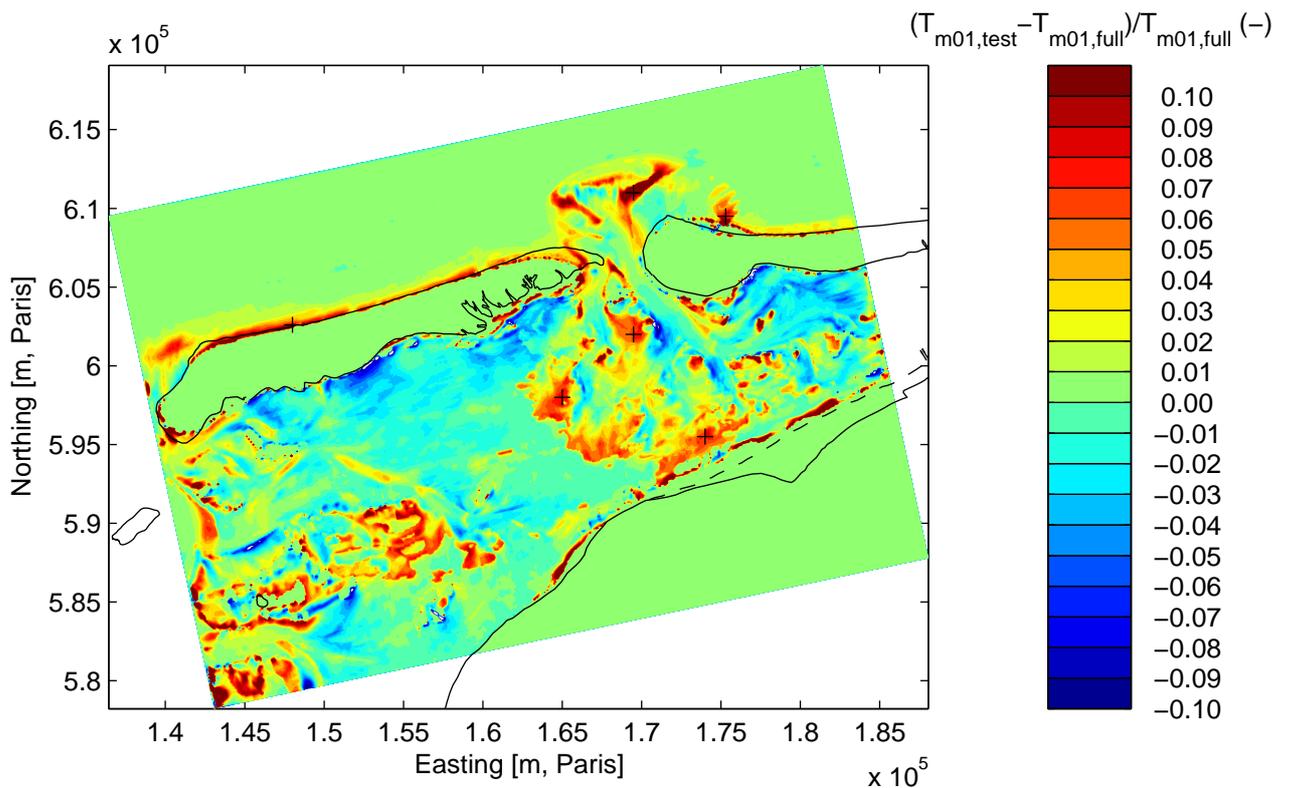
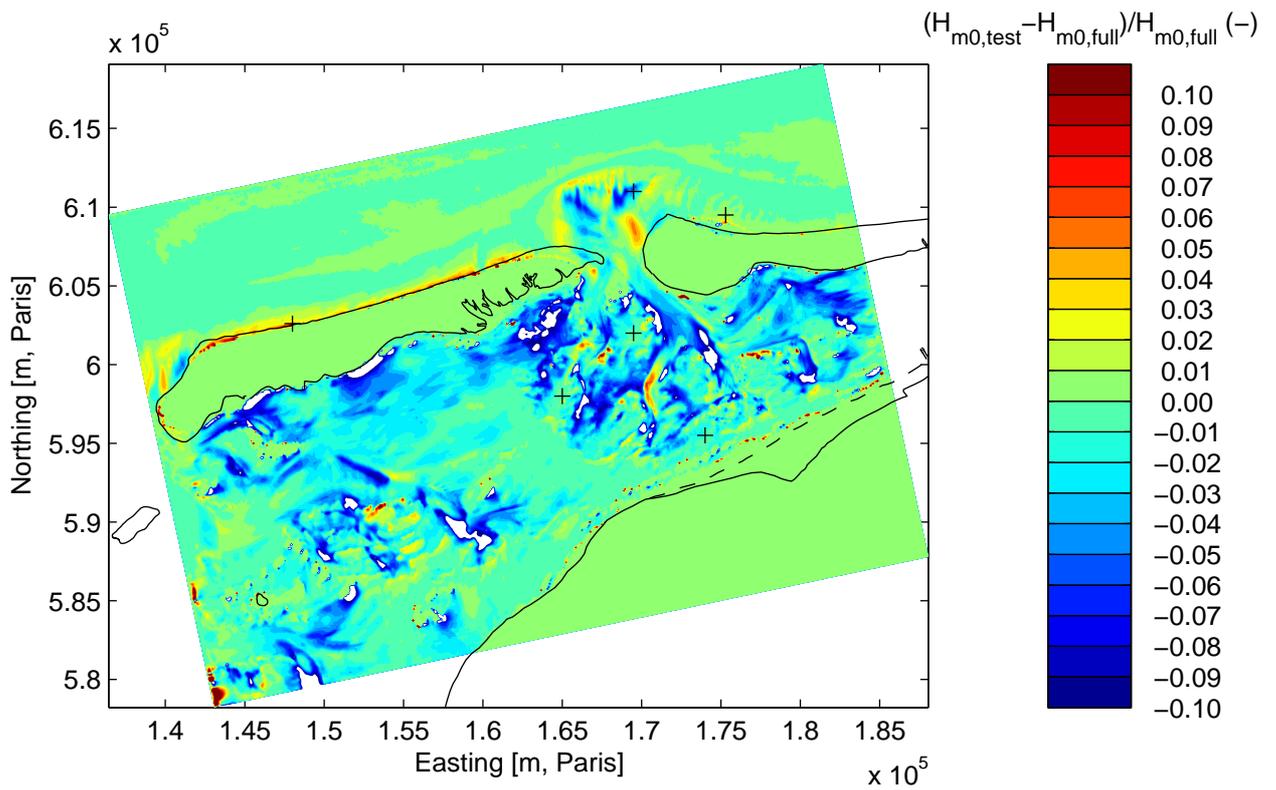
Amelander Zeegat storm of 16/12/2005 at 10:00
 Comparison of computational time

