The not so short introduction to the subgrid-model 1.1.

Nicolette Volp

October 27, 2009

Contents

Chapter 1

Introduction

In this manual a numerical model is introduced, that is able to determine velocities on a fine scale using a method, which has a considerable better efficiency and/ or accuracy than former models. By defining the subgrid on top of a normal grid, the accuracy is strongly improved, without a considerable increase of the computation time. Detailed information about the bottom topography, bottom roughness etc, can be used at subgrid level. Using this method the key parameters, in for example the friction term, can be determined more accurately. In this case the velocity can be calculated more precise, using the depth data at subgrid scale. The details of this method are of course discusses later on.

In the next chapter the discretization of the hydrodynamic equations and the principle of the use and the advantage of the subgrid method is explained. In chapter 3 the code of the model is discussed, including some considerations of choices, which still have to be made. Some tests are performed in chapter 4 to compare the results with an analytical solution for accuracy and to a model using a fine grid for efficiency. The last chapter gives an overview of the main components of the model.

Chapter 2

Equations and model derivation.

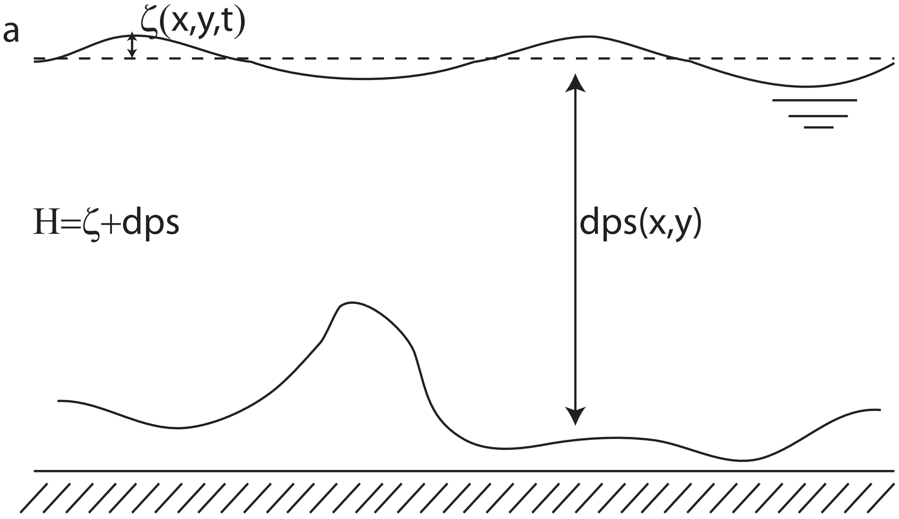

Figure 2.1: Schematic view of the basic principles of the model as it currently embedded in the model.

The derived model is able to describe depth-averaged flows in a tidal estuary, including a river outflow. The depth averaged shallow water equations are given by;

$$\frac{\partial \zeta}{\partial t} + \frac{(\partial H u)}{\partial x} + \frac{(\partial H v)}{\partial y} = 0, \tag{2.1}$$

$$\frac{\partial u}{\partial t} + g\frac{\partial \zeta}{\partial x} + C_{f_x} \parallel u \parallel u = 0, \tag{2.2}$$

$$\frac{\partial v}{\partial t} + g\frac{\partial \zeta}{\partial y} + C_{f_y} \parallel v \parallel v \quad = \quad 0, \tag{2.3}$$

in which ζ is the surface elevation and H the height of the watercolumn, $H(x, y, t) = \zeta(x, y, t) + dps(x, y)$. The depth averaged velocities in the x- and y-direction are given by u and v, respectively. The gravity acceleration is appointed with g and C_{f_q}, for $q = x$,y, is a frictional constant. The friction term is the first term adjusted to the subgrid method and will illustrate the possibilities of a subgrid schematization. This will be discussed in detail in section 2.2. The advective and diffusion terms are not yet included in the model. This will be one of the next steps in the development of the subgridmethod. A dry fall condition or routine is not yet included in the model, but as long as the model is defined upstream and if the following condition holds, the waterlevel will stay positive [**?**]:

$$\frac{\Delta t}{\Delta x} u_m^n \leq 1. \tag{2.4}$$

In words; as long as the ratio of the timestep times the velocity at a gridpoint and the spatial step is equal or smaller than one, non-negative waterlevels are ensured. Here, the indices just indicate that this condition should hold for each gridpoint m at time n. In paragraph 2.1 this set of equations is discretized and in paragraph 2.2 details of the friction term based on a subgrid schematization are described.

