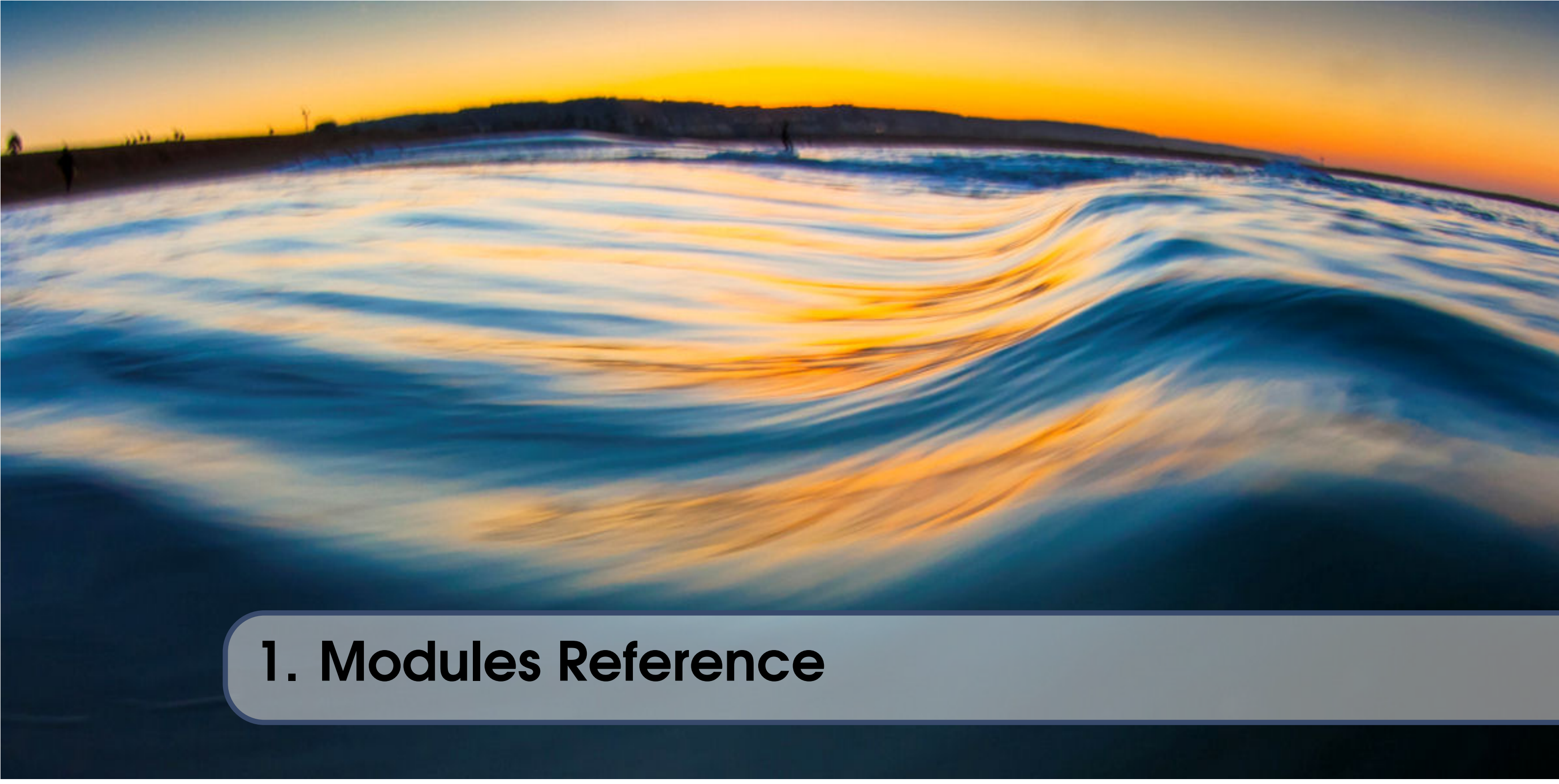
Numerical 2DV coupled frequency perturbation model

package for iFlow

Yoeri Dijkstra

Contents



1. Modules Reference

This chapter provides a short overview of all modules in the package `numerical2DV` and the required input and expected output. The modules have been ordered into sections for the purpose of providing structure to this chapter.

Explanation of terms and colours

Behind the input variables we will mention several data types. While some data types may be obvious, some others are explained in the table below:

Space-separated numbers	real numbers separated by one or more spaces. Do not use comma's or other markers to separate the numbers.
Grid-conform array n-dimensional	a numpy array with n (i.e. some number) or fewer (!) dimensions. More dimensions than n is not allowed. All axes should be grid conform. That means that the length of a dimension should either be 1 or equal to the size of the corresponding grid axis. If n is larger than the grid size, the length of this axis is free. Note that a single number counts as a grid-conform array.
General n-dimensional	either a grid-conform array or a numerical or analytical function. In both cases they may n (i.e. some number) or fewer dimensions.
iFlow grid	a grid variable with underlying subvariables as described in the manual (Dijkstra, 2017)

The cells with input variables have been colour-coded to indicate whether the variable is likely to be given in the input file, computed by another module or given in the configuration file. By the very nature of iFlow this is only indicative and depends on the modules used. As an example, almost any variable given in the input file may be used as a variable in a sensitivity analysis. It then becomes an input parameter of the sensitivity analysis module in the input file. The sensitivity analysis module delivers it to the module that uses this variable.

🟩	Likely a parameter in the input file
🟧	Either in the input file or from another module
🟥	Likely a parameter computed by another module
⬜	Likely a constant in the configuration file `src.config`

1.1 General

1.1.1 RegularGrid

Module for generating a standard computation and output grid with an along-channel (x), vertical (z) and frequency (f) dimension. See the iFlow manual (Dijkstra, 2017) for general information on grids. See Chapter 3 in this manual for specific information on the grids generated here.

Type		Normal
Submodules		None
Input	🟧 `L`	*Number*. Length of the system in the x-direction.
	🟧 `B`	*General 1-dimensional*. Width of the system.
	🟧 `H`	*General 1-dimensional*. Depth of the system between the reference level (i.e. water level at the mouth, typically mean sea level) and the bed.
	🟩 `xgrid, zgrid, fgrid`	*Space-separated values*. Specification for the axes of the standard computation grid. Options: 1) `'equidistant'noCells`. Equidistant axis with number of cells equal to noCells. Note that the number of nodes is noCells+1. 2) `'logarithmic'noCells, gamma`. Distribution as $(e^{\gamma X}-1)/(e^{\gamma}-1)$, with X an equidistant axis with noCells cells and steepness factor γ. 3) `'list' [values]`. Directly prescribe all grid nodes in a list with values. May be dimensionless between 0 and 1 or dimensional (end points need to be in the list). 4) `'integer'maxIndex`. Axis with integer steps, practical for discrete dimensions. This axis type does not have a dimensionless axis between 0 and 1. The argument maxIndex indicates the maximum index of this axis (inclusive). The axis will thus have maxIndex+1 elements. 5) `'file'filepath`. Reads grid points from an ASCII file. The file path incl. extension should be given as argument. The file should contain a single column of values between 0 and 1, corresponding to the grid nodes.
	🟩 `xoutputgrid, zoutputgrid, foutputgrid`	*Space-separated values*. Specification for the axes of the output grid, see above for options.
Output	`grid`	*iFlow grid*. Standard computation grid
	`outputgrid`	*iFlow grid*. Standard output grid.

1.1.2 HigherOrderIterator

Auxiliary iterative module. Starts the iteration for higher-order computations (i.e. second or higher order). The higher-order modules in this package compute one order in every iteration. The second order thus does not require this iteration, but, for computations higher than second-order, this iterator ensures that such an iteration is started.

Type		Iterative
Submodules		None
InputInit	`maxOrder`	*Integer.* Maximum order to be computed (inclusive).
	`variables`	*Space-separated strings* . Variables to compute higher-order of. Variable names should be given here without the integer marking the order at the end of their name.
	`@{variables}+{1,2}`	*Data type depends on variables entered.* First-order quantities of the variable names entered above.
Input	`@{variables}+{2,@{maxOrder}+1}`	*Data type depends on variables entered.* Higher-order quantities of the variable names entered above. This input requirement is purely technical and help iFlow to determine the extent of the iteration loop.
Output	`order`	*Integer.* Current order of computation.
	`maxOrder`	*Integer.* Copy of the input variable `maxOrder` to this module.

1.2 Hydrodynamics

1.2.1 HydroLead

Leading-order hydrodynamics using a numerical perturbation model. See Part I of this manual.