Delft3D-WAVE



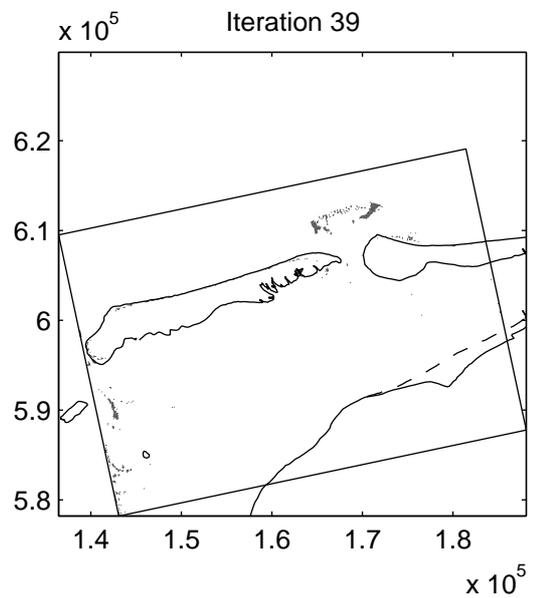
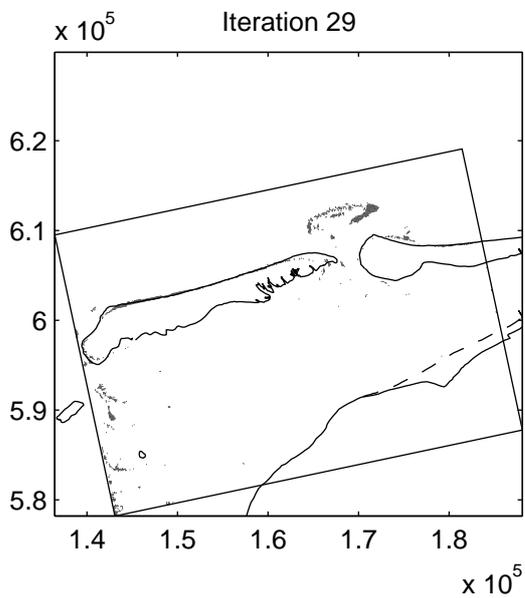
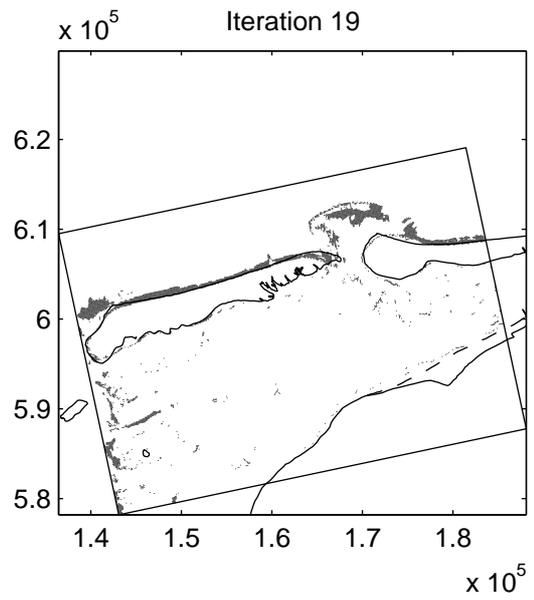
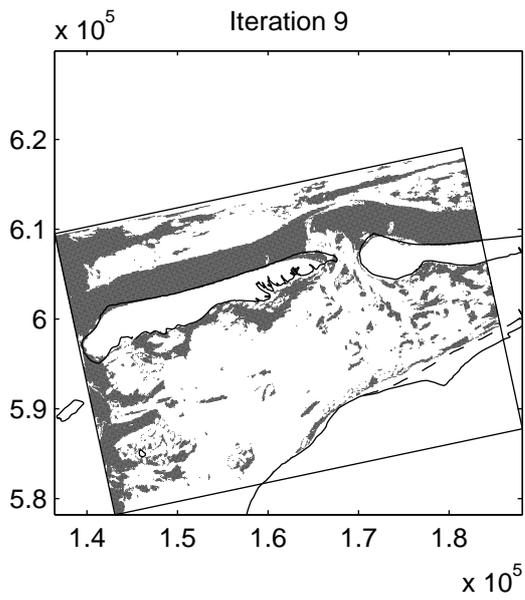
Amelanders Zeegat storm of 2006/02/09 at 11:00
 Calculation masks at various iteration levels, Run 1

