Global tide model with DFlow-FM

Casper Markensteijn

December 23, 2013

Abstract

This thesis is about the beginning of a global tide model in DFlow-FM, a project by Deltares. The first focus lay on finding the best grid that can be used for this model. We tried to grids, a rotated lat-lon grid and a squarely grid with local refinement. After calculating the RMS-value it was found that the local refined grid gave the best results and should therefore be used. The second part of the thesis was about internal tide, how it should be included in the model. This needs future work to improve the results but the way it should be included was found to be a linear friction coefficient.

Contents

Introduction **3**

1 Deltares **4**

2 Model **5**

2.1 D-Flow Flexible Mesh . . . 5

2.2 Why is the model needed? . . . 7

3 Data **8**

3.1 GEBCO . . . 8

3.2 SR95 . . . 8

3.3 My Ocean . . . 9

4 Grid **10**

4.1 Rotated grid . . . 11

4.1.1 Rotation . . . 11

4.1.2 Land . . . 14

4.2 Local refined grid . . . 15

4.2.1 Initial grid . . . 15

4.2.2 Refinement . . . 15

4.2.3 Final grid . . . 16

5 Model description **17**

5.1 Self-attraction and loading . . . 18

6 Internal Tides **19**

6.1 Tidal frequency . . . 20

6.2 Brunt-Väisälä frequency . . . 20

6.3 h . . . 21

7 Results and Discussion **22**

8 Recommendations **28**

9 Acknowledgments **29**

A Rotated grid generation **31**

A.1 Create grid . . . 31

A.2 Rotation . . . 32

A.3 Remove land . . . 32

A.4 Spherical cosine and tangent . . . 33

B Bathymetry **34**
B.1 GEBCO 34
B.2 Bathymetry grid 35
B.3 Get depth into Net-file 36

C Internal Tides **37**
C.1 Linear friction coefficients 37
C.2 h^2 calculation 38

D After processing **39**
D.1 RMS-value 39
D.2 Plot 40

E Results **42**

Introduction

Deltares works with a lot of different hydrodynamic models. Most of these models cover a part of the world and have need of boundary conditions. The problem is that most of these boundary conditions aren't known, therefore they most likely will be an educated guess. But quality models need better boundary conditions than an educated guess. In order to find these boundary conditions a global tide model would be a very useful tool.

So as part of the study's applied mathematics and applied physics this internship at Deltares was started. For the bachelor thesis a global tide model is created in Dflow-FM. This is software created by Deltares for hydrodynamic simulations. The model should eventually become a tool to provide boundary conditions for Deltares's models. The main goal is to provide Deltares with a working model for Dflow-FM. This model won't be good enough yet to provide Deltares with their boundary conditions but future work is planned to further improve. For now this project was concentrated on two questions:

'What grid should be used in the global tide model?'

'What is the best way to include internal tides?'

1 Deltares

Deltares is an independent Dutch research institute it was founded in 2008 by merging GeoDelft, Delft Hydraulics, parts of TNO-built environment and parts of specialized services RIZA, RIKZ, DWW and Rijkswaterstaat. Its primary fields are water, subsurface and infrastructure with the main focus on deltas, coastal regions and river basins. Most of the work is done in cooperation with governments, companies, universities and other research institutes all over the world. With all these partners Deltares constantly extends its knowledge and aims towards the sustainable enhancement of the living environment, with technological solutions that have the support of society and putting into practice its motto: 'Enabling Delta Life'. The internship described in this report has been done in the Hydrodynamics and Real-Time Systems section. This part of Deltares does research on environmental hydrodynamics in marine and estuarine water systems and provides consultancy to partners in this field. It also monitors and forecasts flood risk, water quality and ecological conditions[1].

[1] http://www.deltares.nl/en

2 Model

2.1 D-Flow Flexible Mesh

As told before the model has been made in the Dflow-FM software, what is the successor of Delft3D-FLOW. The application is used by Deltares for making flow models. Thea Vuik already has made a global tide model in Delft3D-FLOW (See [2]) and one of the main problems she encountered was that Delft3D-FLOW can only use structured grids. But also that a local refinement would help to improve the model, as there is more dissipation on ridges than on deeper water and therefore a higher resolution is wanted on those locations. But in order to do this you need to allow unstructured grids as between two refinement levels you would have triangles as cells instead of rectangles. Therefore has been chosen to start a new project where a global tide model should be created in Dflow-FM so the local refinement can be applied. This model is a hydrodynamic model and in order to run this a driving force is needed. In Dflow-FM this force is the gravitation law of Newton and the equations which are solved are the shallow-water equations (See [13]), given by:

$$\begin{aligned} \frac{\partial h}{\partial t} + \nabla(hu) = 0 \\ \frac{\partial u}{\partial t} + \frac{1}{h}(\nabla(huu) - u\nabla(hu)) = -g\nabla\zeta + \nabla(\nu(\nabla u + \nabla u^T)) + \frac{\tau}{h} - 2\Omega usin(\phi)) \end{aligned} \tag{1}$$

Where:

h	the water height
u	the velocity vector
g	the gravitational acceleration
ζ	the water level
ν	the viscosity
τ	the bottom friction given by:
Ω	earth's rotating frequency
ϕ	the latitude

τ	$= \frac{g}{C^2}\|u\|u$

C	the Chezy coefficient

But for a global model you need more than only the shallow-water equations, as without boundary conditions the initial water level would be a solution and therefore the model would not run. Therefore it should also satisfy the Tide generating potential:

$$\frac{\partial \zeta}{\partial t} + \frac{1}{Rcos(\phi)}\left(\frac{\partial u_1 h}{\partial \lambda} + \frac{\partial u_2 hcos(\phi)}{\partial \phi}\right) = 0 \tag{2}$$

Here λ is the longitude,u_1 and u_2 are respectively the velocity in the ϕ direction and the λ direction and R is the radius of the earth. The gravity is taken as a constant, because the fluctuations are too small to be a big influence on the model.

Furthermore there are some other conditions that are important. First of all the calculations of this model are done in the spherical domain and therefore the ∇ in the differential equation is also in the spherical domain. Secondly, there is no flow through the land-sea boundary and the initial value of the water level is the mean sea level. Finally, the differential equations are solved using the finite volume method.

2.2 Why is the model needed?

As said in the introduction one of the main reasons for the need of the model is that for modeling smaller regions in DFlow-FM (or Delft3D) boundary conditions are needed. At the moment there isn't a model which can predict those boundary conditions. So values which seems likely are taken and after the model has been run those will be calibrated in order to get better results. But those better results doesn't mean that the boundary conditions where the reason for a misfit between the results and the measurements. It could be that another part of the model isn't working right and is causing the bad results. In order to be sure that the models are working fine the boundary conditions should be predicted in advance. One of the ways to do this is to make a global tide model. Such a model will predict the velocities and waterlevels at oceans, where most of the boundary conditions are wanted. These boundary conditions make it easier to create smaller models and takes less work because there isn't any calibration needed. Also this will eventually make the models better, as other parts of the model will be seen as the cause for bad results.

A second reason for making the model is found in sensitivity studies of models. Some models use for instance a change in waterlevel in order to model a change in weather on parts that are outside the model. That kind of changes can be tested by changing the weather conditions in a global model and than see what the influences are on the points of interest.

Eventually the model could also become a global storm-surge model. In order to become this someone should improve the model by implementing global important wind forcing. This way it could eventually predict weather conditions, especially storms, and it could also predict what the influences on different parts of the world would be.

3 Data

In the making of this model three different datasets have been used. First of all the GEBCO data has been used for the bathymetry. Secondly, the SR95 data has been used to verify the results. Last, the MyOcean datasets has been used to calculate the Brunt Väisälä frequency. In the next subsections those datasets will be explained a bit.

3.1 GEBCO

The General Bathymetric Chart of the Oceans[2], or GEBCO, is an international group of experts who develop a range of bathymetric data sets. One of those data sets is a bathymetric grid. This grid has a resolution of $\frac{1}{60}^{\circ}$ which means that every degree of longitude or latitude is divided into 60 pieces. The grid has been put together by merging many different regional grids. In order to use the data on less powerful pc's Matlab was used to interpolate those data to a grid with lower resolution of $\frac{1}{16}^{\circ}$.(See Appendix B.1) For a more detailed description of the dataset the user guide can be read (See [12]).

3.2 SR95

SR95 is an updated version of the global tide model described by Schrama and Ray (See [10]). The tidal solution is derived by a harmonic method and is solved for five constituents: $M2, S2, N2, O1, K1$. Next to that $P1, Q1$ and $K2$ constituents were adopted from the FES94.1 model. Finally 16 minor constituents were included by linear inference from the major constituents in order to compute tidal height predictions. The altimeter data that has been used in this model is from both TOPEX and POSEIDON altimetry. For a detailed description of the model the original paper by Schrama and Ray should be read.

The data from this model is used in order to investigate the quality of the results presented in this report. In order to do this, the data for January first and second was compiled and, as will be seen later in this report, it was used to calculate the RMS-values between SR95 and the model described in this report.

[2] http://www.gebco.net/

3.3 My Ocean

In order to calculate the Brunt-Väisälä frequency the derivative of the water density in respect to the depth is needed. The density of seawater depends on the temperature and salinity of it. Those can be found in the My Ocean dataset for different depths. Therefore the density for those depths can be calculated an from there also the derivative. The dataset has been created by the Global Monitoring for Environment and security program (GMES). The sets are mainly created by measurements of satellites and In-situ techniques (for instance drifting buoys). This has created two global sets that has been used, one for temperature and one for salinity.[3].