Type		Normal
Submodules	`tide`	externally forced tidal flow. Forced by input parameters `A0` and `phase0`.
	`river`	externally forced river flow. Forced by input parameter `Q0`.
Input	`BottomBC`	*string.* Bottom boundary condition type. Allows for values `'PartialSlip'` or `'NoSlip'`
	`Av`	*General 3-dimensional.* Vertical eddy viscosity in m^2/s.
	`roughness`	*General 3-dimensional. Second dimension is length 1.* Roughness coefficient s_f (if `BottomBC=='PartialSlip'`) or z_0 (if `BottomBC=='NoSlip'`). May vary x and time, but not in z. Therefore the second dimension needs to have length 1.
	`grid`	*iFlow grid.*
	`OMEGA`	*Number.* Angular frequency of the lowest-frequency component in rad/s
	`G`	*Number.* Acceleration of gravity in m^2/s
Input sub-modules	`A0`	Only `tide` *space-separated numbers.* Water level amplitude at the seaward boundary in metres. The first value corresponds to subtidal (should equal 0). The second value corresponds to the frequency with angular frequency ω (standard M_2 tide). The third value corresponds angular frequency 2ω (standard M_4) etc. The number of values should be smaller than or equal to the maximum resolved frequency (i.e. `fmax+1` in the grid).
	`phase0`	Only `tide`

		space-separated numbers. Water level phase at the seaward boundary in degrees. Similar to `A0`. First element should equal 0.
	`Q0`	Only `river` *number.* Leading-order river discharge at the landward boundary in m^3/s.
Output	`zeta0`	*Numerical function 3-dimensional. . Second dimension is length 1.* Leading-order water level elevation in metres. Saved as numerical function with its x-derivative.
	`u0`	*Numerical function 3-dimensional.* Horizontal flow velocity, saved as numerical function with its z-derivative.
	`w0`	*Array 3-dimensional.* Vertical flow velocity.

1.2.2 HydroFirst

First-order hydrodynamics using a numerical perturbation model. See Part I of this manual.

Type		Normal
Submodules	`tide`	externally forced tidal flow. Forced by input parameters `A1` and `phase1`.
	`river`	externally forced river flow. Forced by input parameter `Q1`.
	`adv`	internally generated flow by momentum advection.
	`nostress`	internally generated flow through velocity-depth-asymmetry; interactions between the velocity gradient (i.e. the shape of the velocity profile) and the water level.
	`stokes`	internally generated tidal return flow that compensates for the net mass transport in the leading order.
	`baroc`	flow induced by a horizontal density gradient.
	`mixing`	flow induced by first-order eddy viscosity contributions.
Input	`BottomBC`	*string.* Bottom boundary condition type, see module HydroLead.
	`Av`	*General 3-dimensional.* Vertical eddy viscosity in m^2/s, see module HydroLead.
	`roughness`	*General 3-dimensional. Second dimension is length 1.* Roughness coefficient s_f, see module HydroLead.
	`grid`	*iFlow grid.*
	`OMEGA`	*Number.* Angular frequency of the lowest-frequency component in rad/s
	`G`	*Number.* Acceleration of gravity in m^2/s
	`BETA`	*Number.* Conversion parameter for salinity in $\rho = \rho_0(1+\beta s)$
Input sub-modules	`A1`	Only `tide` *space-separated numbers.* Water level amplitude at the seaward boundary in metres, see module HydroLead.
	`phase1`	Only `tide` *space-separated numbers.* Water level phase at the seaward boundary in degrees, see module HydroLead.

	Q1	Only river *number*. First-order river discharge at the landward boundary in m^3/s.
	u0	Only stokes, nostress, adv, mixing *General 3-dimensional* Leading-order horizontal flow velocity (m/s).
	zeta0	Only stokes, nostress *General 3-dimensional* Leading-order water level elevation (m). Second dimension should be length 1.
	w0	Only adv *General 3-dimensional* Leading-order vertical flow velocity (m/s).
	s0	Only baroc *General 3-dimensional* Leading-order salinity (psu).
Output	zeta1	*Numerical function 3-dimensional. . Second dimension is length 1.* Leading-order water level elevation in metres. Saved as numerical function with its x-derivative.
	u1	*Numerical function 3-dimensional.* Horizontal flow velocity, saved as numerical function with its z-derivative.
	w1	*Array 3-dimensional.* Vertical flow velocity.

1.2.3 HydroHigher

Higher-order hydrodynamics using a numerical perturbation model. See Chapter 7 of this manual.
Use in combination with the module HigherOrderIteration.

Type		Normal
Submodules	adv	internally generated flow by momentum advection.
	nostress	internally generated flow through velocity-depth-asymmetry; interactions between the velocity gradient (i.e. the shape of the velocity profile) and the water level.
	stokes	internally generated tidal return flow that compensates for the net mass transport in the lower orders.
	baroc	flow induced by a horizontal density gradient.
	mixing	flow induced by higher-order eddy viscosity contributions.
	densitydrift	flow induced by the interaction between the horizontal density gradient and moving surface level.
Input	maxContributions	*Integer.* Maximum number of separate contributions saved per submodule.
	maxOrder	*Integer.* Maximum order to be computed (inclusive). Is output of the module HigherOrderIteration.
	BottomBC	*string.* Bottom boundary condition type, see module HydroLead.
	Av	*General 3-dimensional.* Leading-order vertical eddy viscosity in m^2/s, see module HydroLead.

	`roughness`	*General 3-dimensional. Second dimension is length 1.* Roughness coefficient s_f, see module HydroLead.
	`grid`	*iFlow grid.*
	`OMEGA`	*Number.* Angular frequency of the lowest-frequency component in rad/s
	`G`	*Number.* Acceleration of gravity in m^2/s
	`BETA`	*Number.* Conversion parameter for salinity in $\rho = \rho_0(1+\beta s)$
Input submodules	`u0, u1`	Only `stokes, nostress, adv, mixing, densitydrift` *General 3-dimensional* Leading- and first-order horizontal flow velocity (m/s).
	`zeta0`	Only `stokes, nostress, mixing, densitydrift` *General 3-dimensional* Leadingand first-order water level elevation (m). Second dimension should be length 1.
	`w0, w1`	Only `adv` *General 3-dimensional* Leading- and first-order vertical flow velocity (m/s).
	`Av+{2,@{maxOrder}+1}`	Only `mixing` *General 3-dimensional* Higher-order vertical eddy viscosity (m^2/s).
	`s+{1,@{maxOrder}}`	Only `baroc` *General 3-dimensional* Salinity (psu).
	`s+{0,@{maxOrder}-1}`	Only `densitydrift` *General 3-dimensional* Salinity (psu).
Output	`zeta+{2,@{maxOrder}+1}`	*Numerical function 3-dimensional. . Second dimension is length 1.* Water level elevation in metres. Saved as numerical function with its x-derivative.
	`u+{2,@{maxOrder}+1}`	*Array 3-dimensional.* Horizontal flow velocity.
	`w+{2,@{maxOrder}+1}`	*Array 3-dimensional.* Vertical flow velocity.
	`surfder`	*Array 5-dimensional.* Vertical derivatives of u at the surface (in 1/s), Mainly for internal use, but given as output for analysis purposes. The array is structured as (x, z (length 1), f, order of u, order of derivative).
	`surfstress`	*Array 5-dimensional.* Vertical shear divergence (in m/s^s) at the surface, Mainly for internal use, but given as output for analysis purposes. The array is structured as (x, z (length 1), f, order of u, order of derivative).

1.2.4 ReferenceLevel

Estimation of the river-induced subtidal water level set-up. This is used to set a reference level R and is used together with H to form the reference depth. This is especially useful if H is partially negative or small. See also Chapter 8 of this manual.

Type	Iterative
Submodules	None

InputInit	`Q0, Q1`	*Number.* Leading- and first-order river discharges (in m^3/s). If one is used, the other can be omitted. If both are set, only Q0 will be used.
	`H`	*General 1-dimensional.* Depth in metres (see RegularGrid).
Input	`BottomBC`	*string.* Bottom boundary condition type, see module HydroLead.
	`Av`	*General 3-dimensional.* Leading-order vertical eddy viscosity in m^2/s, see module HydroLead.
	`roughness`	*General 3-dimensional. Second dimension is length 1.* Roughness coefficient s_f, see module HydroLead.
	`B`	*General 1-dimensional.* Width of the system.
	`grid`	*iFlow grid.*
	`G`	*Number.* Acceleration of gravity in m^2/s
Output	`R`	*Array 1-dimensional.* Reference level (in metres), which is the estimated river-induced set-up of the zero-reference level (i.e. water level at the open boundary). the reference level is always zero at $x = 0$.

1.3 Sediment

1.3.1 SedDynamicLead

Leading-order sediment model, see Chapter 9.

Type		Normal
Submodules	`erosion`	Resuspension of sediment by the flow.
Input	`ws`	*General 3-dimensional.* Leading-order fall velocity (in m/s).
	`u0`	*General 3-dimensional.* Leading-order horizontal velocity (in m/s).
	`Av`	*General 3-dimensional.* Leading-order vertical eddy viscosity (in m^2/s).
	`grid`	*iFlow grid.*
	`sigma_rho`	*General 2-dimensional.* Prandtl-Schmidt number to convert the vertical eddy viscosity to a vertical eddy diffusivity as $K_v = \frac{A_v}{\sigma_rho}$.
	`G`	*Number.* Acceleration of gravity (in m/s^2).
	`OMEGA`	*Number.* Angular frequency of the slowest considered tidal frequency (standard M_2).
	`RHO0`	*Number.* Reference density of water (in kg/m^3).
	`DS`	*Number.* Typical sediment diameter (in m).
Output	`hatc0, a`	*Array 3-dimensional.* Leading-order sediment concentration, divided by the availability a (in kg/m^3). NB. this output variable consists of a main index `hatc0` and sub-index `a`.