Delft3D-WAVE



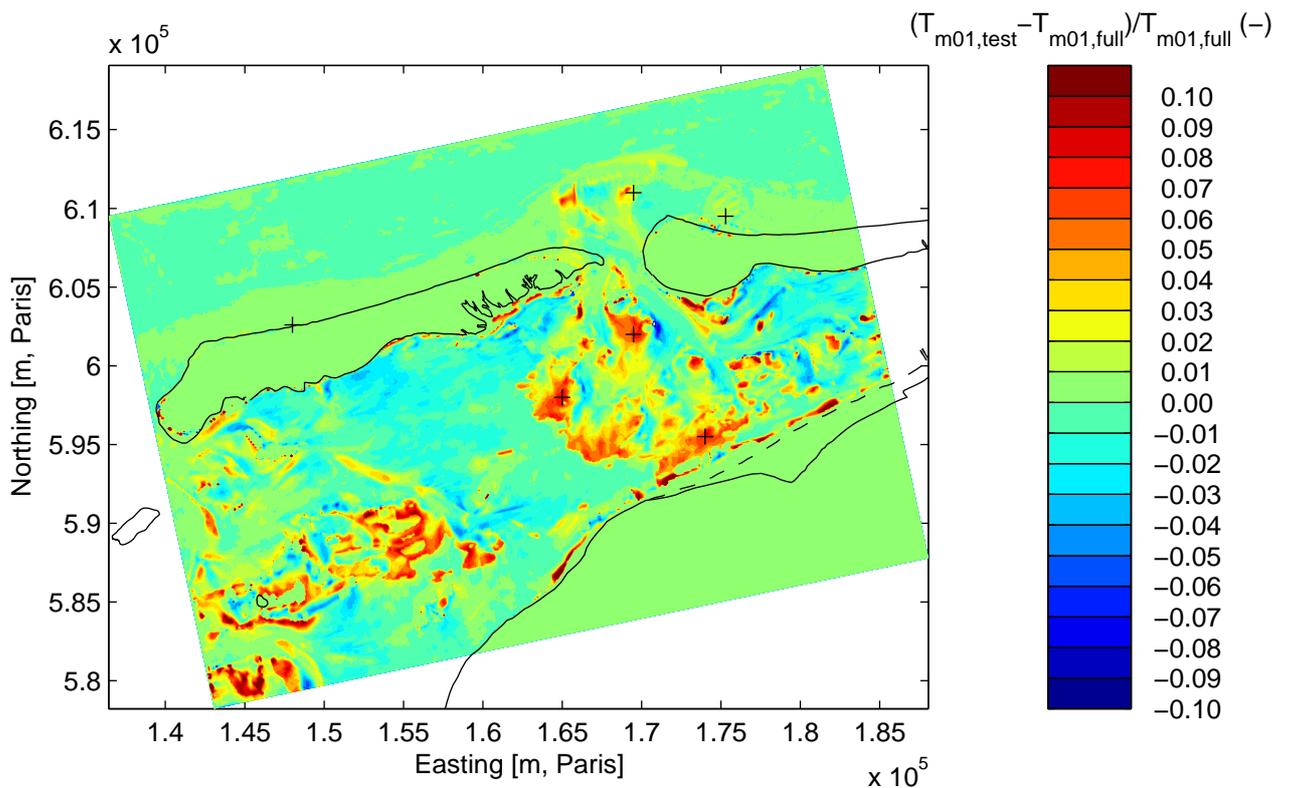
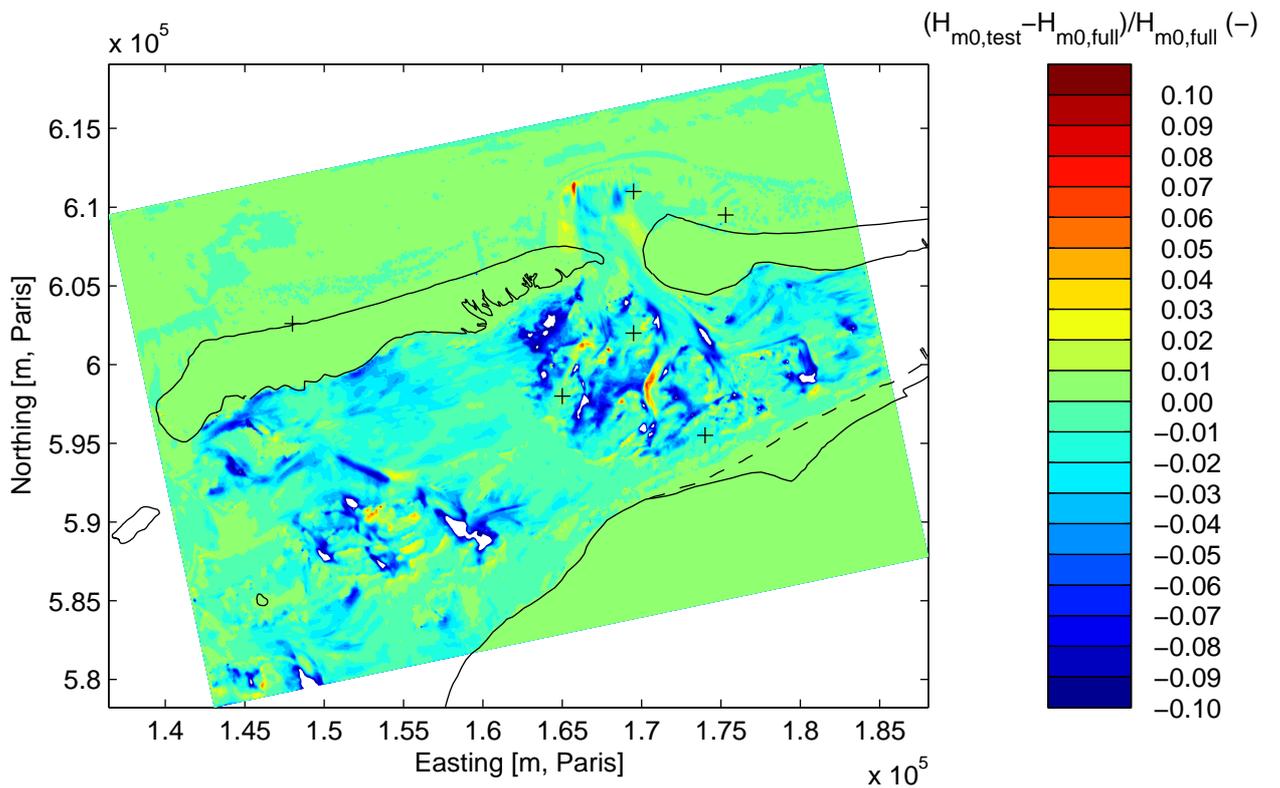
Ameland Zeegat storm of 09/02/2006 at 11:00
 Error relative to full computation after 40 iterations, Run 1