2.1 Discretization of the model equations

Figure 2.2 shows a staggered grid, the cells are numbered in the x-direction by m and in the y-direction by n. The surface elevation ($\zeta_{n,m}$) is assumed to be constant throughout a cell. The velocities $u_{m,n}$ and $v_{m,n}$ are defined at the transitions of the cells, the red (blue) stars in figure 2.2. These are called $u(v)$-points. To obtain the $v_{m,n}(u_{m,n})$-velocity at a $u(v)$-point, the four $v(u)$-velocities of the surrounding velocity points are averaged. Usually, the depth of a grid cell is defined at the center of the cell, together with the elevation point. In case of the subgrid-method this is no longer the trivial thing to do, because the depth is known at subgrid level. For the derivation of the system it is useful to assume the depths at the centers of the cells. In the next chapter and in chapter 3 the considerations of defining the depth will be discussed more thoroughly. Currently, the system consists of a one dimensional flow, so only two boundaries need to be considered. At the inflow boundary, $m = mmax + 1$, there is a river outflow into the estuary. At the other side, at $m = 0$, the system flows out into a tidal sea.

To discretize the set of equations in space, a first order upwind rule is performed. An Alternating Direction Implicit (ADI)-method is used to solve this system of equations. The basic principle of an ADI-method is that in/ during one timestep the velocities are determined twice. The timestep is split up in two. The first step is used to determine

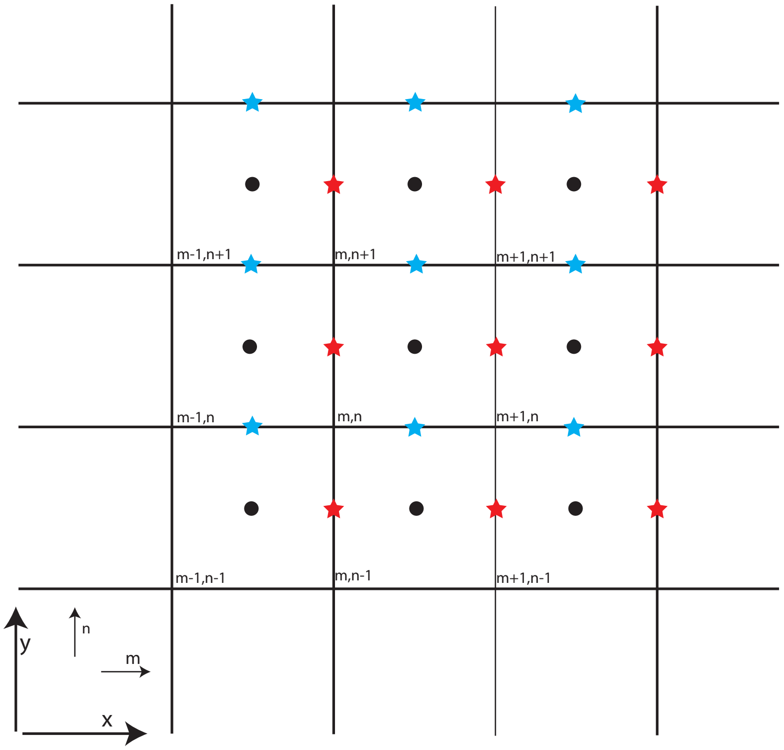

Figure 2.2: A staggered grid appointing u and v velocity points with the red and blue stars respectively and the black dots appoint where the surface elevation is defined.

u implicit at time $t = n + \theta$ and v is calculated directly at time $t = n + 1 - \theta$. This is switched, during the second step. Then, v is determined implicit and u explicit. The value of θ indicates the 'implicitness' of the solver algorithm. Depending on the kind of flow, a specific value between 0 and 1 for θ is chosen. If θ is 1, the system is fully implicit in one direction and if θ is 0 the system is explicit.