1.3.2 SedDynamicFirst

First-order sediment model, see Chapter 9.

Type		Normal
Submodules	`erosion`	Resuspension of sediment by the flow.
	`sedadv`	Horizontal advection of sediment.
	`noflux`	Correction to the sediment concentration due to variations of the water level.
	`fallvel`	Effects of first-order changes to the fall velocity.
	`mixing`	Effects of first-order changes to the eddy diffusivity.
Input		Same as SedDynamicLead
	`hatc0, a`	Only for submodules `sedadv, noflux, fallvel, mixing` Leading-order sediment concentration, divided by the availability a (in kg/m^3).
	`w0`	Only for submodules `sedadv` Leading-order vertical velocity (in m/s).
	`ws1`	Only for submodules `fallvel` First-order fall velocity (in m/s).
	`Av1`	Only for submodules `mixing` First-order eddy viscosity (in m^2/s).
	`u1`	Only for submodules `erosion` First-order horizontal velocity (in m/s).
	`zeta0`	Only for submodules `noflux` Leading-order water level elevation (in m).
Output	`hatc1, a` `hatc1, ax`	*Array 3-dimensional.* First-order sediment concentration in two parts. One part divided by the availability a, the other part divided by a_x (in kg/m^3). The concentration is retrieved as $c^1 = \hat{c}_a^1 a + \hat{c}_{a_x}^1 a_x$. NB. this output variable consists of a main index `hatc1` and sub-indices `a` and `ax`.

1.3.3 SedDynamicSecond

Second-order sediment model, see Chapter 9.

Type		Normal
Submodules	`erosion`	Resuspension of sediment by the flow.
Input		Same as SedDynamicLead, except for `u0`
	`u1`	*General 3-dimensional.* First-order horizontal velocity (in m/s).
Output	`hatc2, a`	*Array 3-dimensional.* Second-order sediment concentration by river-induced resuspension, divided by the availability a (in kg/m^3). NB. this output variable consists of a main index `hatc2` and sub-index `a`.

1.3.4 StaticAvailability

Model for the water-bed exchange of sediment, resulting in the sediment availability, see Chapter 10.

Type		Normal

Submodules		None
Input	`Kh`	*General 1-dimensional.* Horizontal eddy diffusivity.
	`sedbc`	*String.* Type of boundary condition. Currently allows for `astar` and `csea` (see below).
	`@sedbc`	*Number.* If `sedbc` equals `astar`, use the domain-average availability a^* as input (dimensionless). Else use the depth-averaged subtidal concentration `csea` at the open boundary (in kg/m^3).
	`B`	`General 1-dimensional.` Width (in m).
	`zeta0`	*General 3-dimensional.* Leading-order water elevation (in m/s).
	`u0`	*General 3-dimensional.* Leading-order horizontal velocity (in m/s).
	`u1`	*General 3-dimensional.* First-order horizontal velocity (in m/s).
	`hatc0, a,` `hatc1, a,` `hatc1, ax,` `hatc2, a`	*General 3-dimensional.* Scaled sediment concentrations, see output of SedDynamicLead, SedDynamicFirst, SedDynamicSecond.
	`grid`	`iFlow grid.`
Output	`a`	*Array 1-dimensional.* Sediment availability (dimensionless).
	`c0`	*Array 3-dimensional.* Leading-order sediment concentration (in kg/m^3).
	`c1`	*Array 3-dimensional.* First-order sediment concentration (in kg/m^3).
	`c2`	*Array 3-dimensional.* Second-order sediment concentration due to rivder-induced resuspension (in kg/m^3).

1.4 Salinity

1.4.1 SalinityLead

Prognostic (resolving) salinity model numerical perturbation model for the leading-order salinity. The model assumptions lead to a fully well mixed salinity field to leading-order. For more explanation of the model we refer to McCarthy (1993); Wei et al. (2016).

Type		Normal
Submodules		None
Input	`ssea`	*Number.* Salinity (in psu) at the seaward boundary $x = 0$.
	`Q1`	*Number.* First-order river discharge at the landward boundary (in m^3/s).
	`Kh`	*General 1-dimensional.* Horizontal eddy diffusivity (in m^2/s).
	`H`	*General 1-dimensional.* Depth (in m), see also RegularGrid.
	`B`	*General 1-dimensional.* Width (in m), see also RegularGrid.
	`u0`	*General 3-dimensional.* Leading-order horizontal velocity (in m/s).
	`grid`	*iFlow grid.*
Output	`s0`	*Numerical function 3-dimensional.* Leading-order salinity, saved together with its x-derivative. Uniform in vertical (i.e. length 1 in z-direction).

	s1var	*Numerical function 3-dimensional.* Incomplete first-order salinity, saved together with its z-derivative. This quantity still needs to be closed by a higher-order balance.

1.4.2 SalinityFirst

Prognostic (resolving) salinity model numerical perturbation model for the leading-order salinity. For more explanation of the model we refer to McCarthy (1993); Wei et al. (2016).

Type		Normal
Submodules		None
Input	ssea	*Number.* Salinity (in psu) at the seaward boundary $x = 0$.
	Q1	*Number.* First-order river discharge at the landward boundary (in m^3/s).
	Kh	*General 1-dimensional.* Horizontal eddy diffusivity (in m^2/s).
	H	*General 1-dimensional.* Depth (in m), see also RegularGrid.
	B	*General 1-dimensional.* Width (in m), see also RegularGrid.
	u0	*General 3-dimensional.* Leading-order horizontal velocity (in m/s).
	u1	*General 3-dimensional.* First-order horizontal velocity (in m/s).
	s0	*General 3-dimensional.* Leading-order salinity.
	s1var	*General 3-dimensional.* Incomplete first-order salinity, still to be closed.
	grid	*iFlow grid.*
Output	s1	*Numerical function 3-dimensional.* First-order salinity, saved together with its x-derivative.
	s2var	*Numerical function 3-dimensional.* Incomplete second-order salinity, saved together with its z-derivative. This quantity still needs to be closed by a higher-order balance.

2. Introduction: approach and domain

Insight into the hydrodynamical mechanisms that govern the flow in estuaries is essential to learn more about the processes that govern the transport of sediment transport, oxygen or nutrients. This manual presents a detailed derivation and description of a two-dimensional idealised numerical package for iFlow that aims at this. This manual contains one separate chapter on the grid generating module and then four parts discussing:

1. Hydrodynamics
2. Turbulence models
3. Salinity
4. Sediment dynamics

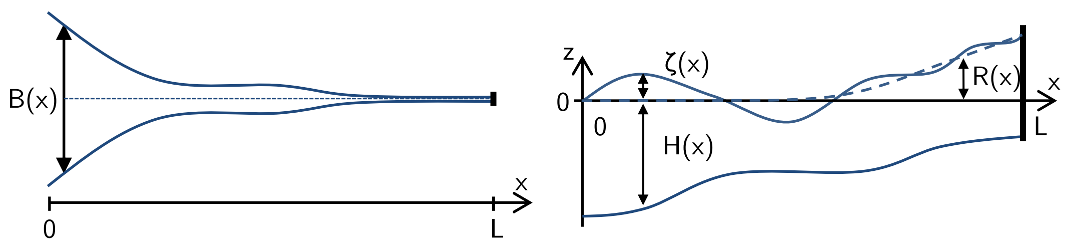
Every part of this manual will contain one or more chapters discussing the model equations, their derivation or solution method. The final chapter in each part contains a detailed description on the use of the provided iFlow modules.

The model is based on the perturbation approach, earlier adopted by e.g. Ianniello (1977, 1979); Chernetsky et al. (2010); Ensing et al. (tion) for hydrodynamics, McCarthy (1993); Wei et al. (2016) for salinity and Chernetsky et al. (2010); Ensing et al. (tion) for sediment dynamics. The perturbation approach involves a scaling of the equations to distinguish between the terms that balance at leading order and much smaller terms that balance at higher orders. Under suitable assumptions, the leading-order balance becomes linear and therefore much easier to solve than the original non-linear set of equations. The approach does however not neglect the non-linear terms or other higher-order terms. Instead, linear estimates of these terms appear as forcing mechanisms to linear higher-order balances. Theoretically, the full solution to the non-linear system is obtained when an infinite number of higher-order balances is solved for. Practically, we typically only solve for the leading-order and first-order balances, which provide a reasonably accurate estimate of the full solution. Due to the linearity of the equations at each order, the effect of different forcing mechanisms can be identified. These forcing mechanisms include externally applied flows and internally generated flows by non-linear effects and density gradients.