[3] http://myocean.eu

4 Grid

An important question for this model was what a good grid was to use in DFlow-FM. First a few requirements for the grid:

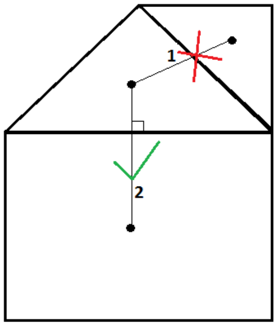


Figure 1: A picture introducing orthogonality, the two cells connecting at 2 are orthogonal while the two cells connecting at 1 aren't.

- The grid has to be orthogonal (figure 1).
- The cells should not have small angles.
- A node can't have to many links connected to it.
- The sides of the cells may not defer to much.
- It is preferred to have 4 sides on each cell.
- More resolution on the shelf is wanted because there is a lot of dissipation.

Looking at these requirements a standard latitude-longitude grid is not possible because that would results in a point at the North pole with an amount of links equal to 360 devided by the resolution of the grid connected to it. In order to remove this point from our model a standard lat-lon grid was used but has been rotated so the North and South pole of the grid are both on land and can be removed. How this grid was generated will be explained in the next paragraph.

The problem with this grid is that a high resolution is wanted in order to get good results, but a high resolution on the whole grid is not computational practical, because there would also be a high resolution on parts that do not need it. That is why it is wanted to refine the grid on the points that are important while remaining a lower resolution on points that don't need a higher one. In essence this means that we want a high resolution on low depths and a lower resolution on the higher depths. How this was performed will be explained in paragraph 4.2.2.

4.1 Rotated grid

In order to rotate a grid we first need a starting grid. The starting grid is a standard lat-lon grid with latitude between $-90°$ and $90°$ and longitude between $-180°$ and $180°$. For a qualitative model the grid needs a high resolution, but it must also remain computational practical. That's why a grid resolution of $\frac{1}{2}^{\circ}$ was chosen, this means on every degree latitude or longitude 2 cells are located.

4.1.1 Rotation

For the rotation of the grid a method called stereographic projection as explained by [1] has been used. This method preserves angles and therefore also the orthogonality of the grid. One problem is that the poles will be located on land but not 180° apart, which means that the grid will be stretched, this is something that should happen as less as possible. Therefore the mapfactor is also calculated, but first the rotation itself will be explained.

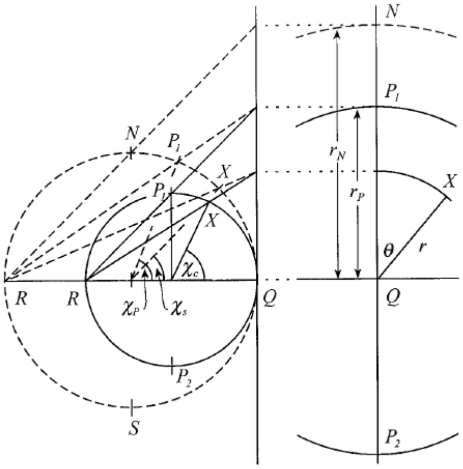


Figure 2: Points of the small sphere are stereographically projected onto the plane and then reprojected onto the original sphere.

The method uses that the two rotated poles can construct a new smaller sphere, starting with a standard lat-lon grid. This sphere has one tangent point (called Q) with the original sphere (figure 2). When the coordinates of this point has been calculated (λ_Q, ϕ_Q) (although ϕ'_Q will be used which is $90° - \phi_Q$), together with the half-angle between the two new poles (χ_P), the angle between the latitude of the new north-pole and the line between the new poles (α) and the angle between the latitude of Q and the line between the new poles (β) (figure 3), all points of the original grid can be rotated to the new grid.
How this process works will now be explained. In order to calculate previous stated angles the spherical cosine and tangent functions will be used, which are

defined as follow:

$$
\begin{aligned}
a = f_c(A,b,c) = cos^{-1}(cos(A)sin(b)sin(c) + cos(b)cos(c)) \\
B = f_t(A,b,c) = atan2(sin(A), \frac{sin(c)}{tan(b)} - cos(c)cos(A))
\end{aligned}
\tag{3}
$$

Here f_c is the spherical cosine and f_t is the spherical tangent function. Those are in reference to a spherical triangle with vertices (A,B,C) and opposite sides (a,b,c). With these functions the angles and coordinates can be calculated:

$$
\begin{aligned}
\chi_P &= \frac{1}{2} f_c(\lambda_1 - \lambda_2, \phi'_2, \phi'_1) \\
\alpha &= f_t(\lambda_1 - \lambda_2, \phi'_2, \phi'_1) \\
\lambda_Q &= \lambda_1 - f_t(\alpha, \chi_P, \phi'_1) \\
\phi'_Q &= f_c(\alpha, \phi'_1, \chi_P) \\
\beta &= f_t(\alpha, \phi'_1, \chi_P)
\end{aligned}
\tag{4}
$$

Here (λ_1, ϕ_1) and (λ_2, ϕ_2) are respectively the coordinates of the new North- and South-Pole in the (lon,lat) domain. With these values the new coordinates for every point X (λ_c, ϕ_c) of the lat-lon grid can be calculated. First the distance (χ_c) between Q and X and the angle between the latitude of Q and χ_c (θ_c) on the small sphere are calculated (distance given in radians):

$$
\begin{aligned}
\chi_c &= f_c(\lambda_c, \phi'_c, \frac{\pi}{2}) \\
\theta_c &= f_t(\lambda_c, \phi'_c, \frac{\pi}{2})
\end{aligned}
\tag{5}
$$

Than those values are transformed to the corresponding values θ_s and χ_s on the original sphere:

$$
\begin{aligned}
\chi_s = 2 * arctan(r_P * tan(\frac{\chi_c}{2})) \\
\theta_s = \theta_c + \beta
\end{aligned}
\tag{6}
$$

Here r_P is the radial distance of the new north-pole to Q in the projection plane and the first equation is derived from $r = r_P tan(\frac{\chi_c}{2}) = r_N tan(\frac{\chi_s}{2})$(figure 2). Where r_N is taken 1 to simplify the expression, this only makes the original sphere a sphere with radius 1, so it does not influence the rotation. It neither has an influence on the model, because Dflow-FM only needs the longitude and latitude coordinates and not the radius.

With these values the new coordinates for point X are calculated:

$$\begin{aligned} \lambda_s &= \lambda_Q + f_t(\theta_s, \chi_s, \phi'_Q) \\ \phi_s &= \frac{\pi}{2} - f_c(\theta_s, \chi_s, \phi'_Q) \end{aligned} \tag{7}$$

In this method the grid has another starting longitude than $-180°$, it starts at ϕ_Q. For practical reasons it should be on $-180°$ so all grid points that have a longitude smaller than $-180°$ are moved by 360° while all points with a longitude bigger than 180° are moved by $-360°$. This way it does not matter what new 'North'- and 'South' pole is chosen, the grid has always longitude between $-180°$ and 180°.

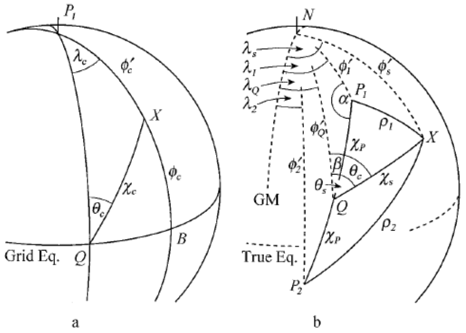


Figure 3: The angles on the smaller sphere and on the original sphere that are needed for the rotation of the grid. Here GM is the Greenwich Meridian.

At last a value called map factor ($\frac{1}{m}$) is calculated. This value gives for every gridcell the magnification factor in respect to the standard lat-lon grid. This factor needs to be as close to one in order to get the best grid, as when, for instance, a factor of 2 is reached there would be a cell with half the resolution of the chosen grid-resolution. This value is calculated as follow:

$$\frac{1}{m} = \frac{\partial \chi_s}{\partial \chi_c} = \frac{r_P sec^2(\chi_c/2)}{1 + r_P^2 tan^2(\chi_c/2)} \tag{8}$$

To have this value as close to 1 as possible, the 'North' and 'South' pole should have a radial distance between them as close to 180° as possible. So in order to meet that requirement the 'North' pole is put on Canada and the 'South' pole remains on Antarctica, but is shifted a bit to the North and East. This gives a mapfactor between 0.84 and 1.19. In figure 4 you can find a grid with The north pole on Spain and the South pole on the ocean with a resolution of 15°, here you can see the effect of the stretching.

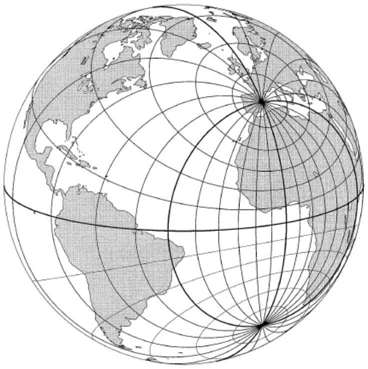


Figure 4: an example of a rotated grid with a resolution of 15°.

4.1.2 Land

The grid, generated as explained before, still contains land. But those points of the grid that are on land are of course not important for an ocean model. They would only use more computer time while they have no influence on the oceans tides. That's why the land is removed. This is done by interpolating the bathymetry data out of GEBCO to the grid and than remove all points of the grid which have a height bigger than 0 (or mean sea-level). There is a small problem with this method as there are parts of land with which have a height smaller than zero. Those values are now still included in the model, but this are very small parts of land and are thus not a big influence on the model and are therefore not removed by another method.