Discretization of the governing equation

The discrete equations for the first step are

$$\frac{\zeta_{m,n}^{n+\frac{1}{2}} - \zeta_{m,n}^{n}}{\frac{1}{2}\Delta t} + \frac{H_{m,n}^{n} u_{m,n}^{n+\theta} - H_{m-1,n}^{n} u_{m-1,n}^{n+\theta}}{\Delta x} + \frac{H_{m,n}^{n} v_{m,n}^{n} - H_{m,n-1}^{n} v_{m,n-1}^{n}}{\Delta y} = 0, \quad (2.5)$$

$$\frac{u_{m,n}^{n+\theta} - u_{m,n}^{n}}{\theta \Delta t} + g \frac{\zeta_{m,n}^{n+\frac{1}{2}} - \zeta_{m-1,n}^{n+\frac{1}{2}}}{\Delta x} + C_{f_x} \parallel u_{m,n}^{n+\theta} \parallel u_{m,n}^{n+\theta} = 0, \quad (2.6)$$

$$\frac{v^{n+1-\theta}_{m,n} - v^{n}_{m,n}}{(1-\theta)\Delta t} + g\frac{\zeta^{n}_{m,n} - \zeta^{n}_{m,n-1}}{\Delta y} + C_{f_y} \parallel v^{n+1-\theta}_{m,n} \parallel v^{n+1-\theta}_{m,n} \quad = \quad 0, \tag{2.7}$$

in which the symbols are the discrete variants of those in equations 2.1-2.3. From now on, a part of the friction term is replaced by a coefficient, because that is where the subgrid becomes important and is discussed in the next section.

$$\begin{array}{rcl} C_{f_x} \parallel u^{n+\theta}_{m,n} \parallel & \rightarrow & \lambda_x \\ C_{f_y} \parallel v^{n+1-\theta}_{m,n} \parallel & \rightarrow & \lambda_y \end{array}$$

From equation 2.7, it is clear that v can be determined explicitly.

To calculate $u^{n+\theta}_{m,n}$ and $\zeta^{n+\frac{1}{2}}_{m,n}$, $u^{n+\theta}_{m,n}$ is substituted in equation 2.5, which becomes

$$-a_{m,n}\zeta^{n+\frac{1}{2}}_{m-1,n} + b_{m,n}\zeta^{n+\frac{1}{2}}_{m,n} - c_{m,n}\zeta^{n+\frac{1}{2}}_{m+1,n} = d_{m,n}, \tag{2.8}$$

with

$$a_{m,n} = \frac{(\Delta t)^2 \frac{1}{2} g\theta}{(\Delta x)^2(1-\theta\Delta t\lambda_x)} H^{n}_{m-1,n}, \tag{2.9}$$

$$c_{m,n} = \frac{(\Delta t)^2 \frac{1}{2} g\theta}{(\Delta x)^2(1-\theta\Delta t\lambda_x)} H^{n}_{m,n}, \tag{2.10}$$

$$\begin{aligned} b_{m,n} &= \frac{(\Delta t)^2 g\theta}{(\Delta x)^2(1-\theta\Delta t\lambda_x)} \left(H^{n}_{m-1,n} + H^{n}_{m,n}\right) \\ &= 1 + a_{m,n} + c_{m,n}, \end{aligned} \tag{2.11}$$

$$\begin{aligned} d_{m,n} &= \zeta^{n}_{m,n} - \frac{\frac{1}{2}\Delta t}{\Delta x(1-\theta\Delta t\lambda_x)} \left[H^{n}_{m-1,n}u^{n}_{m-1,n} - H^{n}_{m,n}u^{n}_{m,n}\right] \\ &+ \frac{\frac{1}{2}\Delta t}{\Delta y} \left[H^{n}_{m,n-1}v^{n}_{m,n-1} - H_{m,n}v^{n}_{m,n}\right], \end{aligned} \tag{2.12}$$