Compared to earlier perturbation models, one of the main unique and novel features this model is that it allows for leading-order temporal variations of turbulent mixing. In practice, the eddy viscosity varies strongly over a tidal cycle due to variations in the velocity or density gradient, giving rise to a range of interesting interactions and feedback mechanisms. Additionally the model supports any vertical profile of the eddy viscosity. The model also supports multiple tidal frequencies that are overtides of the M_2 tide, including the M_6 and M_8 tide.

The model domain is a two-dimensional width-averaged area as sketched in Figure 2.1. The width can be supplied in along-channel direction to account for changes of the width over the domain. The length of the estuary between the seaward boundary $x = 0$ and the landward boundary is denoted by L and can be freely chosen. The width, B, and depth, H, can be provided as arbitrary smooth functions of x. The depth H is relative to the mean sea level (MSL) defined at $z = 0$. Details on the functions that iFlow supports for the depth and width are provided in the manual on the auxiliary module package.

The surface level relative to $z = 0$ is expressed as $R + \zeta$ and is computed by the model. By default the reference level $R = 0$ and ζ is equal to the surface level. The use of a non-zero reference level is however required if the river bed is above MSL over part of the domain. The depth H is then negative, which poses a problem in further calculations. In this case iFlow computes the reference level R as a quick estimate of the mean surface level and ensures that $H + R$ is always positive. More details on the computation of R are provided in the part on hydrodynamics.



(a) Top view (b) Side view

Figure 2.1: Model domain. The model is two-dimensional in along-channel (x) and vertical (z) direction and is width-averaged. The depth and width are allowed to vary smoothly with x.

3. Grids

The numerical2DV package contains a module *RegularGrid* that generates computational and output grids. The grid is curvi-linear as exemplified in Figure 3.1. The grid is constructed by defining a dimensionless x-axis, $\hat{x}$, and a dimensionless z-axis, $\hat{z}$, both between 0 and 1. The dimensional axes x and z are then obtained by scaling the start and end points with the boundary locations, according to

$$x = \hat{x}L, \tag{3.1}$$

$$z(x) = -H(x)\hat{z} + R(x)\left(1-\hat{z}\right). \tag{3.2}$$

This implies that the x-axis is independent of z, while the z-axis depends on x. The dimensionless $\hat{z}$-axis is rescaled to fit between $-H(x)$ and $R(x)$.

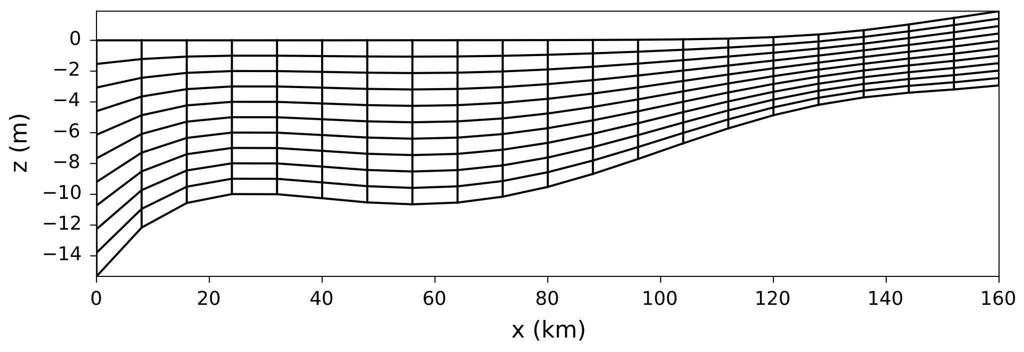


Figure 3.1: Example of a grid with 20 x 10 grid cells.

The dimensionless axes may be equidistant or non-equidistant. iFlow v2.3 defines the following types of axes

- 'equidistant'. Equidistant axis with grid points on the boundaries. It takes the maximum grid index as argument
- 'logarithmic'. Points are distributed as $\left(e^{\gamma X}-1\right)/(e^{\gamma}-1)$, where X is a set of equidistant

points and γ is a steepness factor. It takes the maximum grid index and steepness factor as arguments
- 'list'. Define axis directly by prescribing the coordinates. It takes a list of either dimensional or dimensionless values as input
- 'integer'. Axis with integer steps, practical for discrete dimensions. This axis type is an exception as it is not saved as points between 0 and 1 and does not require boundary definitions.
- 'file'. Reads grid points from an ASCII file. The file path should be given as an argument. The file should contain a single column of grid points between 0 and 1.

The *RegularGrid* module generates a working grid that becomes available as the standard grid variable `grid` and an output grid that becomes available as the standard grid for writing output `outputgrid`. The module requires the depth $H(x)$, width $B(x)$ and length L as input variables, for which it is recommended to use a geometry generating module. Furthermore *RegularGrid* needs input as given in Code sample 3.1. The input provides the input for the x, z and f axes for the working and output grid. The axis type can be selected from the list of axis types above. The axis arguments should correspond to the axis type. Note that the f-axis should typically be of the type `integer`.

■ Code sample 3.1 — input for RegularGrid.

```
module     numerical2DV.RegularGrid
xgrid      [axis type]    [axis arguments]
zgrid      [axis type]    [axis arguments]
fgrid      [axis type]    [axis arguments]

xoutputgrid   [axis type]    [axis arguments]
zoutputgrid   [axis type]    [axis arguments]
foutputgrid   [axis type]    [axis arguments]
```

■

Leading-order and first-order hydrodynamics

This part discusses the width-averaged equations for water motion, presents the reduction of the complexity of these equations using the perturbation approach and harmonic analysis and discusses the numerical implementation of the reduced model. A flow diagram of these steps is presented in Figure 3.2. Chapter 4 presents the equations and the first two steps in the model complexity reduction, the scaling and perturbation approach, to derive the leading-order and first-order systems of equations. Chapter 5 then presents the harmonic analysis, resulting in the novel coupled-frequency form of the equations. The numerical implementation is consequently discussed in Chapter 6. The extension of the model to arbitrary higher order is discussed in Chapter 7. Finally, Chapter **??** gives a description of the hydrodynamic modules provided in this package and the required input.

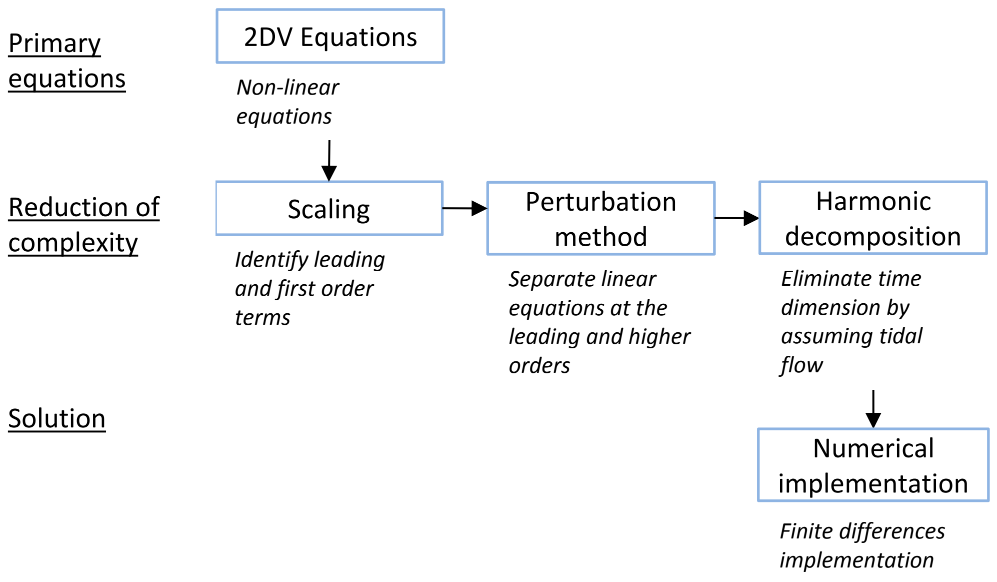

Primary equations
2DV Equations
Non-linear equations
Reduction of complexity
Scaling
Identify leading and first order terms
Perturbation method
Separate linear equations at the leading and higher orders
Harmonic decomposition
Eliminate time dimension by assuming tidal flow
Solution
Numerical implementation
Finite differences implementation

Figure 3.2: Flow diagram of the steps presented in this part on the presentation of the equations for hydrodynamics, the reduction of the complexity of these equations and the numerical implementation.

4. Equations and perturbation approach

The starting point for this chapter are the Reynold-averaged width-averaged momentum, continuity and depth-averaged continuity equations and boundary conditions (see e.g. Chernetsky et al., 2010). In these equations we neglect the effects of Coriolis and assume that density variations are small compared to the average density, allowing for the Boussinesq approximation. The equations solve for the horizontal velocity u, vertical velocity w and water level ζ in the horizontal and vertical coordinates x and z and time t.