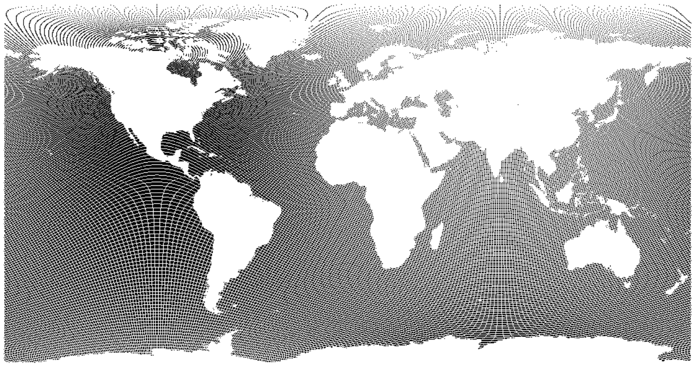


Figure 5: The resulting grid with the land removed with a solution of $\frac{1}{2}^{\circ}$.

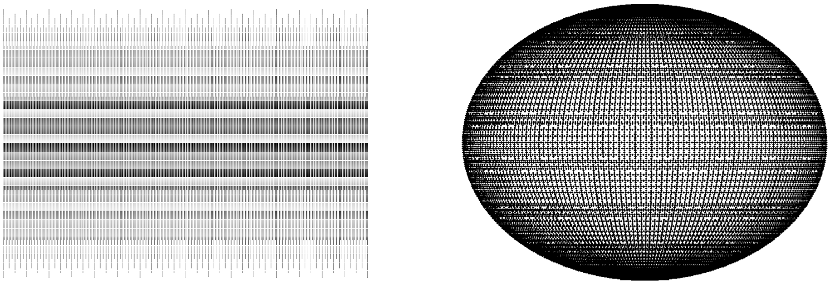
The grid was generated using Matlab, the scripts can be found in Appendix A and the resulting grid in figure 5.

4.2 Local refined grid

The next thing to do was to refine the grid on parts of importance. In Dflow-FM an option is created to do that, this will be explained in paragraph 4.2.2. In order to use this option the grid should have cells which are as squarely as possible. But the rotated grid has many cells which are not squarely and therefore the refinement can not be used on that grid. Therefore a new initial grid was generated as explained in the next paragraph.

4.2.1 Initial grid

As the initial grid should be as squarely as possible a lat-lon grid is not a good choice. Although in spherical coordinates it has perfectly square cells it does not when looking in Cartesian coordinates. Therefore a uniform grid has been used and has been refined, but not in the same way on all directions. The refinement depends on the position of a cell on the earth and was done so that all cells have width of about $50km$. But because of the bending there are cells with different width and also cells with hanging nodes, which means that it has a node which is not connected in at least one direction. In figure 6a the initial grid can be found and in figure 6b can be found how it looks on a sphere.



(a)

(b)

Figure 6: The initial grid for the refinement method with in (a) a flat display and in (b) it is displayed on a sphere.

4.2.2 Refinement

This initial grid has been used in Dflow-FM where the refinement option refines the grid based on the bathymetry in all directions at the same time. In order to do this the minimum bottom level ($|z|_{min}$) of the cell will be determined. Using this the wave courant number, which looks as follow:

$$Courant = C\Delta t_{max}/h \tag{9}$$

will be determined. Here $C = \sqrt{g|z|_{min}}$ is the wave speed, Δt_{max} a maximum time step and h the minimum width of the cell.

The cell will be refined by a factor 2 in all directions when $Courant < 1$ and $|\frac{h}{2} - h_{min}| < |h - h_{min}|$. Here $\frac{h}{2}$ is the new gridcell width and h_{min} is a minimum width, chosen by the user. After this has been done some cells will be refined in order to get proper nesting, which means that there is only one refinement level difference between neighboring cells. After the user says that the refinement should be finished, or after there are no more cells which can be refined according to previous stated conditions, the hanging nodes will be connected to the neighboring cell nodes, which results in triangles. Doing this results in a grid as in figure 7. Some parts of the world are clear in this grid, as for instance the Rocky Mountains and the Alps, because these are places with bigger heights and therefore the refined isn't applied on those points.

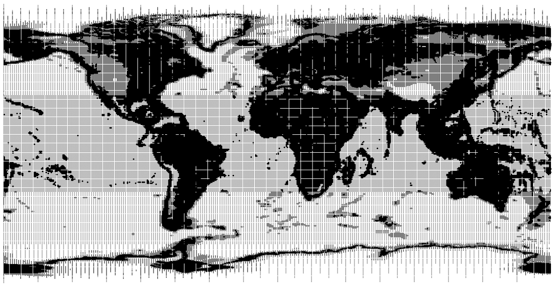


Figure 7: The refined grid, with the land still in it.

4.2.3 Final grid

The minimum width of a cell has been chosen to be $5km$. So the refinement was applied till there were no more cells to refine. This gave a grid with about 1,3 million points. But it still contained land. This was removed as follow. First a polygon was made with a rough sketch of the continents. All points that were within this polygon were removed and then an option of Dflow-FM was used to remove all points where every neighboring point has a height of more than $10m$ in respect to the mean sea level. Now the grid has about 850.000 points left which is still computational practical. This grid can be found in figure 8.

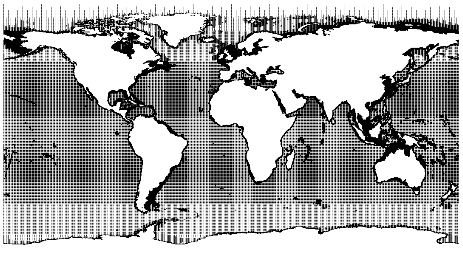


Figure 8: The final local refined grid with the land removed.

5 Model description

Usually DFlow-FM is applied to models that contain coastal areas, estuaries or rivers. Those models always contain boundary conditions which makes the differential equation of equation 1 sufficient but for a global model there are no boundary conditions therefore you need the tidal forcing (equation 2) in order to get the model started, otherwise the initial condition is a perfect solution and therefore the model won't change. Other things that become important on a global model are the self-attraction and loading and the internal tides of the ocean. They will respectively be explained in paragraph 5.1 and chapter 6.

In order to run a good model some parameters are needed, the important ones are specified in table 1.

Starting time	22-12-1991 0:00:00
End time	03-01-1992 0:00:00
Time step	$5min$
Friction coefficient	62.651
Friction kind	Chezy
Gravity	$8.829m/s^2$
Initial water level	$0m$
Observation interval	$30min$
Mean water density	$1024.0164kg/m^3$

Table 1: Some of the parameters needed for the model

The starting time is more than one week before the first of January 1992, because the initial value for the water level is $0m$. So DFlow-FM needs some time to get the real water level values and tests have proven that about a week is enough so that the initial water level has no influence on the outcome.

The model values are being stored every 30 minutes of model time. This time interval has been chosen because the SR95 dataset has also been compiled with a 30 minutes interval. The values that will be stored are the values on a 1° lat-lon grid, where the land has been removed, of points that has been added to the model as observation points.

The friction coefficient in 1 is the bottom friction coefficient that has been chosen. With this value we have a $c_d = \frac{g}{C^2}$ in equation 1 equal to 0.0025 which corresponds to the value specified in [7]. But this value has only been used in the run of the model with the internal tide included. The other two runs, that have been done with the 2 different kind of grids, have been run with the friction coefficients that have been calculated in [2]. Those values have been used because they gave reasonable results. Therefore they were a good option for a run to investigate what grid was better.

5.1 Self-attraction and loading

Not only the ocean responds to tidal forcing. The solid earth and the atmosphere do as well. The atmospheric tides are primarily driven by solar diurnal heating of the atmospheric column, because the gravitational tides are very small. But even those are still small, a change of tides of around $1cm$ of water maximum, and are therefore neglected. But the solid earth tides are quite important for calculations and applications where gravity is included. The solid earth responds to the tidal forcing as an elastic body and has therefore a lot of components. For the ocean tides the influence is mostly seen in a reduction of the amplitude of the equilibrium tides. As the earth and oceans deform in the same direction under the tidal forcing of the moon and sun.

This is not an one-way effect though, because the ocean tides have influence on the solid earth below as well. This loading due to ocean tides deforms elastically and extend to continents and are important to gravimetry. These effects are called self-attraction and loading and should be included in the global tide model. In order to fully insert those influences there should be a convolution of the ocean tide with the SAL Green's function (See [8]). This convolution should be done each time-step and is therefore not computational practical. As this was not the main focus of this thesis we used an approximation which was provided by [5], which lowers the tide by a factor $1 - \beta$ where β has been chosen 0.1 following [9]. In the global tide model the factor has been included through the gravitational acceleration. Instead of $g = 9.813 \frac{m}{s^2}$, $g = 0.9 * 9.813 = 8.829 \frac{m}{s^2}$ has been used, as this has the same effect as correcting the tides. This can be seen in equation 1 in the first part of the right-hand-side. Although it is a very crude approximation, as no scalar fits all parts of the ocean and because the scalar itself is already an approximation, it is sufficient for the current phase of this model.

6 Internal Tides

Internal tides are in essence internal waves at a tidal frequency. They are for instance generated when a stratified fluid is forced to undergo vertical deflections under gravity by moving over sloping bathymetry. An example can be found in the atmosphere were the flow of air over mountains generates internal waves which are important for the vertical momentum. In the oceans we can find a similar mechanism around ridges, narrow straits and continental shelfedges. Here the tidal currents can interact with the sloping bathymetry and generate internal waves. The transfer of energy from long barotropic tides to much shorter internal waves is important to the mixing and energy of the interior of the ocean. Although internal waves have been known for 200 years, it was only late in the 20^{th} century that the importance of internal waves was recognized for the energy dissipation in deep ocean (See [11]). Until then the common believe was that they only mattered in coastal regions, but following [5] about one-third of the tidal energy dissipation dissipates within the internal tides.