Delft3D-WAVE



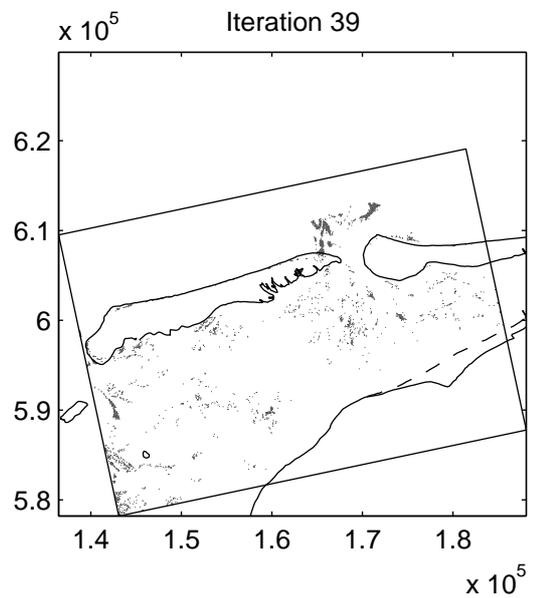
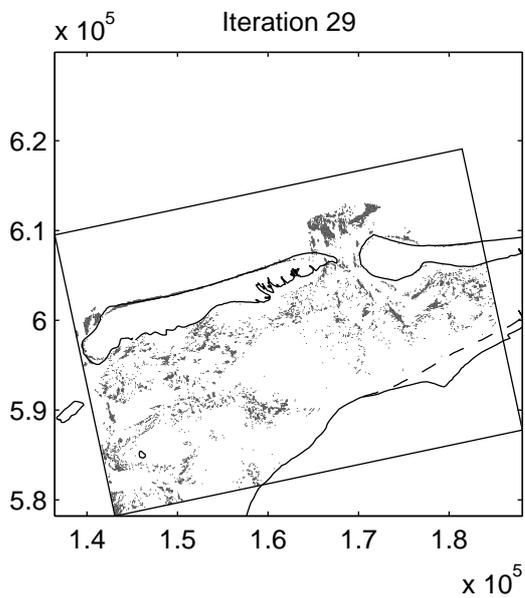
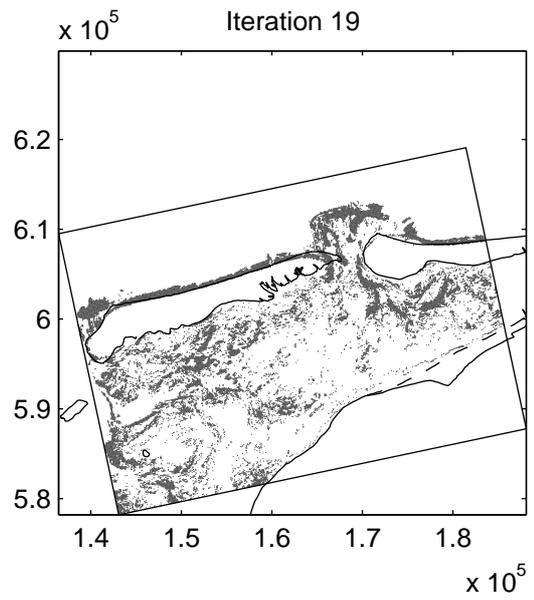
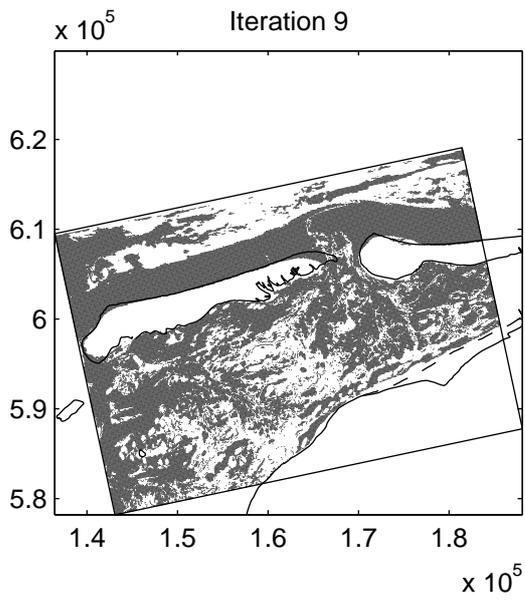
Amelanders Zeegat storm of 2006/02/09 at 11:00
 Calculation masks at various iteration levels, Run 2