2D equations

The equations read

$$u_t + uu_x + wu_z = -g(R_x + \zeta_x) - g\int_z^{R+\zeta} \frac{\rho_x}{\rho_0}\, d\tilde{z} + (A_v u_z)_z, \qquad \text{(momentum balance)} \qquad (4.1)$$

- $A_v u_z(x, R+\zeta, t) = 0,$ (no-stress) (4.2)
- $A_v u_z(x, -H, t) = s_f u(x, -H, t),$ (partial slip), or (4.3)

  $u(x, -H, t) = 0,$ (no-slip) (4.4)

$$\zeta_t + \frac{1}{B}\left(B\int_{-H}^{R+\zeta} u\, dz\right)_x = 0, \qquad \text{(Depth-averaged continuity)} \qquad (4.5)$$

- $\zeta(0,t) = A(t),$ (tidal forcing) (4.6)
- $B(L)\int_{-H(L)}^{R(L)+\zeta(L,t)} u(L,z,t)\, dz = Q.$ (river forcing and tidal reflection) (4.7)

$$w_z + \frac{1}{B}(Bu)_x = 0, \qquad \text{(continuity)} \qquad (4.8)$$

- $w(x, -H, t) + u(x, -H, t)H_x = 0.$ (kinematic) (4.9)

Here g is the acceleration of gravity, ρ is the density with reference density ρ_0. The density follows from the salinity s (in PSU) according to a linear equation of state: $\rho = \rho_0(1 + 7\cdot 10^4 s)$. The eddy viscosity is denoted by A_v and s_f is a partial slip roughness parameter. Which of the bottom boundary conditions (4.3) or (4.3) is used depends on the formulation for the eddy viscosity and is discussed in more detail below. $A(t)$ is the tidal forcing at the seaward boundary $x = 0$ and Q is the river discharge, The subscripts x, z and t in the equations denote derivatives with respect to these dimensions.

Forcing

The water motion is forced by a periodic water level variation $A(t)$ at the seaward boundary $x = 0$, see Equation (4.6). The allowed boundary signal consists of the semi-diurnal M_2 tide (i.e. 12.42 hour periodic) and any of its overtides. At the landward boundary $x = L$, the model is forced by a constant river discharge Q, see Equation (4.7). Typically it is assumed that the river-induced velocity is much smaller than the typical M_2 tidal velocity, so that it only appears at the first order. However, leading-order river discharges are allowed in the model. The tidal wave is reflected at the landward boundary, simulating the effect of a tidal weir or locking complex. The water motion is further forced by a horizontal density gradient, ρ_x in Equation 4.1. This density gradient follows from a prescribed horizontal salinity gradient that is constant in time. The density forcing is assumed to be a first-order effect in both solution methods.

4.1 Scaling

Scaling assumptions

The model equations are analysed using a mathematical scaling analysis. This analysis is used to identify which terms balance at the leading order and which terms are much smaller. The present scaling analysis pivots around three crucial assumptions.

1. We define
$$\varepsilon = \frac{A_{M_2}}{H_0} \ll 1, \tag{4.10}$$
i.e. the ratio of the typical water level amplitude to the typical depth is much smaller than unity. The small parameter ε will be used to define first-order effects. A term is therefore said to be of leading order if it is $\mathcal{O}(1)$ in its scaled dimensionless form and of first-order if it is $\mathcal{O}(\varepsilon)$.
2. The typical tidal wave length and the typical length-scale of bathymetrical variations are of the same order of magnitude as the length of tidal influence into the estuary. This implies that sudden, local bathymetry variations are not allowed. Rather, bathymetrical changes should be smooth over the length of the estuary.
3. The horizontal density gradient is small. More precisely the internal Froude number should be of the order of ε or equivalently $\rho_x L/\rho_0$ should be of the order of ε^2, where L is the length of tidal influence. As a consequence, the baroclinic pressure term $g\int_z^{R+\zeta} \frac{\rho_x}{\rho_0}\, d\tilde{z}$ in Equation 4.1 is of the order of ε. These three assumptions result in all non-linearities in u, w and ζ to be first-order effects.

Typical scales

For the scaling, the equations are transformed to a dimensionless system in order to establish the order of magnitude of each term. The equations are scaled by using six typical scales, which are presented in Table 4.1. The table presents three more scales that are derived from the other six. The velocity scale U follows from the scaling of the depth-averaged continuity equation (4.5), which in dimensionless form reads

$$\frac{A_{M_2}}{T_{M_2}}\zeta^*_{t^*} + \frac{(H_0 + A_{M_2})U}{L_{\text{tide}}}\int_{-H^*}^{R^*+\zeta^*} (H^*\overline{u}^*)_{x^*}\, dz + \frac{H_0 U}{L_{\text{conv}}}\int_{-H^*}^{R^*+\zeta^*} H^*\overline{u}^*\, dz = 0.$$

It follows that the velocity scale U must obey

$$U = \frac{A_{M_2}}{T_{M_2}} \frac{\min(L_{\text{tide}}, L_{\text{conv}})}{(H_0 + A_{M_2})} \approx \frac{A_{M_2}}{T_{M_2}} \frac{L}{H_0},$$

where it is assumed that $H_0 \gg A_{M_2}$ (using assumption 1 above). The scale of the velocity thus depends on the governing length scale, which is either the tidal wave length scale or the typical scale at which the estuary geometry changes. It is assumed that these length scales are both of the same order of magnitude as the length of tidal influence in the estuary (assumption 2 above), such that

$$\mathcal{O}(L_{\text{tide}}) = \mathcal{O}(L_{\text{conv}}) = L.$$

This requires the estuarine geometry to vary gradually.

Similar to U, W is derived from the continuity equation 4.8. It follows that

$$W = \frac{H_0}{L} U.$$

The typical eddy viscosity follows from the uniform, stationary barotropic momentum balance $(A_v u_z)_z = g\zeta_x$. It follows that

$$\mathcal{N} = \frac{H_0^2}{T_{M_2}}.$$

Scale		Dimensionless quantity
T_{M_2}	M_2 tidal period	$t = T_{M_2} t^*$
A_{M_2}	M_2 tidal amplitude at the seaward side	$\zeta = A_{M_2}\zeta^*$
H_0	Average depth at seaward side	$z = H_0 z^*$ and $R = A_{M_2} R^*$
L_{tide}	Tidal wave length	$x = L_{\text{tide}} x^*$ for u_x, w_x, ζ_x
L_{conv}	convergence length	$x = L_{\text{conv}} x^*$ for H_x, B_x
$\mathcal{R}_x$	Typical density gradient	$\rho_x = \mathcal{R}_x \rho^*_{x^*}$

Scale		Dimensionless quantity
U	Typical horizontal velocity of the M_2 tide	$u = U u^*$
W	Typical vertical velocity of the M_2 tide	$w = W w^*$
$\mathcal{N}$	Typical eddy viscosity	$A_v = \mathcal{N} A_v{}^*$

Table 4.1: Scales (upper table) and derived scales (lower scales) for deriving the dimensionless equations.

Reference level

For the scaling of the reference level R, we will assume that the actual magnitude scales with the local depth and is therefore best made dimensionless as $R = H_0 R^*$. However, the horizontal gradient scales with the horizontal water level gradient and scales as $R_x = \frac{A_{M_2}}{L_{\text{tide}}} R^*_x$. This seemingly inconsistent scaling is required in systems where the depth H relative to $z = 0$ varies from $\mathcal{O}(H_0)$ at the mouth to $\mathcal{O}(A_{M_2})$ further upstream. The reference level is assumed to be of the same order of magnitude as the depth further upstream, but since the depth is scaled by H_0 everywhere, R also needs to be scaled with H_0. However, the variation of R from a value of zero the mouth to $\mathcal{O}(A_{M_2})$ further upstream yield that R_x scales with $\frac{A_{M_2}}{L_{\text{tide}}}$.