In the same article an expression for the energy dissipation of the internal tides can be found:

$$E_{int} = \frac{\sqrt{\omega^2 - f^2}}{2\omega} \alpha \rho_0 \kappa h^2 N |u|^2 \tag{10}$$

Where ω is the tidal frequency and $f = 2 * \Omega sin(\phi)$, with $\Omega = 7.292115 * 10^-5 rad/s$ and ϕ is the latitude, is the Coriolis parameter. α is a proportional constant which is taken 1.1 following [11]. $\rho_0 = 1024.0164 kg/m^3$ is the mean ocean density and $\kappa = \frac{2*\pi}{10000} m^{-1}$ is the characterizing wavenumber found optimal by [7]. h the characterizing ocean depth and N is the Brunt-Väisälä frequency.

In Dflow-FM only bottom friction is included. An expression for the energy dissipation of the bottom friction is as follow:

$$E_{Bottom} = rho_0 \frac{g}{C^2} |u|^3 \tag{11}$$

Where C is the Chezy coefficient. As this has been included as (See equation 1):

$$\tau = c_d ||u|| u \tag{12}$$

Where $c_d = \frac{g}{C^2}$ for Chezy. It can be seen that in the equation for internal tides there is a u less, because the other parts together have a unit of m/s. This implies that a good way to implement this in the model would be to have another friction coefficient with unit m/s which is implemented as follow:

$$\tau_2 = c_d u \tag{13}$$

Where c_d is equal to:

$$c_d = \frac{\alpha}{2} \frac{\sqrt{\omega^2 - f^2}}{\omega} \kappa h^2 N \tag{14}$$

In order to calculate this c_d the Tidal frequency, N and h^2 should be calculated. How this has been done will be explained in the next paragraphs.

6.1 Tidal frequency

The tidal frequency is not a constant as the different tidal constituents have different frequency's. It is not computational practical to work with different frequency's so the only difference that is made is between diurnal tides and semi-diurnal tides. Those have frequency's of respectively $\omega = 7.026285 * 10^-5 rad/s$ and $\omega = 1.405257 * 10^-4 rad/s$. Propagating internal tides occur equatorward of the latitude where $\omega = f$ but bottom-trapped internal tides occur at latitudes where $\omega < f$ (See [7]). Because of this semi-diurnal tides occur at frequency's poleward of 74.5° and diurnal tides poleward of 30°.

6.2 Brunt-Väisälä frequency

As told before N is the Brunt-Väisälä frequency. This frequency is the angular frequency of the vertical oscillation in a statically stable environment. It is derived by making a Newton's approximation on the following differential equation[4]

$$\rho_0 \frac{\partial^2 z'}{\partial t^2} = -g(\rho(z) - \rho(z + z')) \tag{15}$$

Where z is the depth, ρ_0 the mean ocean density and g is the gravitational acceleration defined positive. After the approximation we move ρ_0 to the right-hand-side and we get:

$$\frac{\partial^2 z'}{\partial t^2} = \frac{g}{\rho_0} \frac{\partial \rho(z)}{\partial z} z' \tag{16}$$

[4] http://en.wikipedia.org/wiki/Brunt-Väisälä_frequency

Which gives us the following solution:

$$z' = z'_0 e^{\sqrt{-N^2}t} \tag{17}$$

where:

$$N = \sqrt{-\frac{g}{\rho_0}\frac{\partial \rho}{\partial z}} \tag{18}$$

This means that for positive N the fluid isn't stable and for negative N z' oscillates.

6.3 h

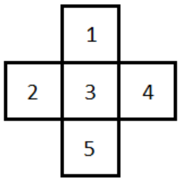


Figure 9: The cell figuration for the formula of h.

h is the characterizing height of a cell. There are some authors (for instance in [3] or [4]) who give ways to calculate this parameter. But all those articles are very theoretical and use for instance a flat sea floor with one bump in it, but this is not a realistic representation of the reality, as no part of the sea floor can be considered flat. Therefore a way had to be found to calculate the value of h. For cells as in figure 9 the value was calculated with equation 19.

$$h = \frac{1}{2}(\sqrt{\frac{(h_1 - h_3)}{d(x_1, x_3)}^2 + \frac{(h_5 - h_3)}{d(x_5, x_3)}^2} * d(x_1, x_5) + \sqrt{\frac{(h_2 - h_3)}{d(x_2, x_3)}^2 + \frac{(h_4 - h_3)}{d(x_4, x_3)}^2} * d(x_2, x_4)) \tag{19}$$

Here $d(x_i, x_j)$ is the distance between two cell centers and h_i is the depth of the cell. This equation has been formed by taking the rms value of the derivatives in the horizontal direction and in the vertical direction and than multiply those values with half the distance between the corresponding cell centers.

7 Results and Discussion

As said before the model has been run in three different ways. The first two runs where done to verify that the local refined grid indeed leads to better results than the rotated grid. This was done by using a file with frictioncoefficients that has been explained in [2]. In order to test the accuracy of the results a representative mean value of the difference between the model results and the SR95 dataset was needed. But if the mean was calculated this would not have let to a value that is representative, as there are positive and negative values which could lead to values canceling each other out. The RMS-value on the other hand uses the squares of the differences and neglects therefore the negative and positive values. This is therefore a quantity that is used in many different kind of studies where you have negative and positive values that have influences, but where the sign doesn't matter. For an accurate model the RMS-value should be as low as possible on all times, because that means that the model is close to the SR95 dataset. Some of the RMS-values that have been calculated can be found in table 2.

Time	Rotated grid	Local refined grid
01-Jan-1992 00:00:00	15.4616 cm	12.1744 cm
01-Jan-1992 06:00:00	16.3238 cm	12.5019 cm
01-Jan-1992 12:00:00	16.7981 cm	12.5475 cm
01-Jan-1992 18:00:00	14.4946 cm	11.7414 cm
02-Jan-1992 00:00:00	16.1780 cm	12.9954 cm
02-Jan-1992 06:00:00	17.1368 cm	12.9984 cm
02-Jan-1992 12:00:00	16.6466 cm	12.6641 cm
02-Jan-1992 18:00:00	14.4275 cm	12.7026 cm

Table 2: RMS-values for some times of the first two model runs.

From the table can be seen that the local refined grid is indeed the grid that gives the best results. Now the question is if there is a good chance that a better grid can be found with a reasonable amount of points. The local refined grid has an resolution of about 1/20° on the shallowest parts while remaining a resolution of about 1/2° on the deeper parts of the ocean. In order to improve the grid you could do two things, but both aren't really computational practical. First thing that could be done is to have a high resolution on the entire grid, but as said before this would result in a high resolution on parts that don't need it, for instance the deep oceans. And another problem is that a grid of 1/8° would have about 4 million points, while the local refined grid has better resolution on the important parts and it only has about 850.000 points, so this isn't really an option. The other way the grid could be improved is by refining it even more, but the question is if this would result in a grid with a noticeable improvement in the RMS-value. And if you want to do this you will come across some problems with the bathymetry set.

As you would have to get a set with a resolution of 1/60° into DFlow-FM, which you can't do with most computers. So you would have to use a supercomputer even to create the grid and that is not something you want. So in my opinion the refined grid that has been used is about as good as it gets and therefore future work should not focus on the grid.

The third run of the model was done with the grid that was considered the best, so with the local refined grid. In this run the uniform friction coefficient was constant equal to 62.651 and did only contain the bottom friction. For the internal tides a linear friction coefficient was added as explained in chapter 6. In table 3 the RMS-values that have been calculated for this run can be found. What instantly stands out is that the RMS-values are larger than the values of table 2. The first question that arises is if there is to much or to less dissipation due to the internal tide. Therefore the results of the times in the table have been plotted and can be found in figures 10 till 17. From this pictures it is clear that in the pictures that have been generated from the model results the amplitude of the tides is higher than the amplitude in the SR95 data. As the amplitude is higher this means that there needs to be more dissipation. This means that the internal tide doesn't lead to enough dissipation yet, as the results from the first 2 runs are also still better than this run. So there are some options why this is the case. First of all it can be because of the h value, which hasn't been confirmed by other articles. Therefore one of the things that should be looked upon in future work is this value and how it could be calculated that it is the right value. Another thing that could lead to problems is the κ value. As this value has been said to be the best value, but it would probably be better to calculate this value from the bathymetry, instead of using 1 value for the entire earth. The last thing that could be the problem is the α value. As this is a proportionality constant that has all kind of values in different articles (see for instance in [6] or [11]). It is clear that future work should first focus on the internal tide, as this is clearly not good enough. Because of time reasons it could not be looked upon anymore in this project.

Time	Linear friction
01-Jan-1992 00:00:00	20.1529 cm
01-Jan-1992 06:00:00	20.6380 cm
01-Jan-1992 12:00:00	20.9347 cm
01-Jan-1992 18:00:00	20.1510 cm
02-Jan-1992 00:00:00	21.5604 cm
02-Jan-1992 06:00:00	21.5604 cm
02-Jan-1992 12:00:00	21.1303 cm
02-Jan-1992 18:00:00	20.3251 cm

Table 3: RMS-values of some times of the third model run.