Delft3D-WAVE



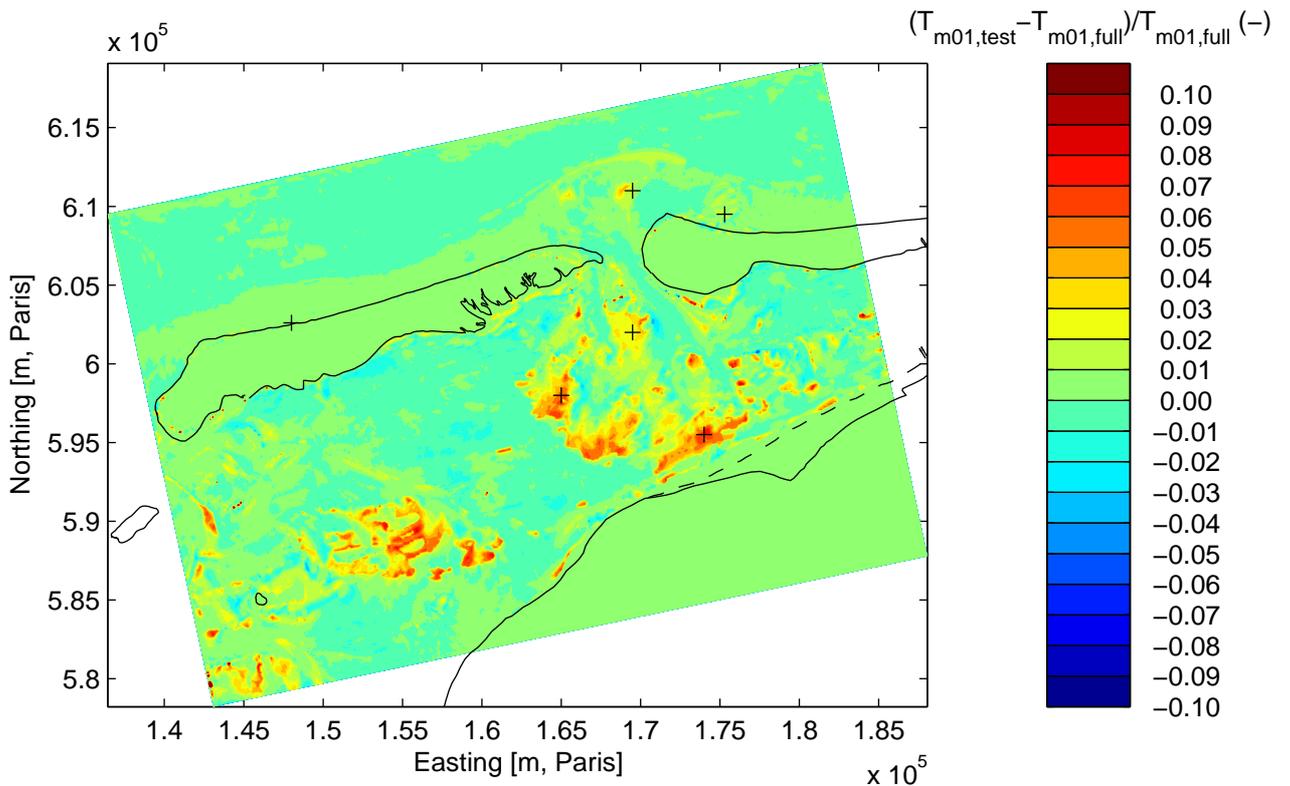
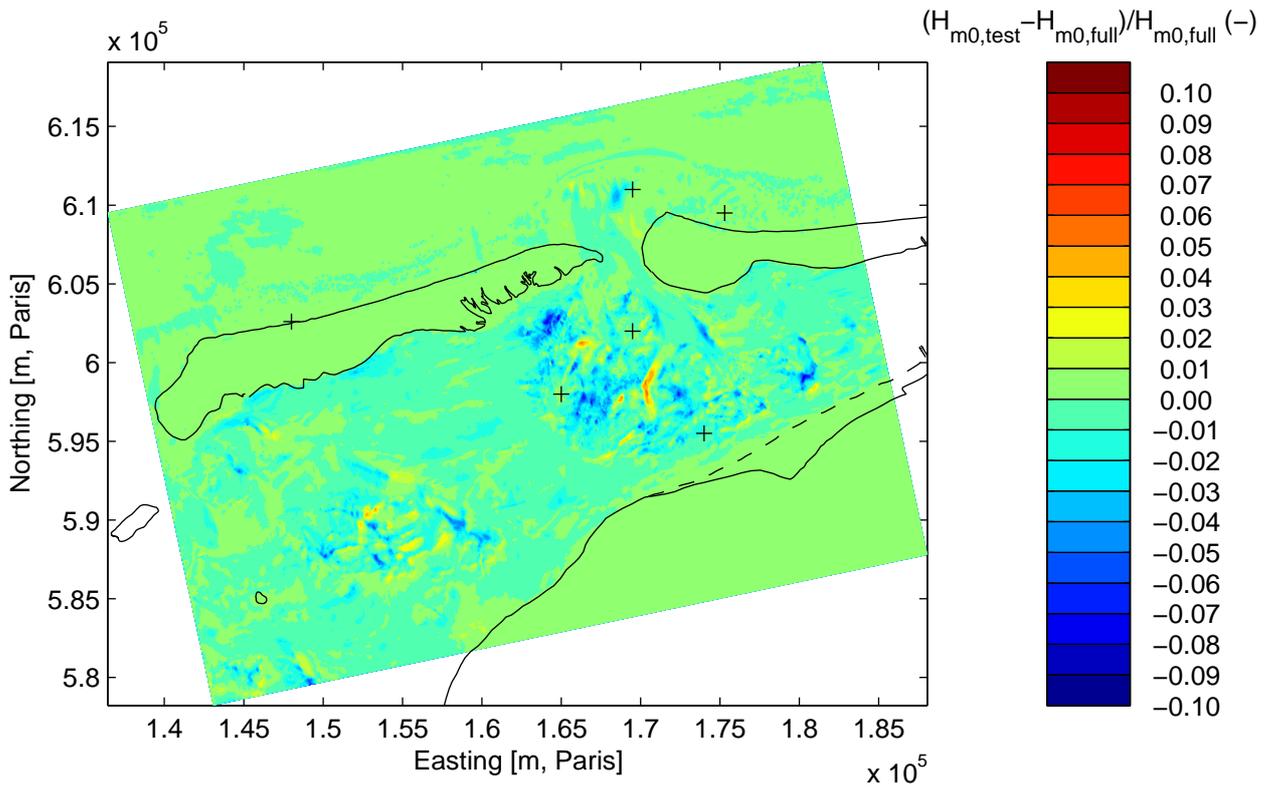
Ameland Zeegat storm of 09/02/2006 at 11:00
 Error relative to full computation after 40 iterations, Run 2