Momentum equation

Using these dimensionless variables, the dimensionless momentum equation (4.1) is can be written as

$$u^*_{t^*} + \frac{A_{M_2}}{H_0} u^* u^*_{x^*} + \frac{A_{M_2}}{H_0} w^* u^*_{z^*} = -gH_0 \frac{T^2_{M_2}}{L^2}(R^*_x + \zeta^*_x) - \frac{g}{\rho_0}\frac{H_0 T_{M_2}\mathcal{R}_x}{U}\int_{z^*}^{R^*+\varepsilon\zeta^*} \rho^*_{x^*}\, d\tilde{z} + H_0^2 (A_v{}^* u^*_{z^*})_{z^*}.$$

The factor $\frac{A_{M_2}}{H_0}$ in front of the momentum advection term $uu_x + wu_z$ denotes the ratio of the typical tidal amplitude and the typical depth. This parameter will be called ε and is assumed to be much smaller than unity, i.e.

$$\varepsilon = \frac{A_{M_2}}{H_0}.$$

The other factors that appear in the dimensionless momentum equation can be related to the magnitude of ε. These factors are considered below. Firstly, $gH_0\frac{T^2_{M_2}}{L^2}$ can be rewritten by using that $\sqrt{gH_0}$ equals the barotropic shallow-water wave velocity c_E. This wave velocity can also be estimated as $c_E = \frac{L}{T_{M_2}}$. It follows that

$$gH_0 \frac{T_{M_2}^2}{L^2} = 1.$$

Secondly, $\frac{g}{\rho_0}\frac{H_0 T_{M_2}\mathscr{R}_x}{U}$ is simplified to

$$\frac{g}{\rho_0}\frac{H_0 T_{M_2}\mathscr{R}_x}{U} = \frac{c_I}{U} = \frac{\mathscr{R}_x L}{\rho_0}\frac{H_0}{A_{M_2}},$$

where c_I is the baroclinic (internal) wave velocity, defined as $c_I = c_E\frac{\mathscr{R}_x L}{\rho_0}$. The term $\frac{c_I}{U}$ is also known as the internal Froude number. This is assumed to be of the order of ε. This gives an order-of-magnitude estimate for the allowable density gradient $\mathscr{R}_x$:

$$\frac{\mathscr{R}_x L}{\rho_0} = \mathscr{O}(\varepsilon^2).$$

The integral boundaries in the baroclinic term contain order ε terms. It is therefore reasonable to linearise the integral boundary around $z = R$ (i.e. $\zeta = 0$) by a Taylor expansion to find

$$\int_{z^*}^{R^*+\varepsilon\zeta^*} \rho_{x^*}^*\, d\tilde{z} = \int_{z^*}^{R^*} \rho_{x^*}^*\, d\tilde{z} + \varepsilon\rho_{x^*}^*(R^*)\zeta^* + \dots$$

The dimensional momentum equation then has terms of the following order of magnitude:

$$\underbrace{u_t}_{\mathscr{O}(1)} + \underbrace{uu_x}_{\mathscr{O}(\varepsilon)} + \underbrace{wu_z}_{\mathscr{O}(\varepsilon)} = \underbrace{-g\zeta_x}_{\mathscr{O}(1)} + \underbrace{g\int_z^R \frac{\rho_x}{\rho_0}\, d\tilde{z}}_{\mathscr{O}(\varepsilon)} + \underbrace{g\zeta\frac{\rho_x(R)}{\rho_0}}_{\mathscr{O}(\varepsilon^2)} + \underbrace{(A_v u_z)_z}_{\mathscr{O}(1)}$$

The dimensionless form of the depth-averaged continuity equation 4.5 is Depth-averaged continuity

$$\zeta_{t^*}^* + \frac{1}{B^*}\left(B^*\int_{-H^*}^{R^*+\varepsilon\zeta^*} u^*\, dz^*\right)_{x^*} = 0.$$

All terms are of the same order, except for the integration boundary ζ, which is of order ε. This is again linearised using a Taylor expansion according to

$$\int_{-H}^{R+\zeta} u\, dz = \int_{-H}^{R} u\, dz + \zeta u(x,R,t) + \dots .$$

The dimensional equation then has terms of the following order of magnitude:

$$\underbrace{\zeta_t}_{\mathscr{O}(1)} + \underbrace{\frac{1}{B}\left(B\int_{-H}^{R} u\, dz\right)_x}_{\mathscr{O}(1)} + \underbrace{(\zeta u(x,R,t))_x}_{\mathscr{O}(\varepsilon)} = 0.$$

The continuity equation (4.8) finally has only two terms, which balance at leading order ($\mathscr{O}(1)$). Continuity

The surface boundary condition (4.2) needs to be linearised to apply it at our fixed Boundary conditions reference level $z = R$. This linearisation follows from a Taylor expansion

$$A_V u_z(x, R+\zeta, t) = \underbrace{A_V u_z(x, R, t)}_{\mathscr{O}(1)} + \underbrace{A_V u_{zz}(x, R, t)\zeta}_{\mathscr{O}(\varepsilon)} + \dots.$$

The ordering of this boundary condition in terms of ε is obtained by observing that $z^* = \frac{A_{M_2}}{H_0}\zeta^* = \mathscr{O}(\varepsilon)$. The bottom boundary condition (4.3) or (4.4) has maximum two terms, which should therefore balance. This balance is at the leading order ($\mathscr{O}(1)$).

The horizontal boundary condition at the seaward side ($x = 0$) is separated into a leading-order and a first-order tidal forcing. The leading-order tidal forcing can theoretically consist of multiple harmonic components, but we will always assume it to consist of a single M_2 tidal component. The first-order forcing can similarly consist of any number of over-tides of the M_2 tide, but we will assume it to consist of a single M_4 tidal forcing. The resulting ordered version of the boundary condition then reads

$$\zeta(0,t) = \underbrace{A_{M_2}\cos(\omega t)}_{\mathscr{O}(1)} + \underbrace{A_{M_4}\cos(2\omega t - \phi)}_{\mathscr{O}(\varepsilon)}.$$

The river discharge prescribed at the landward side can be prescribed as either a leading-order or a first-order forcing, depending on its strength. Whether the river discharge is a leading-order or first-order mechanisms follows from writing the landward boundary condition (4.7) in a dimensionless form:

$$\int_{-H^*}^{\varepsilon\zeta^*(L,t)} u^*(L,\tilde{z},t)\,d\tilde{z} = \frac{Q}{UH_0B} = \begin{cases} \mathscr{O}(1) & \text{(leading order)} \\ \mathscr{O}(\varepsilon) & \text{(first order)} \end{cases}.$$

The condition compares the typical river-induced velocity to the typical tidal velocity. Note that this ratio increases to infinity near the end of the tidal influence, where the tidal velocity is by definition small compared to the river discharge. The normative river velocity and tidal velocity for determining whether the river is a leading or first order effect are those in the main area of interest. The upper bound of the integral in the boundary condition is linearised around $z = 0$ by using a Taylor expansion. The expression then reads

$$\underbrace{\int_{-H}^{R} u(L,z,t)\,dz}_{\mathscr{O}(1)} + \underbrace{\zeta(L,t)u(L,0,t)}_{\mathscr{O}(\varepsilon)} + \dots = \underbrace{\frac{Q}{B}}_{\mathscr{O}(1)\text{ or }\mathscr{O}(\varepsilon)}$$

4.2 Ordering & overview of the equations

In addition to the scaling of the equations, we will write the solution variables u, w and ζ as an asymptotic series ordered in the small parameter ε, i.e.

$$u = u^0 + u^1 + u^2 + \dots, \tag{4.11}$$

$$w = w^0 + w^1 + w^2 + \dots, \tag{4.12}$$

$$\zeta = \zeta^0 + \zeta^1 + \zeta^2 + \dots, \tag{4.13}$$

where u^0, w^0 and ζ^0 are of leading order, u^1, w^1 and ζ^1 are assumed to be of order ε, u^2, w^2 and ζ^2 are of order ε^2 etcetera. Furthermore we assume that the eddy viscosity A_v has a similar ordering

$$A_V = {A_V}^0 + {A_V}^1 + {A_V}^2 + \dots. \tag{4.14}$$

In the following, it will be assumed that the reference level R follows the river-induced sub-tidal water level set-up. To denote this we will write R_x as $R_x^0 + R_x^1$. If the river is taken into account at the leading order, it is assumed that the reference level is fully captured by R_x^0 and $R_x^1 = 0$. If the river is only taken into account at the first order, R_x^0 is ignored (i.e. $R_x^0 = 0$). Since this is no formal ordering, we will keep R without order notation wherever possible.

Substituting these series in the momentum, continuity and depth-averaged continuity equations, we obtain a system of equations that is identical to (4.1)-(4.9), but that is ordered in ε. By construction, the terms of first and higher orders are much smaller than the terms of leading order. As a first approximations, these terms are therefore neglected to obtain the leading-order system

$$u_t^0 - ({A_v}^0 u_z^0)_z = -g(R_x^0 + \zeta_x^0), \tag{4.15}$$

- $${A_v}^0 u_z^0(x,R,t) = 0, \tag{4.16}$$
- $$\begin{cases} {A_v}^0 u_z^0(x,-H,t) = s_f u^0(x,-H,t) & \text{(partial-slip) or} \\ u^0(x,-H,t) = 0 & \text{(no-slip).} \end{cases} \tag{4.17}$$

$$\frac{1}{B}(Bu)_x^0 + w_z^0 = 0, \tag{4.18}$$

- $$w^0(x,-H,t) = -u^0(x,-H,t)H_x. \tag{4.19}$$

$$\zeta_t^0 + \frac{1}{B}\left(\int_{-H}^{R} Bu^0\right)_x = 0, \tag{4.20}$$

- $$\zeta^0(0,t) = A^0(t), \tag{4.21}$$
- $$\int_{-H}^{R} u^0(L,z,t)\,dz = \frac{Q^0}{B}. \tag{4.22}$$