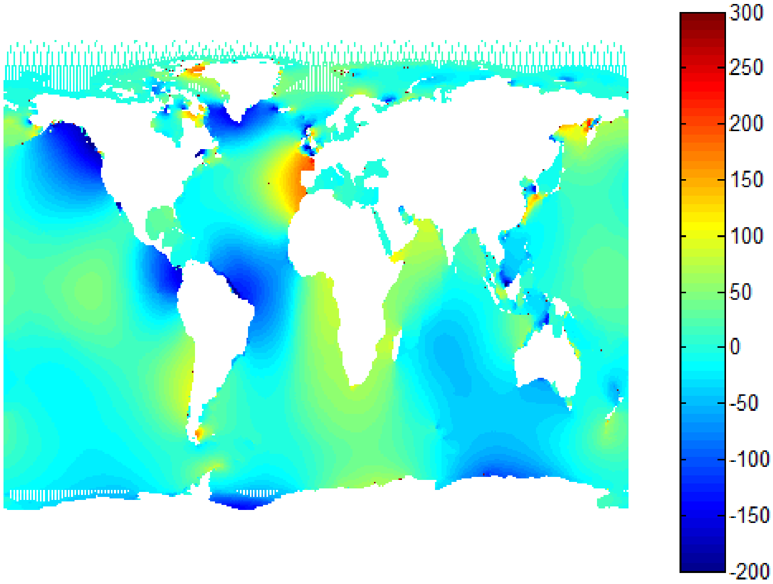


Figure 10: The model water level for 01-01-1992 0:00:00h.

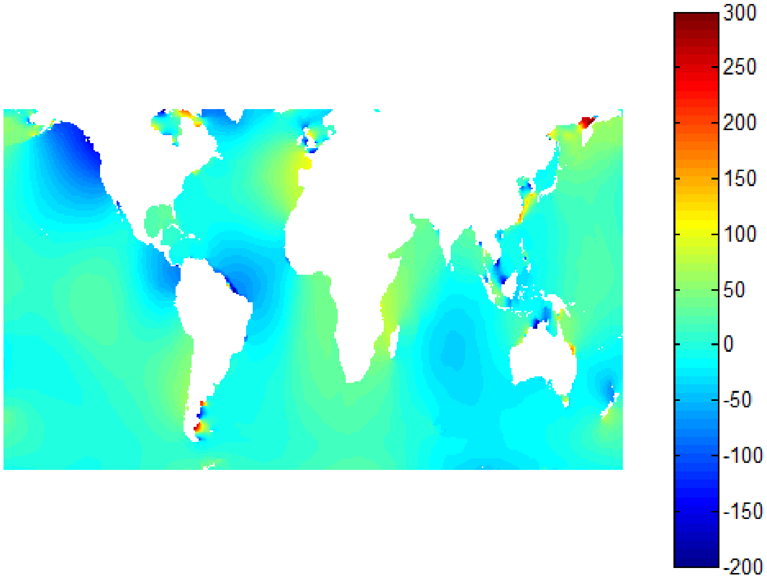


Figure 11: The data water level for 01-01-1992 0:00:00h.

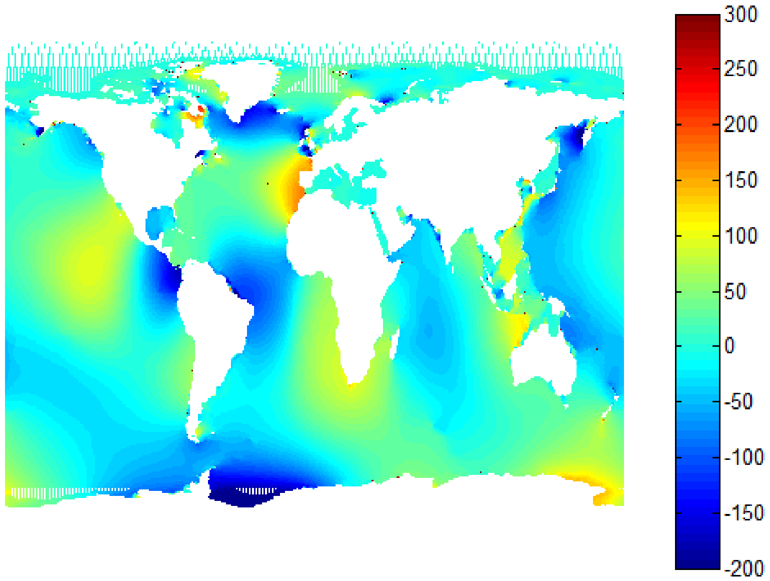


Figure 12: The model water level for 01-01-1992 12:00:00h.

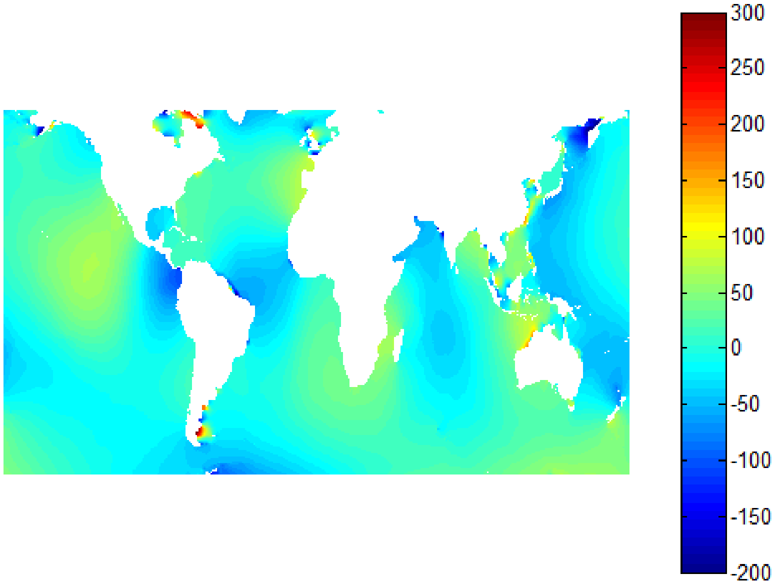


Figure 13: The data water level for 01-01-1992 12:00:00h.

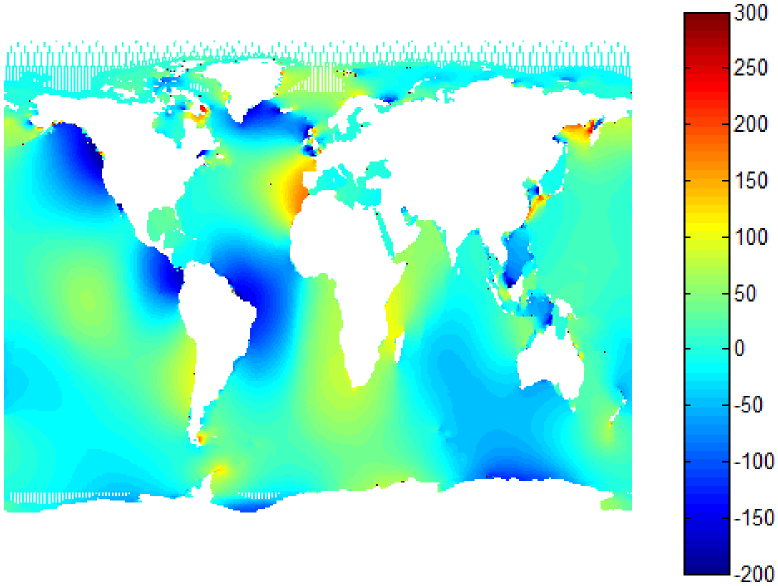


Figure 14: The model water level for 02-01-1992 0:00:00h.

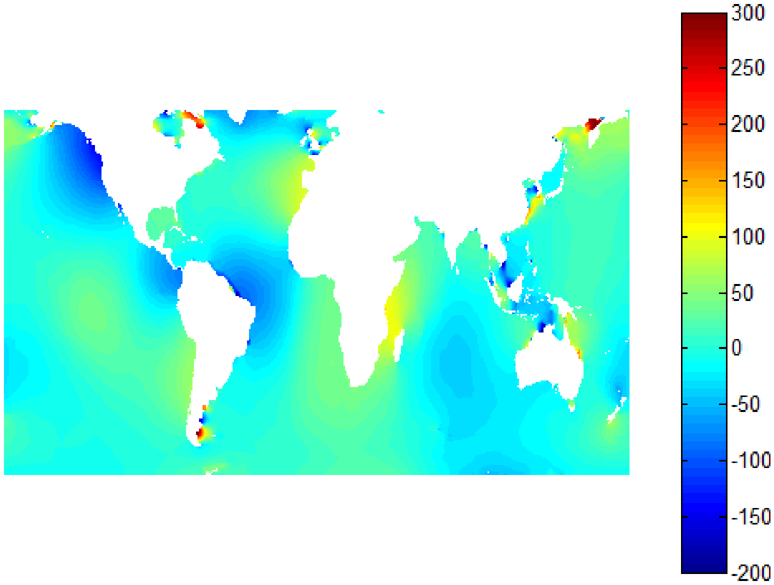


Figure 15: The data water level for 02-01-1992 0:00:00h.