When written in matrix notation

$$A\vec{\zeta} = \vec{D}, \tag{2.13}$$

the matrix A, consisting of coefficients $a_{m,n}$, $b_{m,n}$ and $c_{m,n}$, has a diagonal form, it has only non-zero entries at the diagonal and the two diagonals next to it. This can clearly be seen in equation 2.8. Boundary conditions are needed to close the set of equation at $m = 0$ and at $m = mmax + 1$, which are discussed below. To solve this system of equations, the matrix has to be converted into an upper triangular form (Gauss elimination), such that

$$\zeta^{n}_{m,n} - e_{m,n}\zeta^{n+\frac{1}{2}}_{m+1,n} = f_{m,n}. \tag{2.14}$$

Rewriting the equations in the upper triangular form, the coefficients $e_{m,n}$ and $f_{m,n}$ can be written in terms of $a_{m,n}$, $b_{m,n}$, $c_{m,n}$ and $d_{m,n}$;

$$e_{m,n} = \frac{c_{m,n}}{b_{m,n} - a_{m,n}e_{m-1,n}} \tag{2.15}$$

$$f_{m,n} = \frac{d_{m,n} + a_{m,n}f_{m-1,m}}{b_{m,n} - a_{m,n}f_{m-1,n}} \tag{2.16}$$

Now ζ and u can be derived, using the boundary conditions and by iteration of 2.16, 2.16 and equation 2.14.
It might be the case, that Guus uses different constant in the code..

Having the new values for ζ at $t = n + \frac{1}{2}$ and for u and v at $t = n + \theta$ and $t = n + 1 - \theta$ respectively, the procedure described above can be repeated to get all the variable values at $t = n + 1$. In this second phase, the discretized equations are;

$$\begin{aligned}
\frac{\zeta^{n+1}_{m,n} - \zeta^{n+\frac{1}{2}}_{m,n}}{\frac{1}{2}\Delta t} + H^n_{n,m}\frac{u^{n+\theta}_{m+1,n} - u^{n+\theta}_{m,n}}{\Delta x} + H^n_{n,m}\frac{v^{n+1}_{m,n+1} - v^{n+1}_{m,n}}{\Delta y} &= 0, && (2.17)\\
\frac{u^{n+1}_{m,n} - u^{n+\theta}_{m,n}}{(1-\theta)\Delta t} + g\frac{\zeta^{n+\frac{1}{2}}_{m,n} - \zeta^{n+\frac{1}{2}}_{m-1,n}}{\Delta x} + C_{f_x} \parallel u^{n+1}_{m,n} \parallel u^{n+1}_{m,n} &= 0, && (2.18)\\
\frac{v^{n+1}_{m,n} - v^{n+1-\theta}_{m,n}}{\theta\Delta t} + g\frac{\zeta^{n+1}_{m,n} - \zeta^{n+1}_{m,n-1}}{\Delta y} + C_{f_y} \parallel v^{n+1}_{m,n} \parallel v^{n+1}_{m,n} &= 0, && (2.19)
\end{aligned}$$

Boundary conditions

It is already mentioned that at one side, at $m = 0$, the system is bounded by a tidal sea. The elevation at this boundary is naturally given by the tidal elevation. This is (now) represented by a sinusoidal signal consisting of a single frequency. At the opposite site, a river flows into the system.

At this point, the boundary condition is not yet completely clear to me. If the river is assumed to be stationary, uniform and the tangent velocity v is also assumed zero (which is a consequence of neglecting diffusion and advection), implying;

$$\begin{aligned}
v &= 0,\\
\frac{\partial u}{\partial t} &= 0,\\
\frac{\partial u}{\partial x} &= 0, \text{implies} \qquad \frac{\partial H}{\partial x} = 0.
\end{aligned}$$

If I'm correct, this last equation gives a correct, non reflecting?, boundary condition. It sets the elevation of the water level equal to the gradient of the bottom, a river in equilibrium.