The residual after computing the leading-order system is dominated by the first-order terms. Neglecting the second and higher order we obtain the first-order system, which is an approximation for the residual. We define a short-hand notation for the forcing terms to the first-order system, reading

$$\varsigma^1(x,z,t) = g\int_{-H}^{R} \frac{\rho_x}{\rho_0}\,dz, \qquad \text{(Baroclinic)} \tag{4.23}$$

$$\eta^1(x,z,t) = u^0(x,z,t)u_x^0(x,z,t) + w^0(x,z,t)u_z^0(x,z,t), \qquad \text{(Advection)} \tag{4.24}$$

$$\psi^1(x,z,t) = {A_v}^1(x,z,t)u_z^0(x,z,t). \qquad \text{(Turbulence)} \tag{4.25}$$

$$\gamma^1(x,t) = \zeta^0(x,t)u^0(x,R,t), \qquad \text{(Tidal return flow)} \tag{4.26}$$

$$\chi^1(x,t) = \zeta^0(x,t)\left({A_v}^0(x,R,t)u_z^0(x,R,t)\right)_z. \qquad \text{(No-stress)} \tag{4.27}$$

Using this notation we obtain the first-order system

$$u_t^1 - ({A_v}^0 u_z^1)_z = -g(R_x^1 + \zeta_x^1) + \varsigma^1 + \psi_z^1 - \eta^1, \tag{4.28}$$

- $${A_v}^0 u_z^1(x,R,t) = -\chi^1(x,t) - \psi(x,R,t) = 0, \tag{4.29}$$
- $$\begin{cases} {A_v}^0 u_z^1(x,-H,t) - s_f u^1(x,-H,t) = -\psi(x,-H,t), & \text{(partial-slip), or} \\ u^1(x,-H,t) = 0 & \text{(no-slip).} \end{cases} \tag{4.30}$$

$$\frac{1}{B}(Bu^1)_x + w_z^1 = 0, \tag{4.31}$$

- $$w^1(x,-H,t) = -u^1(x,-H,t)H_x. \tag{4.32}$$

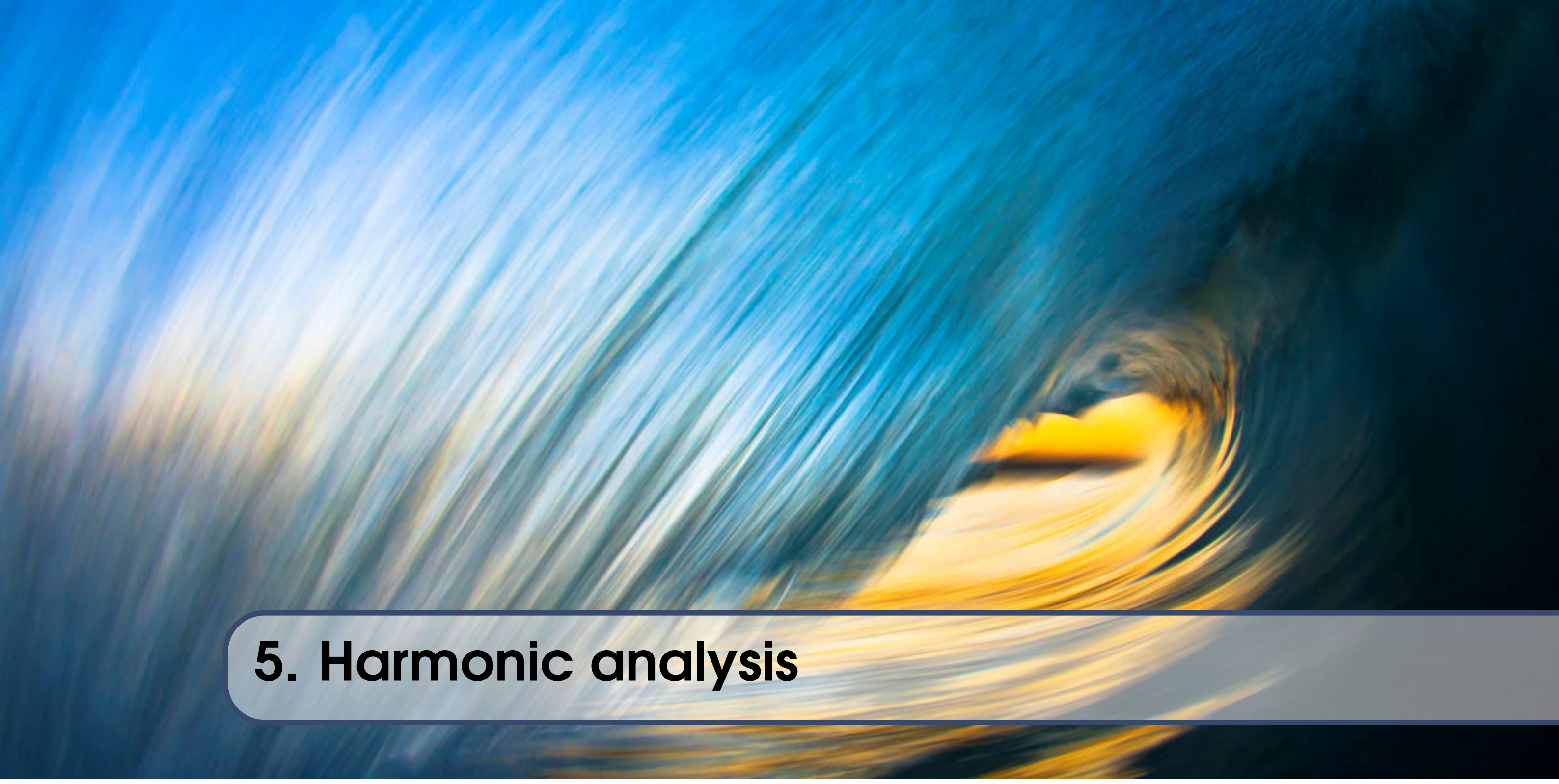
$$\zeta_t^1 + \frac{1}{B}\left(\int_{-H}^{R} Bu^1\right)_x + \frac{1}{B}\left(B\gamma^1\right)_x = 0, \tag{4.33}$$

- $$\zeta^1(0,t) = A^1(t), \tag{4.34}$$
- $$\int_{-H}^{R} u^1(L,z,t)\,dz = \frac{Q^1}{B} - \gamma^1(L,t). \tag{4.35}$$

The leading-order system contains two forcing mechanisms, while the first-order system contains seven. These mechanisms listed in Table 4.2.

Short name	Explanation	Symbol	Order
Tide	Tidal amplitude forced at the seaward boundary	A	0 and 1
River	Constant river discharge at the landward boundary	Q	0 or 1
Baroclinic	Forcing by the along-channel baroclinic pressure gradient	ς	1
Advection	Effect of momentum advection $uu_x + wu_z$	η	1
Tidal return flow	The return flow required to compensate for the mass flux induced by Stokes drift	γ	1
Turbulence	Effect of higher-order turbulent diffusion	ψ	1
No-stress	Correction for the alteration of the velocity profile due to the application of the no-stress boundary condition at $z = R$ instead of the real surface $z = R + \zeta$	χ	1

Table 4.2: Separate forcing mechanisms to the water motion, the mathematical symbol used in the equations and the order at which these mechanisms appear.



5. Harmonic analysis

This chapter presents the derivation of a coupled-frequency form of the ordered equations derived in Chaper 4. This form is obtained by assuming that the solution to the equations consists of a sum of tidal components and a subtidal component. Here, we assume that all tidal components are overtides of the M_2 tide. This assumption allows us to eliminate the time derivatives from the equations and obtain sets of ordinary differential equations (ODEs). It will be shown that these ODEs take the form of a set of matrix equations, in which the tidal components are coupled.

We will express the solution to the equation u^i, w^i, ζ^i $(i = 1, 2)$ in terms of a sum of tidal components. To this end we approximate the solution by a complex Fourier series expansion. This takes the form

$$\begin{aligned} u^i(x,z,t) &= Re\left(\sum_{n=0}^{p} \hat{u}_n^i(x,z)e^{ni\omega t}\right), \\ w^i(x,z,t) &= Re\left(\sum_{n=0}^{p} \hat{w}_n^i(x,z)e^{ni\omega t}\right), \\ \zeta^i(x,t) &= Re\left(\sum_{n=0}^{p} \hat{\zeta}_n^i(x)e^{ni\omega t}\right). \end{aligned}$$

The quantities of the form $\hat{u}_n^i$ symbol indicate the complex amplitude of u in frequency component n and order i. The Fourier series expansion is cut-off after tidal component p, because a numerical computation can only handle a finite number of tidal components. Please note that the Fourier series approximation becomes exact when one takes the limit $p \to \infty$. Similarly we will expand the eddy viscosity, partial slip roughness parameter, density, tidal forcing and river forcing. The corresponding Fourier series read

$$\begin{aligned}
{A_v}^i(x,z,t) &= Re\left(\sum_{n=0}^{p} \hat{A}^i_{vt,m}(x,z)e^{ni\omega t}\right),\\
s^i_f(x,z,t) &= Re\left(\sum_{n=0}^{p} \hat{s}^i_{f,m}(x,z)e^{ni\omega t}\right),\\
\rho^i(x,z,t) &= Re\left(\sum_{n=0}^{p} \hat{\rho}^i_m(x,z)e^{ni\omega t}\right),\\
A^i(t) &= Re\left(\sum_{n=0}^{p} \hat{A}^i_m e^{ni\omega t}\right),\\
Q^i &= \hat{Q}^i_0.
\end{aligned}$$