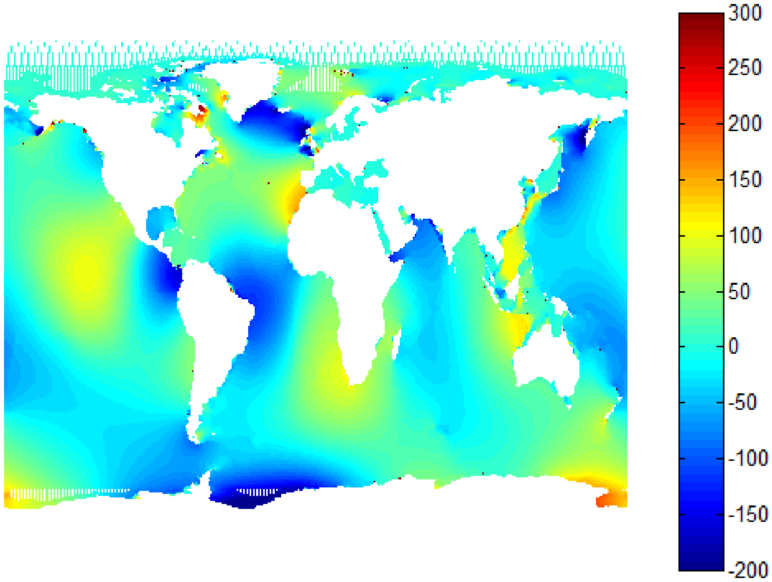


Figure 16: The model water level for 02-01-1992 12:00:00h.

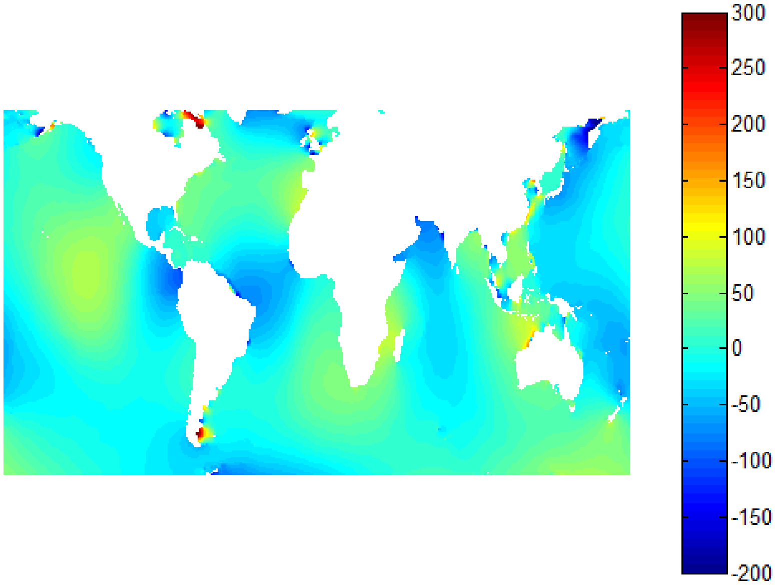


Figure 17: The data water level for 02-01-1992 12:00:00h.

8 Recommendations

From the results it is clear that this project is heading in the right direction, but also that it still needs future work before it gets where Deltares wants it to be. This future work should not focus on the grid, because, as I have pointed out earlier, in my opinion it won't get much better. Therefore the future work should focus on the model it self and first of all on the internal tides, as those results aren't very good at the moment. We know it can be better because of the results we got with the roughness out of [2].

Another thing that needs more work is the self-attraction and loading. As said before the approximation that has been used in this report is very crude. The first thing that can be looked upon is if the convolution integral can be calculated one time, and that out of that solution a beta can be found for every point of the grid that gives the same solution as the convolution. And if you really want to go all the way you should calculate the convolution integral every time step, but the question is if this would lead to a result that is so much better that it is worth the time it will probably take. If those things will be improved you will have a model that is good, and maybe already good enough for the prediction of boundary conditions.

But in order to get a storm-surge model or a sensitivity model it will take some further improvements. As, in order to get those models working, the model should also include wind forcing and that isn't included yet. Finally, if this has also been included someone could start a project to see if a 3D-model instead of a 2D-model is possible, and especially if such a model could still be run in a reasonable amount of time. As this model should run in a reasonable amount of time, as people probably don't want to wait to long for their boundary conditions.

9 Acknowledgments

First of all I would like to thank my supervisor at Deltares and also from my study Applied Mathematics, Martin Verlaan. He helped me a lot as my primary supervisor. Second I'd like to thank my supervisor Rob Mudde for taking place in my thesis committee from my study Applied Physics. I'd also like to thank Herman Kernkamp and Sander van der Pijl, two colleagues at Deltares, for their big help with Dflow-FM and the refinement. Finally, I'd like to thank Bart van den Dries and Kees Vuik for completing my thesis committee.

References

[1] Ross J. Murray *Explicit generation of orthogonal grids for ocean models* 1996: Journal of computational physics 126.

[2] Thea Vuik *Global tide model with Delft3D-FLOW* 2011: Master internship TU Delft.

[3] Lorenzo et al *Numerical and analytical estimates of M_2 tidal conversion at steep ocean ridges* 2005: Journal of physical oceanography, 36.

[4] Stefan G. Llewellyn Smith and W.R. Young *Conversion of the barotropic tide* 2001: Journal of physical oceanography, 32.

[5] Egbert et al *Numerical modeling of the global semidiurnal tide in the present day and in the last glacial maximum* 2004: Journal of geophysical research, 109.

[6] T.H. Bell, Jr *Topographically generated internal waves in the open ocean* 1975: Journal of geophysical research, 80.3.

[7] Steven R. Jayne and Louis C. St. Laurent *Parameterizing tidal dissipation over rough topography* 2001: Geophysical research letters 28.5.

[8] M. Hendershott *The effects of solid earth deformation on global ocean tides* 1972: Geophysical journal of the royal astronomical society 29.

[9] E.W. Schwiderski *Global ocean tides: I. A detailed hydrodynamical interpolation model* 1978: Naval surface weapons center Dahlgren.

[10] E.J.O. Schrama and R.D. Ray *A preliminary tidal analysis of TOPEX/-POSEIDON altimetry* 1994: Journal of geophysical research 99.12

[11] Lakshmi H. Kantho and Carol Anne Clayson *Numerical models of oceans and oceanic processes.*

[12] *User guide to the GEBCO one minute grid*

[13] *Technical reference manual DFlow-FM*

A Rotated grid generation

A.1 Create grid

```
clear all; close all; clc;
%This script creates a standard lat−lon grid and transforms is to a grid
%with 'North− and South Pole' on place of your choice.

%Create standard grid with resolution delta.
delta=1/2;
slon = −180:delta:180;
slat = −90:delta:90;
[slon,slat] = ndgrid(slon,slat);

%Move all points to their new location with the new 'North Pole' on Canada
%and the 'South Pole' still on the South Pole but shifted.
[nlon, nlat, map]=changepole(deg2rad(80),deg2rad(−85),deg2rad(−100), ...
    deg2rad(65),deg2rad(slon),deg2rad(slat));

deglon=rad2deg(nlon);
deglat=rad2deg(nlat);

%Remove all points that are on land from the grid.
[lon,lat]=removeland(deglon,deglat);

%Save the grid.
wlgrid('write','NPonNA',lon,lat,'spherical');

%Determine the maximum and minimum local enlargement.
mapfactor=[min(map) max(map)];
```

A.2 Rotation

```
function [nlon,nlat,mapfactor]=changepole(ZPlonnew,ZPlatnew,NPlonnew,NPlatnew,lon,lat)
%This function rotates the standard grid into the new grid with the shifted
%'North-' and 'South' Pole.

%Calculate parameters.
Chi_p=0.5*fc(NPlonnew-ZPlonnew,pi/2-ZPlatnew,pi/2-NPlatnew);
alpha=ft(NPlonnew-ZPlonnew,pi/2-ZPlatnew,pi/2-NPlatnew);
Qlon=NPlonnew-ft(alpha,Chi_p,pi/2-NPlatnew);
Qlat=fc(alpha,pi/2-NPlatnew,Chi_p);
Beta=ft(alpha,pi/2-NPlatnew,Chi_p);
r_p=tan(Chi_p/2);

%Transform the old points to the new points.

Chi_c=fc(lon,pi/2-lat,pi/2);
theta_c=ft(lon,pi/2-lat,pi/2);

Chi_s=2*atan(r_p*tan(0.5*Chi_c));
theta_s=theta_c+Beta;

nlon=Qlon+ft(theta_s,Chi_s,Qlat);
nlat=pi/2-fc(theta_s,Chi_s,Qlat);

%Make sure the final grid has longitude between -180 and 180 degrees.
nlon(nlon<(-pi))=nlon(nlon<(-pi))+2*pi;
nlon(nlon>pi)=nlon(nlon>pi)-2*pi;

mapfactor=(r_p*sec(Chi_c/2).^2)./(1+r_p^2*tan(Chi_c/2).^2);

end
```

A.3 Remove land

```
function [lon,lat] = removeland(lon,lat)
%This function removes the land from the grid.

%Load the depth file create with readgebco.
load diepten16

lon16=[-180:1/16:180];
lat16=[-90:1/16:90];
%Interpolate the depths to the grid.
d=interp2(lon16, lat16, diepten16, lon, lat);
%Remove all latitudes and longitudes of the grid with height bigger than 0.
lon(d>0)=NaN;
lat(d>0)=NaN;
end
```

A.4 Spherical cosine and tangent

```
function a = fc(A,b,c)
%This function determines the length of a side of a spherical triangle. It
%It is in reference to a triangle with vertices (A,B,C) and opposite sides
%(a,b,c).

a=acos(cos(A).*sin(b).*sin(c)+cos(b).*cos(c));

end
```

```
function B = ft(A,b,c)
%This function calculates the vertice of a triangle with the spherical tangent.
%It uses as reference a triangle with vertices (A,B,C) and opposite sides (a,b,c).

B=atan2(sin(A),sin(c)./tan(b)-cos(c).*cos(A));

end
```