This is actually the case, considering the code. The discrete equation can be written as;

$$\frac{\zeta^{n+\frac{1}{2}}_{mmax+1,n} - \zeta^{n}_{mmax+1,n}}{\frac{1}{2}\Delta t} - \sqrt{gH^{n}_{mmax+1,n}}\frac{\zeta^{n}_{mmax+1,n} - \zeta^{n}_{mmax,n} + \Delta x * i_b}{\Delta x} = 0, \qquad (2.20)$$

in which i_b is the slope of the depth (this is anti-proportional to the bottom slope). The elevation at grid cell $m = mmax + 1$ is constant in time, consequently, the first term of the equation is zero. In that case the factor $\sqrt{gH}$ has no influence and the term left over implies indeed a balance of the surface gradient and the bottom topography. But the thing that is not clear, is why the other terms are prescribed in such a way. Of course the discharge of the river can vary, as well as the water level. Is it correct to assume that, when neglecting frictional effects, perturbations in river outflow travel with $\sqrt{gH}$? Is this why is chosen for this formulation? Another comment is added by Bram, why is the boundary condition at the river side of the system not expressed in terms of the discharge? Assuming a stationary state of the river, the continuity and the momentum equation reduce in 1 dimension to;

$$\frac{\partial \zeta}{\partial t} + u\frac{\partial H}{\partial x} = 0, \qquad (2.21)$$

$$g\frac{\partial \zeta}{\partial x} + C_f \parallel u \parallel u = 0 \qquad \rightarrow \qquad u = \pm\sqrt{gH\frac{\partial \zeta}{\partial x}}. \qquad (2.22)$$

The velocity can be substituted in the continuity equation. If a discharge is subscribed, this would close the boundary condition, right?

2.2 Subgrid schematization of the friction term.

In this section the subgrid schematization of Guus will be displayed. (The basis of this section is now in the word-file.) Eventually, the other subgrid terms will be elucidated here also.

Chapter 3

The Code

In this chapter the model code test_flowmodel is discussed in detail to be able to appoint the critical choices in the model, which might need some extra considerations and perchance some differences with the model 0.1. Model 0.1 is the original code of Guus. An overview is given for every subroutine. Questions and errors are generally written in *italic*. Also an outlook is presented, of what should be done in the (very) near future.

3.1 The current state of the model

Program test_flowmodel

This is the core of the program, all subroutines are called from here.

The only change made here is from defining v1=v0 to v0=v1.

Subroutine allocate_read_grid and initialize

In this subroutine the grid and the bathymetry are defined. In figure 3.1 is the grid schematically shown, including the subgrid cells, velocity points and surface elevation points. The other parameters are defined in the subroutine Initialize. Velocities are defined on the main grid, from 0 to $nmax$, $mmax$, but also at the boundary column and row $mmax + 1$ and $nmax + 1$. This yields also for the surface elevation and the bottom topography. Discharges and cross-sections are only defined in the main grid. The bathymetry is defined up to subgrid level. The elevation points of the subgrid are also located in the center of the cell. The depth (H) is defined according to figure 2.1, it might be more intuitively to define the depth according to figure 3.2. The initial condition of the elevation is appointed