Delft3D-WAVE



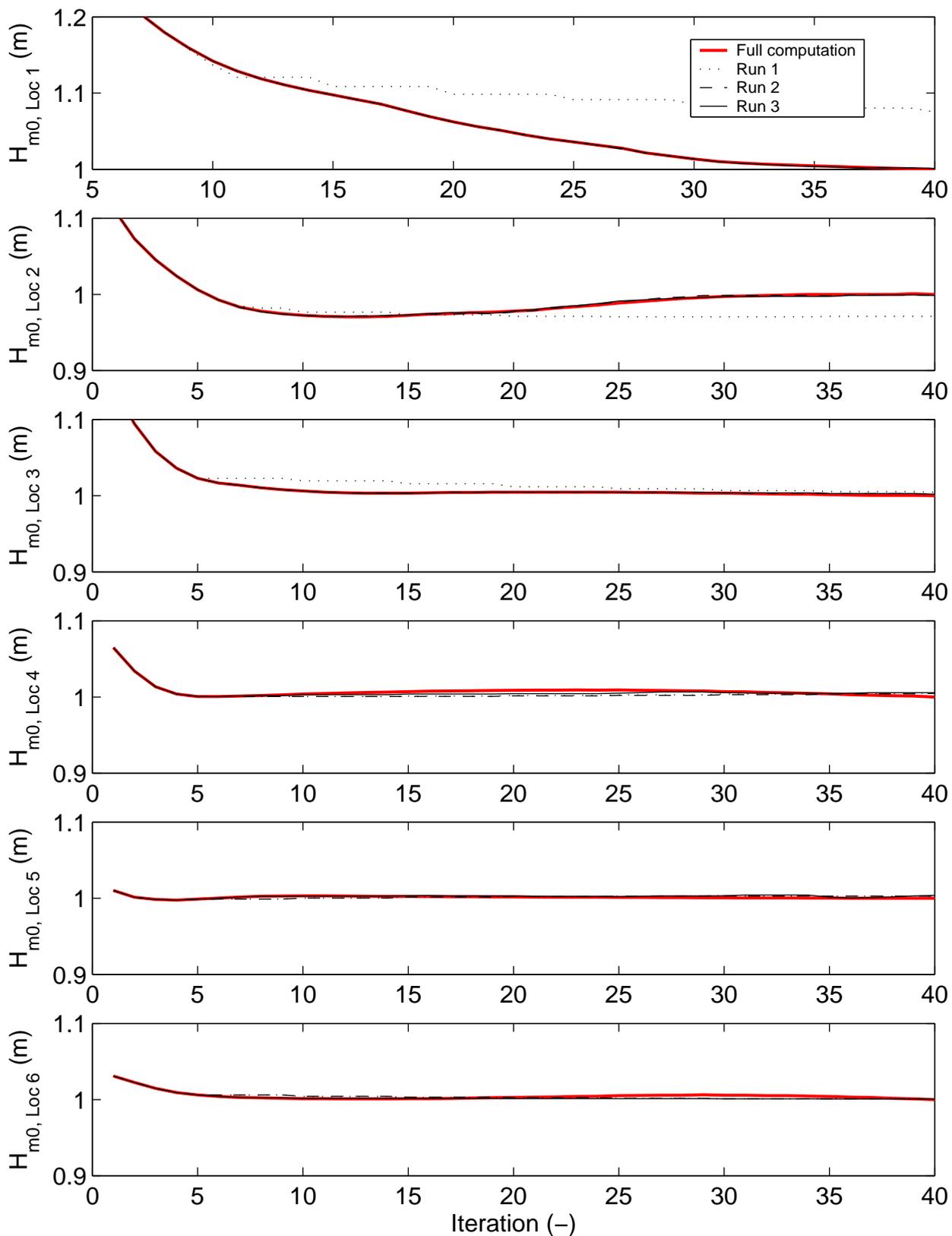
Amelanders Zeegat storm of 2006/02/09 at 11:00
 Calculation masks at various iteration levels, Run 3

Delft3D-WAVE



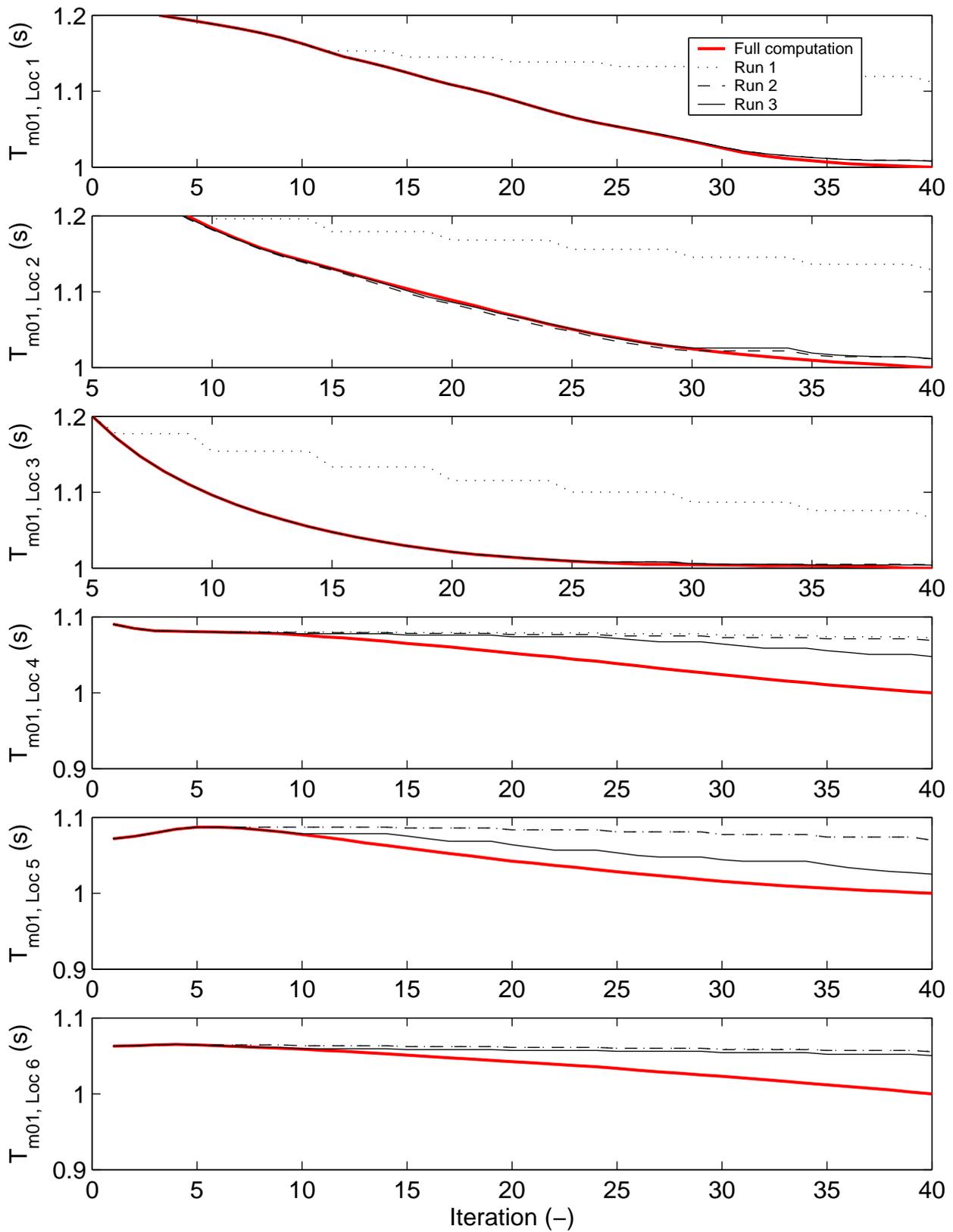
Ameland Zeegat storm of 09/02/2006 at 11:00
 Error relative to full computation after 40 iterations, Run 3