B Bathymetry

B.1 GEBCO

```
%This script reads the dataset from GEBCO and interpolates it to a grid
%with a resolution of 1/16 degree.
clear all; close all; clc;
tic
i=1;
%Read GEBCO part by part so computer doesn't give a memory error
while i<234
    start=(i-1)*10^6+1;
    count=10^6;
    diepten(start:i*count)=ncread('gridone.nc','z',start,count);
    i=i+1;
end
if i==234
    start=(i-1)*10^6+1;
    diepten(start:233312401)=ncread('gridone.nc','z',start,Inf);
end
toc

E=reshape(diepten,21601,10801)';
toc

X=0:1/60:360;
Y=-90:1/60:90;
toc

Xn=0:1/16:360;
Yn=-90:1/16:90;
%Interpolate the GEBCO data into a 1/16 degree grid
d=interp2(X,Y,E,Xn,Yn);
toc
```

B.2 Bathymetry grid

```
%This script creates a bathymetry file that can be used in Dflow—fm in the
%refinement method.

load diepten16

lon16=—180:1/16:180;
lat16=—90:1/16:90;
%The new longitude will be from —181 till 181 so the refinement is also
%good on the sides.
lon=—181:1/16:181;
lat=—90:1/16:90;

[lon,lat]=ndgrid(lon,lat);

%Interpolate the data to a vector version of a 1/16 degree grid.
d=interp2(lon16,lat16,d,reshape(lon,4174577,1),reshape(lat,4174577,1));

%Create a 3—column array with respectively the lon, lat and depth values.
A=[reshape(lon,4174577,1) flipud(reshape(lat,4174577,1)) d];

for i=0:1440
    %Find all coordinates with —181<lon<—180 and make their depth values equal to
    %the values of 179<lon<180.
    b1=i*2897+1;
    b2=(i+1)*2897—15;
    e1=i*2897+8;
    e2=(i+1)*2897—8;
        A(b1:e1,3)=flipud(A(b2:e2,3));

    %Find all coordinates with 180<lon<181 and make their depth values equal to
    %the values of —180<lon<—179.
    b3=(i+1)*2897—7;
    b4=i*2897+9;
    e3=(i+1)*2897;
    e4=i*2897+16;
        A(b3:e3,3)=flipud(A(b4:e4,3));
end
%Save the values as a sample file so it can be used in Dflow—FM
save diepten.xyz —ascii A
```

B.3 Get depth into Net-file

```
%This script reads the used network and adds the depth values to it as
%z−values.

%Load GEBCO data
load diepten16

lon16= −180:1/16:180;
lat16= −90:1/16:90;
%Load grid lat and lon values.
lon=ncread('npopna25_net.nc','NetNode_x');
lat=ncread('npopna25_net.nc','NetNode_y');
%Interpolate depths to the grid.
d=interp2(lon16,lat16,diepten16,lon,lat);
%Add the depths as z−values.
ncwrite('npopna25_net.nc','NetNode_z',d);
```

C Internal Tides

C.1 Linear friction coefficients

```
%This script calculates the c_D values for the internal tide.
clear all; clc; close all;
%Load the Bouyancy frequency with it's corresponding lon and lot
%coordinates and the depth file out of GEBCO.
load N
load latN
load lonN
load diepten16

latN=lat;
lonN=lon;

lon=-180:1/16:180;
lat=-90:1/16:90;
[lon,lat]=ndgrid(lon,lat);
lon=lon';
lat=flipud(lat');
%Load the lon and lat coordinates of the grid being used.
long=ncread('refinedlin_map.nc','FlowLink_xu');
latg=ncread('refinedlin_map.nc','FlowLink_yu');
%Give alpha and kappa it's value.
alpha=1.1;
kappa=2*pi/10000;

s=size(latg);
S=s(1)*s(2);
factor=zeros(s);
%Determine the tidal frequency factor.
O=7.292115*10^-5;
f=2*O*sin(deg2rad(latg));
o1=1.405257*10^-4;
o2=7.026285*10^-5;

for i=1:s(1)
    for j=1:s(2)
        if latg(i,j)>74.5
            factor(i,j)=sqrt(o1^2-f(i,j)^2)/(o1);
        elseif latg(i,j)<-74.5
            factor(i,j)=sqrt(o1^2-f(i,j)^2)/(o1);
        elseif latg(i,j)>30
            factor(i,j)=sqrt(o2^2-f(i,j)^2)/(o2);
        elseif latg(i,j)<-30
            factor(i,j)=sqrt(o2^2-f(i,j)^2)/(o2);
        else
            factor(i,j)=1;
        end
    end
end
%Interpolate the bouyancy frequency to the grid coordinates.
N=griddata(lonN,latN,N,long,latg);
N(isnan(N))=mean(mean(N(~isnan(N))));
```

```
%Calculate h^2 on 1/16 degree grid.
h=geth(diepten16);
%Interpolate those h^2 to the grid coordinates.
h2=interp2(lon,lat,h,long,latg);
%Calculate c_D and save it to a sample file.
cd=1/2*abs(factor)*alpha*kappa.*h2.*N;

cd=[long latg cd];

save cdrefined2.xyz -ascii cd
```

C.2 h^2 calculation

```
function h2 = geth(diepten)
%This function determines the value of h^2.

%The radius of the earth.
R=6371000;

%The value needs to be determined on the cellcenters so one column is
%copied from one side to the other side.
a=size(diepten);
diepten=[0 diepten(1,:) 0; diepten(:,a(2)) diepten diepten(:,1) ; 0 diepten(a(1),:) 0];

a=size(diepten);
%Determine the depth differences.
Dup=diepten(2:a(1),:)-diepten(1:a(1)-1,:);
Ddown=diepten(1:a(1)-1,:)-diepten(2:a(1),:);
Dleft=diepten(:,2:a(2))-diepten(:,1:a(2)-1);
Dright=diepten(:,1:a(2)-1)-diepten(:,2:a(2));
%Determine the distance between two points.
dx=1/16*2*pi/360*R;
%Determine h^2.
h2=((sqrt(Dup(1:a(1)-2,2:a(2)-1).^2/dx^2+Ddown(2:a(1)-1,2:a(2)-1).^2/dx^2)*dx...
    +sqrt(Dleft(2:a(1)-1,1:a(2)-2).^2/dx^2+Dright(2:a(1)-1,2:a(2)-1).^2/dx^2)*dx).^2)/4;
end
```

D After processing

D.1 RMS-value

```
function RMS = readwaterlevel(model,start)
%This function calculates the RMS-value for every half an hour of the first
%and second of January 1992. Here model is the his-file and start is the
%timecode that corresponds with 01-01-1992 0:00:00, that you can find in
%quickplot.

%Load the SR95 data and the model results.
globaltide=load('global_tide.txt','-ascii');

X=ncread(model,'station_x_coordinate');
Y=ncread(model,'station_y_coordinate');
waterlevel=ncread(model,'s1',[1 start],[inf 96]);

lon=-180:1/2:180;
lat=-90:1/2:90;

[lat,lon]=ndgrid(lat,lon);

RMS=zeros(96,1);
tide=zeros(361,721);

%Calculate the RMS-value for every half an hour.
for i=1:1:96

results=[X Y 100*waterlevel(:,i)];

tide2=globaltide(((i-1)*361+(1:361)),:);
tide(:,1:361)=tide2(:,361:721);
tide(:,362:721)=tide2(:,1:360);
tide(tide==1000)=NaN;

data=interp2(lon,lat,tide,X,Y);

diff=abs(results(:,3)-data);
diff(isnan(diff))=[];
RMS(i)=rms(diff);

end

end
```

D.2 Plot

```
function plotwaterlevels(model,start)
%This function plots the waterlevels of the SR95 database, of the model
%results and of the difference between those tho datasets and saves them as
%an .avi file. Here the model is the map-file and start the timecode
%corresponding with 01-01-1992 0:00:00.

%Load the SR95 data and the model results.
globaltide=load('global_tide.txt','-ascii');

X=ncread(model,'FlowElem_xcc');
Y=ncread(model,'FlowElem_ycc');
waterlevel=100*ncread(model,'s1',[1 start],[inf 96]);

lon=-180:1/2:180;
lat=-90:1/2:90;

[lat,lon]=ndgrid(lat,lon);

lon2=reshape(lon,1,260281);
lat2=reshape(lat,1,260281);

tide=zeros(361,721);

%Create .avi files.
video1=VideoWriter('waterlevelsp.avi');
video2=VideoWriter('sr95.avi');
video3=VideoWriter('diffsp.avi');
open(video1);
open(video2);
open(video3);

for k=1:96
%First calculate the difference between the two datasets and then create
%an image for every half an hour between 01-01-1992 0:00h and 03-01-1992
%0:00h for the three sets and save them to the .avi files.
tide2=globaltide(((k-1)*361+(1:361)),:);
tide(:,1:361)=tide2(:,361:721);
tide(:,362:721)=tide2(:,1:360);
tide(tide==1000)=NaN;

data=interp2(lon,lat,tide,X,Y);
diff=abs(waterlevel(:,k)-data);

tide3=reshape(tide,1,260281);

figure(1)
    h=gcf;
    scatter(X,Y,2,waterlevel(:,k),'filled');
    caxis([-200 300])
    axis([-190 190 -100 100])
    colorbar
    M1(k)=getframe(h);
    writeVideo(video1, M1(k));

```

```
figure(2)
    h=gcf;
    scatter(lon2,lat2,2,tide3,'filled');
    caxis([−200 300])
    axis([−190 190 −100 100])
    colorbar
    M2(k)=getframe(h);
    writeVideo(video2, M2(k));

figure(3)
    h=gcf;
    scatter(X,Y,2,diff);
    caxis([0 50])
    axis([−190 190 −100 100])
    colorbar
    M3(k)=getframe(h);
    writeVideo(video3, M3(k));
end

close(video1);
close(video2);
close(video3);

%Let the user know the function is done.
'plotwaterlevels is done'
end
```