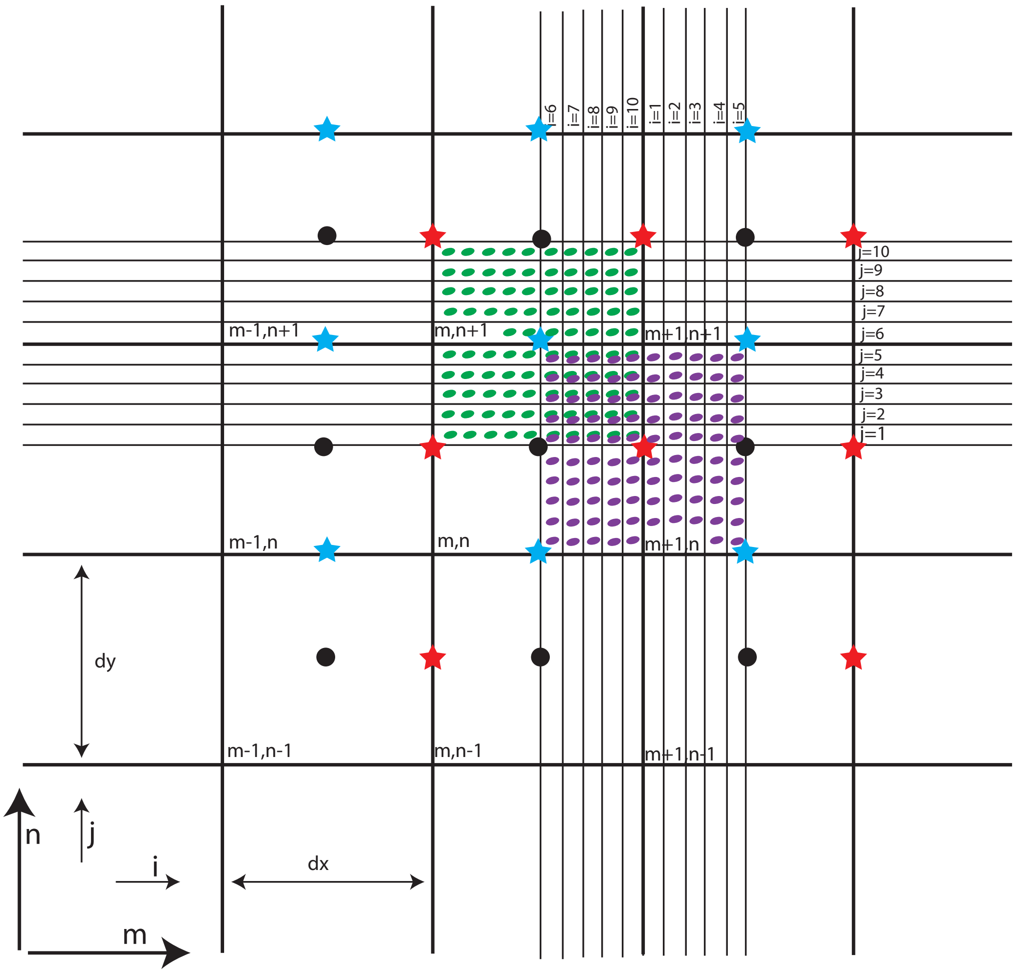

Figure 3.1: The grid appointing u and v velocity points with the red and blue stars respectively, the black dots are the locations of the calculated surface elevation. The green and the purple dotted areas appoint the subgrid level of interest for calculating the friction term.

in combination with the bottom topography. This to ensure non-negative depths. The elevation is initially uniformly distributed over the domain.

To make sure that none of the areas fall dry, due to the tidal variations, the waterlevel is set to a certain value. Is there a reason, to use 0.1 as a reference value? Is this an useful measure for determination whether dry fall is possible?