Delft3D-WAVE



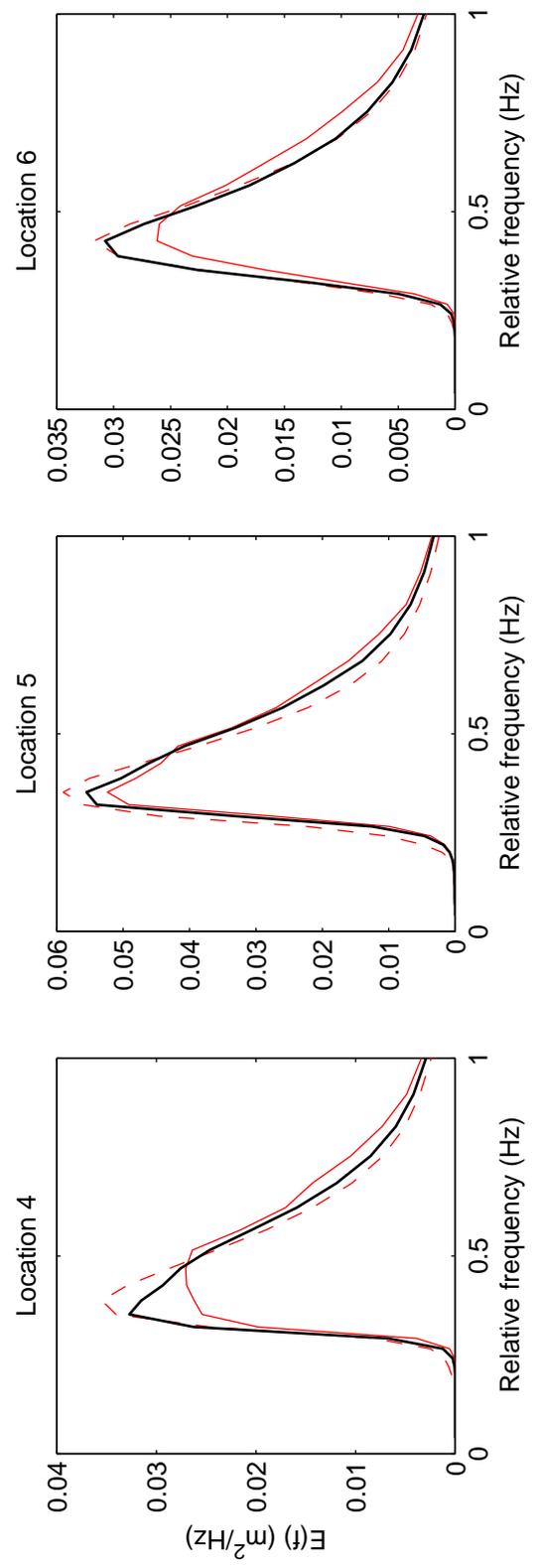
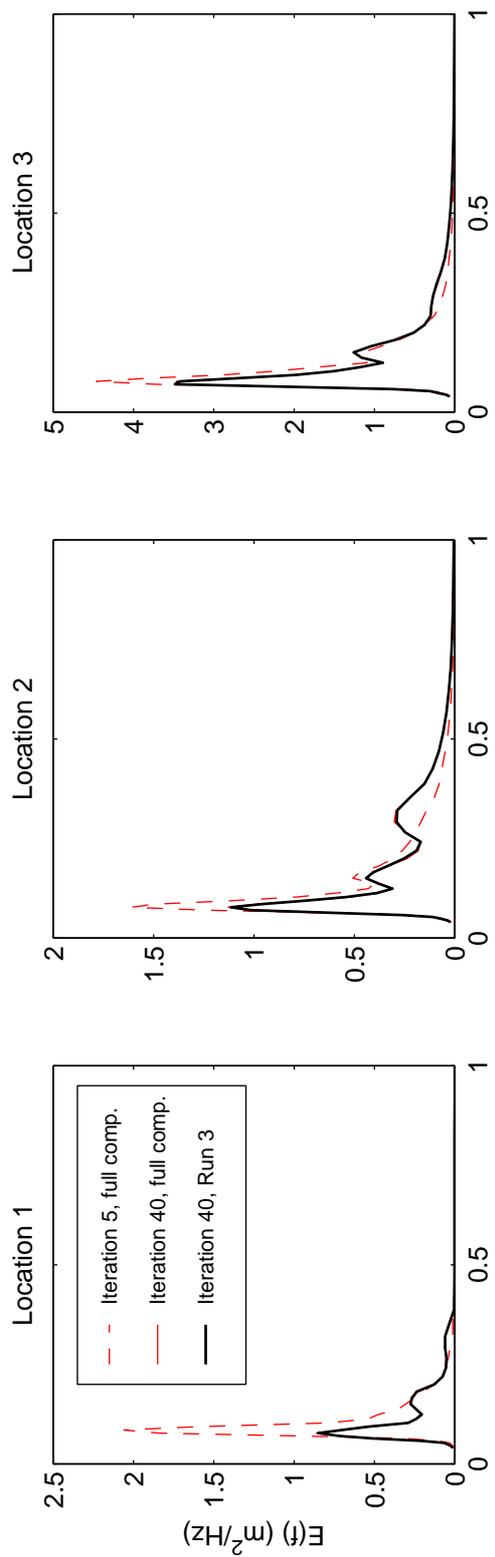
Amelander Zeegat storm of 09/02/2006 at 11:00
 Iteration curves of H_{m0} for Run 1 to 3, normalised with values at iteration 40