E Results

Time	Rotated grid	Refined grid	Linear friction
01-Jan-1992 00:00:00	15.4616312132234	12.1744151058704	20.1529826040167
01-Jan-1992 00:30:00	15.3714862398911	12.0977562319021	19.6696511149158
01-Jan-1992 01:00:00	15.3068232238594	11.9743887345703	19.1152357627575
01-Jan-1992 01:30:00	15.3171659947904	11.8310883349660	18.6094084888837
01-Jan-1992 02:00:00	15.4294831520324	11.7019151318849	18.2723894303834
01-Jan-1992 02:30:00	15.6389393542572	11.6282415802286	18.1959491521938
01-Jan-1992 03:00:00	15.9509795294211	11.6382478687415	18.4041758858473
01-Jan-1992 03:30:00	16.2493674750377	11.7416996929036	18.8527192281070
01-Jan-1992 04:00:00	16.5136995594512	11.9233568435227	19.4346737891774
01-Jan-1992 04:30:00	16.7103659493449	12.1486444311609	20.0115101137571
01-Jan-1992 05:00:00	16.7411495839834	12.3589854671487	20.4579812716014
01-Jan-1992 05:30:00	16.6206510236272	12.4938442154602	20.6808394752694
01-Jan-1992 06:00:00	16.3238763484638	12.5019106454815	20.6380658427757
01-Jan-1992 06:30:00	15.8790670475163	12.3604892582184	20.3350158666189
01-Jan-1992 07:00:00	15.3668507160420	12.0775092620810	19.8337990344215
01-Jan-1992 07:30:00	14.8934222443413	11.6918656224832	19.2372446384428
01-Jan-1992 08:00:00	14.5174096429864	11.2671848025583	18.6808978938171
01-Jan-1992 08:30:00	14.3157938698240	10.8836975425043	18.3019527364330
01-Jan-1992 09:00:00	14.3736385905192	10.6347222985159	18.2067764351097
01-Jan-1992 09:30:00	14.6258355668347	10.5942272508525	18.4242705064849
01-Jan-1992 10:00:00	15.0202079203546	10.7850595615825	18.9014706895592
01-Jan-1992 10:30:00	15.5452859457408	11.1687509924600	19.5221390319915
01-Jan-1992 11:00:00	16.0576831783298	11.6574644636311	20.1486326917323
01-Jan-1992 11:30:00	16.4948659737122	12.1472037930405	20.6506908185262
01-Jan-1992 12:00:00	16.7981396468090	12.5475971775195	20.9347536710632
01-Jan-1992 12:30:00	16.8986383561369	12.8055378359885	20.9567755453519
01-Jan-1992 13:00:00	16.7644051121430	12.8941082827315	20.7197392069052
01-Jan-1992 13:30:00	16.4413065410670	12.8178128325835	20.2719090523198
01-Jan-1992 14:00:00	15.9850661214251	12.6023497077968	19.7067742033692
01-Jan-1992 14:30:00	15.4702436697191	12.2953328516530	19.1461828156305
01-Jan-1992 15:00:00	14.9873888962934	11.9641846541666	18.7197786741153
01-Jan-1992 15:30:00	14.6232937165803	11.6724560750258	18.5330366947214
01-Jan-1992 16:00:00	14.3377811517894	11.4737294797555	18.6223932179528
01-Jan-1992 16:30:00	14.2123913516331	11.3931020486730	18.9427696927933
01-Jan-1992 17:00:00	14.2707102443858	11.4318162521494	19.3873113719524
01-Jan-1992 17:30:00	14.3325264592602	11.5627946134746	19.8274728610061
01-Jan-1992 18:00:00	14.4946432497340	11.7414610218718	20.1510970950416
01-Jan-1992 18:30:00	14.7043945974302	11.9219853206610	20.2868564137716
01-Jan-1992 19:00:00	14.9233149953122	12.0720045317438	20.2169658777470
01-Jan-1992 19:30:00	15.1707432789240	12.1832311625210	19.9745197209682
01-Jan-1992 20:00:00	15.4601258053777	12.2646691880087	19.6435141959008
01-Jan-1992 20:30:00	15.7050970710682	12.3406779225078	19.3415407259776
01-Jan-1992 21:00:00	15.9580414034711	12.4266464815599	19.1866818544521
01-Jan-1992 21:30:00	16.1691468291853	12.5383891774282	19.2621492669873
01-Jan-1992 22:00:00	16.3116051751232	12.6731090285972	19.5849643954860
01-Jan-1992 22:30:00	16.3514254646927	12.8155835996913	20.0957153424888
01-Jan-1992 23:00:00	16.3475692842428	12.9373414080888	20.6798251238035
01-Jan-1992 23:30:00	16.2924905905940	13.0046036954830	21.2072900235590

Time	Rotated grid	Refined grid	Linear friction
02-Jan-1992 00:00:00	16.1780979990712	12.9954809840170	21.5604913951193
02-Jan-1992 00:30:00	16.0409029819303	12.8974868905687	21.6603225060522
02-Jan-1992 01:00:00	15.9038500992477	12.7351342725138	21.4769475537261
02-Jan-1992 01:30:00	15.7953799765765	12.5409673183470	21.0365102954177
02-Jan-1992 02:00:00	15.7249482632239	12.3607661784528	20.4191915673500
02-Jan-1992 02:30:00	15.7580607036076	12.2350445420955	19.7576709864247
02-Jan-1992 03:00:00	15.8863448196127	12.1937637425305	19.2092095406240
02-Jan-1992 03:30:00	16.1138287659168	12.2478929960172	18.9327600578766
02-Jan-1992 04:00:00	16.4437546071908	12.3882353309220	19.0201703987376
02-Jan-1992 04:30:00	16.7367668559636	12.5773679161899	19.4631816478618
02-Jan-1992 05:00:00	16.9700989671129	12.7729882701892	20.1498854508635
02-Jan-1992 05:30:00	17.1497545956747	12.9282728884834	20.9085859577879
02-Jan-1992 06:00:00	17.1368053145172	12.9984578338487	21.5604432104577
02-Jan-1992 06:30:00	16.9500345499593	12.9473851339886	21.9669986387945
02-Jan-1992 07:00:00	16.5919969125783	12.7604550798939	22.0432788282098
02-Jan-1992 07:30:00	16.0724893775853	12.4514480269684	21.7745405415504
02-Jan-1992 08:00:00	15.5121475613758	12.0672950270302	21.2108591572536
02-Jan-1992 08:30:00	15.0021935507372	11.6780497322815	20.4669994517076
02-Jan-1992 09:00:00	14.6340343716450	11.3619838921425	19.7102997341905
02-Jan-1992 09:30:00	14.4861776854567	11.1899574564558	19.1313322446877
02-Jan-1992 10:00:00	14.6229196720429	11.2078182494123	18.8904636954347
02-Jan-1992 10:30:00	14.9823061584862	11.4191414722765	19.0608295099830
02-Jan-1992 11:00:00	15.4972414746780	11.7844199708231	19.5958220146066
02-Jan-1992 11:30:00	16.0891329609644	12.2306147430987	20.3492871490711
02-Jan-1992 12:00:00	16.6466223721463	12.6641073887065	21.1303819288739
02-Jan-1992 12:30:00	17.1085406511534	13.0110333400659	21.7730330929422
02-Jan-1992 13:00:00	17.4076115677841	13.2219428595729	22.1405079118280
02-Jan-1992 13:30:00	17.4602242148524	13.2894338624491	22.1725177339203
02-Jan-1992 14:00:00	17.2747515648119	13.2280189831264	21.8657014629929
02-Jan-1992 14:30:00	16.8778341755279	13.0737462037822	21.2809986561773
02-Jan-1992 15:00:00	16.3487381337068	12.8724209723613	20.5435644371068
02-Jan-1992 15:30:00	15.7582847656093	12.6766468154830	19.8149425986133
02-Jan-1992 16:00:00	15.1897086682506	12.5375300130425	19.2783259199681
02-Jan-1992 16:30:00	14.7627057601363	12.4777083598097	19.0714428834293
02-Jan-1992 17:00:00	14.4569804315148	12.5006910252562	19.2378253817822
02-Jan-1992 17:30:00	14.3570634661912	12.5839764196690	19.7072973720880
02-Jan-1992 18:00:00	14.4275428019217	12.7026265844054	20.3251788848173
02-Jan-1992 18:30:00	14.5913873740254	12.8250728211183	20.9183850643961
02-Jan-1992 19:00:00	14.8758554039554	12.9278928644572	21.3421110345299
02-Jan-1992 19:30:00	15.1926269733189	12.9991437000481	21.5113169979002
02-Jan-1992 20:00:00	15.5075011718216	13.0518409743729	21.4077648601018
02-Jan-1992 20:30:00	15.8580509968407	13.1086348931550	21.0843341890989
02-Jan-1992 21:00:00	16.2023379427002	13.2039561163008	20.6532576386280
02-Jan-1992 21:30:00	16.5140066590581	13.3580621094555	20.2722065619217
02-Jan-1992 22:00:00	16.7876068597748	13.5726653287622	20.0948255970700
02-Jan-1992 22:30:00	17.0019928455423	13.8233571206552	20.2246500307700
02-Jan-1992 23:00:00	17.1123274518410	14.0695787639955	20.6692197094891
02-Jan-1992 23:30:00	17.1204793733661	14.2671153840626	21.3380142952613

Table 4: The RMS-value for every half an hour on the first and second of January 1992.