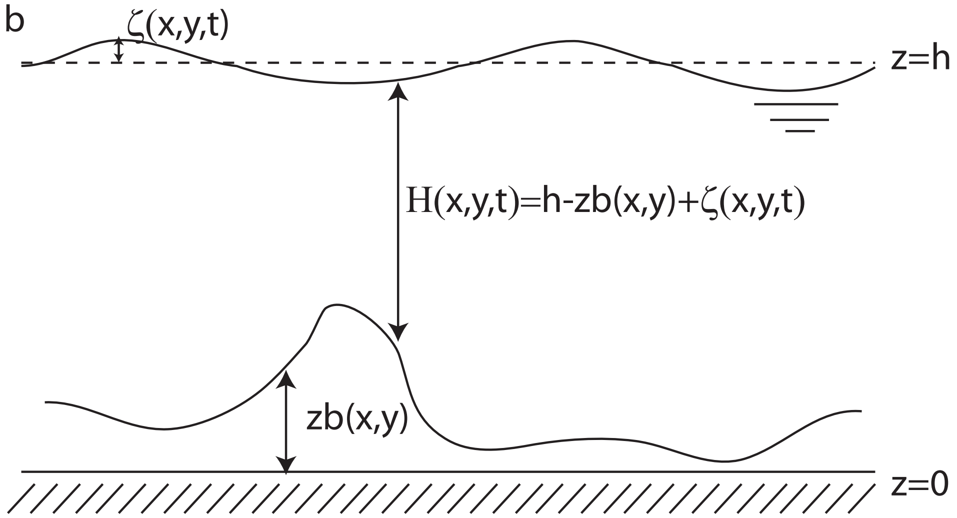

Figure 3.2: Suggested model principles to define the depth.

For a normal use of the ADI-method teta should be changed from 1 to a value between 0 and 1. But because initially the velocity in the y-direction is set to zero and diffusion and advective terms are neglected, nothing will happen in this direction. It is actually like a one dimensional system. The 'tidal' period is now defined at a few seconds, for speed, presumingly.

Subroutine timestep

In this routine the order of the subroutines are defined.

The subroutine curu is now evaluated twice, to make sure that if teta is not equal to one or a half, the system is still consistent. This is probably not a very elegant way of programming.

Subroutine curu

The interesting part of the model is this routine. Here, the friction term is defined at subgrid level. There are in this routine some aspects that need some considerations or some extra research. The friction term in the model is defined as;

$$fric = \Omega_{m,n,i,j} \parallel u \parallel u, \tag{3.1}$$

in which the absolute value of the velocity is generally defined as;

$$\parallel u \parallel = \sqrt{u^2 + v^2}. \tag{3.2}$$

The absolute value of the velocity uses the velocity at a velocity point in both directions. But only one of the two directions is defined at a velocity point. By interpolation of the four surrounding velocity points the value of the missing velocity is determined. A choice should be made between;

* using the squared value after averaging the four velocities. This is usually done in numerical models. In this way the sign of the velocity is considered. This result in special cases for a zero velocity at the intermediate point.

* using the average value of the four squared velocities. This, this drops the sign of the velocity. In the case of opposite velocities the value of the average is higher. This increases the friction term, which could be a correction for the extra shear in this situation.

Whether extra shear should be modeled in the friction term is not obvious, because the term is designed for bottom friction specifically. A suggestion is to check whether the difference between these terms is significant. The second possibility for defining the absolute value of the velocity is currently included in the model.

In this routine the friction term is defined at subgrid level. In section 2.2 it was shown that $\Omega^n_{m,n,i,j}$ is defined as

$$\bar{\Omega}^n_{m,n,i,j} = Cz\sqrt{\bar{h}^n_{m,n,i,j}} \tag{3.3}$$

There are again (at least) two reasonable options for the calculation of the frictional constant $\bar{\Omega}_{m,n}$. The first one is now imbedded in the program;

* this option uses the dotted areas in figure 3.1 to calculate the averaged depth. For determining the friction, to take the averaged depth around a velocity point (the dotted area is of the size of a grid cell) and not to take the averaged depth of the grid itself. Around a velocity point a weighed average is taken per subgrid cell of the two surface elevations surrounding the velocity point. This is added to the local bathometry, which is also defined at subgrid level, to get the total depth.

* An obvious alternative would be to use the subgrid cells of the grid belonging to the velocity points. In other words, not to use the overlaying variant of the subgrid. Because the model is calculating a new surface height using this frictional constant, which itself is defined at the center of a grid cell.

There are some minor changes made in this subroutine. The first one is a change inside the if-statement $i < im$ to $i \leq im$. This because otherwise the row numbered by $i = 10$ in *figure 3.1, would not be taken into account. The same yields for the second if-statement, where $j < im$ is changed to $j \leq im$.*

In the second big do-loop, the one calculating the cross-section in the y-direction, when defining 'sum1' a factor 'humnij' is added, which was missing.

To complete the code for a θ-method, an if statement is included. The constant θ is part of the definition of fac and cu, this is done in version 1.1.

Note that the subgrid velocity is not saved.