Delft3D-WAVE



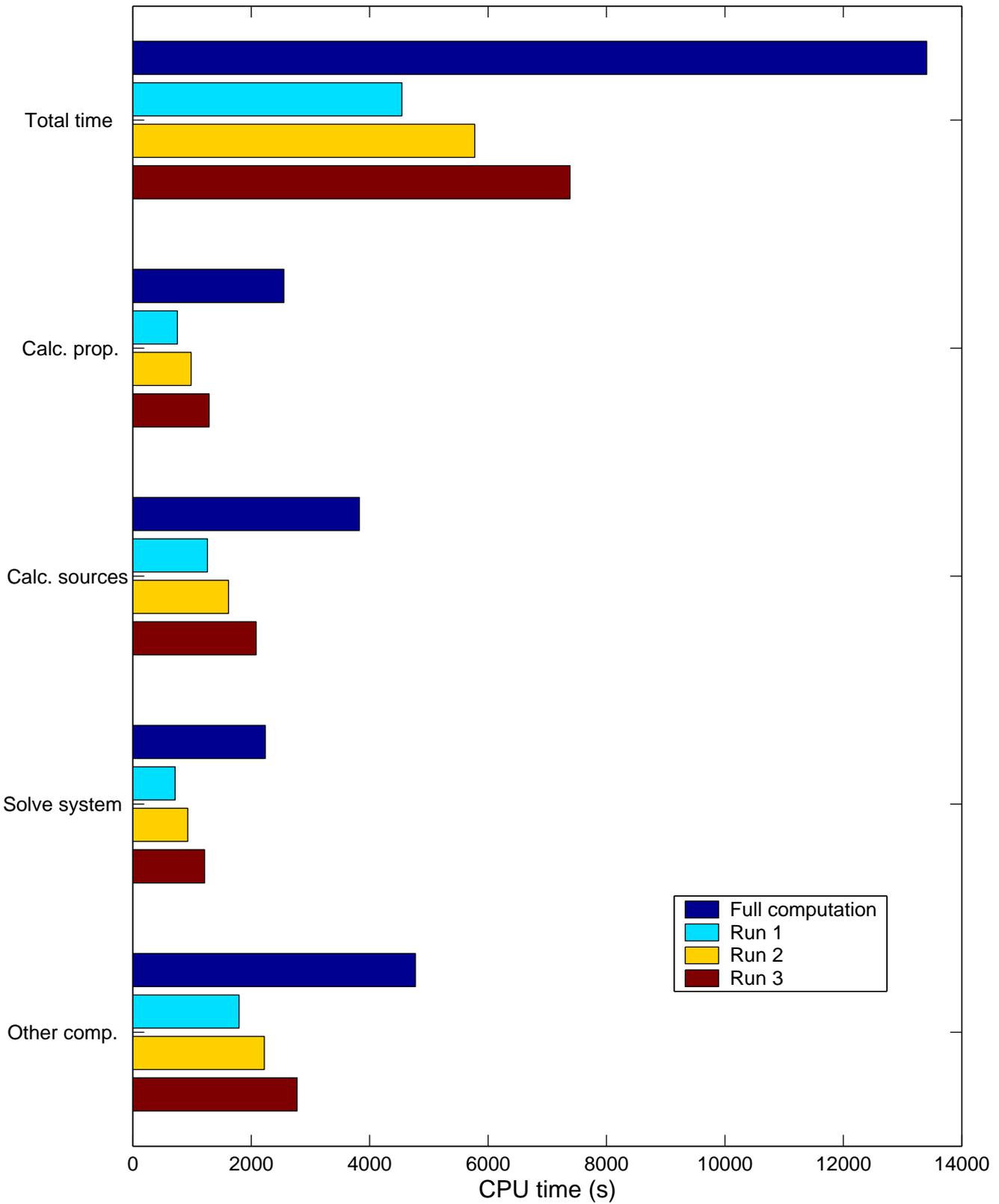
Amelander Zeegat storm of 09/02/2006 at 11:00
 Iteration curves of T_{m01} for Runs 1 to 3, normalised with values at iteration 40

Delft3D-WAVE



Amelander Zeegat storm of 09/02/2006 at 11:00
 Comparison of spectra produced by Run 3 and by full computation

Delft3D-WAVE



Amelander Zeegat storm of 09/02/2006 at 11:00
 Comparison of computational time

Delft3D-WAVE



WL | Delft Hydraulics

Rotterdamseweg 185
postbus 177
2600 MH Delft
telefoon 015 285 85 85
telefax 015 285 85 82
e-mail info@wldelft.nl
internet www.wldelft.nl

Rotterdamseweg 185
p.o. box 177
2600 MH Delft
The Netherlands
telephone +31 15 285 85 85
telefax +31 15 285 85 82
e-mail info@wldelft.nl
internet www.wldelft.nl