Subroutine comp_hu_au

In the model the space derivatives are defined upstream, this because of the stability of the model. This is used to calculate the wetted cross-sections and the transports. These are calculated in this subroutine for both directions. there are several possibilities to calculate the average depth,using the subgrid model. Some are listed here, it is possible to use different methods, for different purposes, i.e. it is not necessarily best to use the same depth for calculating a cross-section as in the case of the friction term. For the cross-sections it is possible to define the depth

* halfway the gridcell, at subgrid $i = im$, in the x-direction, at the waterlevel point. In this case the depth is averaged, using the subgrid values in the y-direction. This is executed in the up-stream cell. When the velocity is below $\parallel u \parallel \leq 1 * 10^6$ m/s, then the cross-section is determined by the maximal elevation between the cell and its neighbor and the averaged depth in the y-direction at the transection. This method is presently programmed, except for that the average depth is determined at the most far off side of the transection of the two cells. This might be a mistake and is therefore replaced by $i = imax$.

* by calculating the average depth at the boundary between the two gridcells. For determining the cross-section in the x-direction this would mean, using either the depth values at $i = imax$ or of the first subgrid values of the next gridcell, depending on the direction of the flow.

* as an average of all the subgrid depth values of the upstream cell. This is probably the value used, when a subgrid is not imposed on the model.

For the calculation of the cross-section for flows with a velocity of $\parallel u \parallel \leq 1 * 10^6$m/s, the expression $dps((m, n, 1, 1 : jmax))$ is changed to $dps((m, n, imax, 1 : jmax))$. The same yields for the other direction.

Subroutines s1uimpvexp, s1uexpvimp and SWEEP

The subroutines s1uimpvexp and s1uexpvimp are similar. They both calculate the water elevation and the velocities in both directions. Succeeding an implicit or explicit way of calculating the velocities. The explicit method is of course very straightforward. But, no explicit calculation is performed. Till now this was no problem, because θ is set to one. Moreover, initially v is set to zero and not having the advective and diffusion terms included v will always stay zero. Nevertheless, the explicit calculation is added to the code for consistency (NOG CONTROLEREN OF HET GOED IS). The Thomas' algorithm is used to calculate the velocity in the other direction and the water level. The constants used in subroutine SWEEP are already defined in equations 2.10-2.12 in section 2.1, as well as the details of this solver algorithm. The in section 2 defined coefficients $e_{m,n}$ and $f_{m,n}$ are not used. Instead, the other constants are continuously redefined to save 'work memory'. In the end the water level equals the coefficient $d_{m,n}$.

The minus signs of the coefficients of the matrix am and cm are not changed, even the definitions in the code and derived in chapter 2 do not match. But in a different book the coefficients are defined differently, currently I'm rewriting one into in the other to check whether they match. The boundaries should still be checked, which is something I will do.

Subroutine hisout

In this last subroutine the variables of interest are recorded with a certain frequency.

3.2 The (near) future

Model 0.1 is currently adjusted to a model in which 2 dimensional flows can be described, using an ADI-method. A few minus signs in subroutines s1uepxvimp and s1uimpvexp are still a bit questionable. Apart from this, the model is ready to be extended. First, an input file would be useful, where parameters and boundary conditions can be defined. It might be useful to include in the input file, the choice to save the variables of interest in for example a matlab-file. After that, the model should be tested and be compared with some analytical solutions and model results with and without a subgrid. When a dryfall procedure in imbedded in the code, then we could try to put a shoal into the estuary and see what will happen.

When, a working model is available, it is important to determine the effect of the subgrid. In which terms is it useful to implement a subgrid and what is the effect. Also in some terms different manners are possible to implement a subgrid, this became already clear in the routines described above. It should become clear, which method is best for which term. This is not necessarily the same for each term. In the end, it would be nice to quantify when the subgrid method, implemented in which term is useful and an improvement of the current models.

Chapter 4

The tests

To test, or not to test.

Chapter 5

Summary and conclusion

Subgrid is absolutely fabulous. Short, Subgrid is abfab!