The solution method is easiest when the solution variables are expressed using not only positive Fourier components, but also negative components. For the solutions u, w and ζ we therefore alternatively define the negative series

$$\begin{aligned}
u^i(x,z,t) &= Re\left(\sum_{n=-p}^{p} \breve{u}^i_n(x,z)e^{ni\omega t}\right),\\
w^i(x,z,t) &= Re\left(\sum_{n=-p}^{p} \breve{w}^i_n(x,z)e^{ni\omega t}\right),\\
\zeta^i(x,t) &= Re\left(\sum_{n=-p}^{p} \breve{\zeta}^i_n(x)e^{ni\omega t}\right).
\end{aligned}$$

It holds that $\hat{u}^i_n = \breve{u}^i_n + \overline{\breve{u}}^i_{-n}$ for $n = 1,\ldots,p$ and $\hat{u}^i_0 = \breve{u}^i_0$. Similar identities hold for w and ζ. In the following we will use a vector notation for the complex amplitudes. A vector will be denoted by an underline $\underline{.}$. We define:

$$\underline{\breve{u}}^i = [\breve{u}^i_{-p},\ldots,\breve{u}^i_0,\ldots,\breve{u}^i_p]. \tag{5.1}$$

Similar definitions are used for w and ζ. For the other quantities A_v, s_f, ρ, A and Q we vectors containing the complex amplitude and its complex conjugate. Illustrating this for A_v, we define

$$\underline{\hat{A}_v}^i = [\overline{\hat{A}^i_{vp}},\ldots,\overline{\hat{A}^i_{v1}},\hat{A}^i_{v0},\ldots,\hat{A}^i_{vp}]. \tag{5.2}$$

5.1 Illustration of the derivation

We will provide an elaborate derivation of the coupled-frequency form of the leading-order momentum equation, assuming $R^0_x = 0$ for the purposes of this illustration. The coupled-frequency forms of the other equations then follow from the same principles. First, the Fourier series defined above are substituted in the momentum equation. This yields:

$$Re\left(\sum_{n=-p}^{p} ni\omega\breve{u}^0_n(x,t)e^{ni\omega t}\right) - Re\left(\sum_{n=-p}^{p} \breve{\psi}^0_n e^{ni\omega t}\right)_z = -gRe\left(\sum_{n=-p}^{p} \breve{\zeta}^0_{x,n}(x)e^{ni\omega t}\right), \tag{5.3}$$

where we define

$$\psi^0 = {A_v}^0 u^0_z$$

and construct a Fourier series for this. Even though ${A_v}^0$ and u^0_z contain at maximum $p+1$ positive harmonic components, their product ψ may contain up to $2p+1$ positive harmonic components. This series is truncated to $p+1$ positive components so that

$$\begin{aligned}\psi &= \sum_{n=0}^{p} \hat{\psi}_n^0 e^{ni\omega t} \\ &= \sum_{n=-p}^{p} \breve{\psi}_n^0 e^{ni\omega t},\end{aligned}$$

with $\hat{\psi}_n^0 = \breve{\psi}_n^0 + \overline{\breve{\psi}}_{-n}^0$ for $n = 1, \ldots, p$ and $\hat{\psi}_0^0 = \breve{\psi}_0^0$.

We can eliminate the summation by gathering the factors in front of each exponential function $e^{ni\omega t}$ for a certain n, i.e.

$$Re\left(ni\omega \hat{u}_n^0(x,t)e^{ni\omega t}\right) = -gRe\left(\hat{\zeta}_{x,n}^0(x)e^{ni\omega t}\right) + Re\left(\hat{\psi}_m^0 e^{ni\omega t}\right)_z. \qquad (n \in [-p,p]) \tag{5.4}$$

This is possible because the exponential functions $e^{ni\omega t}$ ($n \in \mathbb{Z}$) form a set of orthogonal basis functions. The only way of satisfying equation (5.4) is then if the factors in front of each of the basis functions balance.

We will next eliminate the Re. This can be done by simply removing Re from the equations. Indeed, if two complex numbers are equal, then certainly their real parts are also equal. We also divide the equation by $e^{ni\omega t}$ to obtain

$$ni\omega \hat{u}_n^0(x,t) = -g\hat{\zeta}_{x,n}^0(x) + \left(\hat{\psi}_m^0\right)_z. \qquad (n \in [-p,p]) \tag{5.5}$$

We can show that the complex Equation (5.5) does not put stronger requirements on the solution than the real Equation (5.4). Let us consider the simplified real equation

$$Re\left(\hat{a}e^{i\omega t}\right) = Re\left(\hat{b}e^{i\omega t}\right), \tag{5.6}$$

for some complex numbers a and b. This evaluates to

$$Re(\hat{a})\cos(\omega t) + Im(\hat{a})\sin(\omega t) = Re(\hat{b})\cos(\omega t) + Im(\hat{b})\sin(\omega t).$$

By the orthogonality of the cosine and sine, this expression breaks down into two separate equations

$$\begin{aligned}Re(\hat{a}) &= Re(\hat{b}), \\ Im(\hat{a}) &= Im(\hat{b}).\end{aligned}$$

We see that the real Equation (5.6) for complex variables is equivalent to the complex equation $\hat{a} = \hat{b}$.

5.1.1 Product of two Fourier series

The final step in the derivation is finding an expression for $\breve{\psi}_n^0$ in terms of $\breve{u}_{n,z}^0$ and $\hat{A}_{\nu n}^0$. To this end, let us first try to find such an expression for quantities with only positive Fourier components, i.e. $\hat{\psi}_n^0$, i.e. $\hat{\psi}_n^0 = f(\hat{u}_{n,z}^0, \hat{A}_{\nu n}^0)$. Observe that a component u_m^0 (i.e. $Re\left(\hat{u}_m^0 e^{mi\omega t}\right)$) can be written as

$$u_m^0 = Re\left(\hat{u}_m^0 e^{mi\omega t}\right) = \frac{1}{2}\left(\hat{u}_m^0 e^{mi\omega t} + \overline{\hat{u}}_m^0 e^{-mi\omega t}\right),$$

where the overbar denotes the complex conjugate. The product of two specific components $\hat{A}_{\nu n}^0$ and $u_{m,z}^0$ can then be written as

$$\begin{aligned}A_{\nu}{}_n^0 u_{m,z}^0 &= \left[\hat{A}_{\nu n}^0 e^{ni\omega t} + \overline{\hat{A}}_{\nu}{}_n^0 e^{-ni\omega t}\right]\left[\hat{u}_m^0 e^{mi\omega t} + \overline{\hat{u}}_m^0 e^{-mi\omega t}\right] \\ &= \frac{1}{4}\left(\hat{A}_{\nu n}^0 \hat{u}_{z,m}^0 e^{i(n+m)\omega t} + \hat{A}_{\nu n}^0 \overline{\hat{u}}_{m,z}^0 e^{i(n-m)\omega t} + \overline{\hat{A}}_{\nu}{}_n^0 \hat{u}_{m,z}^0 e^{i(-n+m)\omega t} + \overline{\hat{A}}_{\nu}{}_n^0 \hat{u}_{m,z}^0 e^{i(-n-m)\omega t}\right) \\ &= \frac{1}{2}Re\left(\hat{A}_{\nu n}^0 \hat{u}_{m,z}^0 e^{i(m+n)\omega t} + \overline{\hat{A}_{\nu n}^0} \hat{u}_{m,z}^0 e^{i(m-n)\omega t}\right).\end{aligned} \tag{5.7}$$

It follows then that

$$\begin{aligned}\psi^0 &= \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} A_{\nu}{}^{0}_{n} u^0_{m,z} \\ &= \frac{1}{2}Re\left(\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\left(\hat{A}^0_{\nu n}\hat{u}^0_{m,z}e^{i(n+m)\omega t} + \overline{\hat{A}^0_{\nu n}}\hat{u}^0_{z,m}e^{i(m-n)\omega t}\right)\right).\end{aligned}$$

We will next define $\hat{A}^0_{\nu-n} = \overline{\hat{A}^0_{\nu n}}$ for $n > 0$. We then find

$$\begin{aligned}\psi^0 &= \frac{1}{2}Re\left(\sum_{n=-\infty}^{\infty}\sum_{m=0}^{\infty}\left(\hat{A}^0_{\nu n}\hat{u}^0_{m,z}e^{i(n+m)\omega t}\right)\right), \\ &= \frac{1}{2}Re\left(\sum_{k=-\infty}^{\infty}\sum_{m=0}^{\infty}\left(\hat{A}^0_{\nu k-m}\hat{u}^0_{m,z}e^{ki\omega t}\right)\right).\end{aligned}$$

We have found a double infinite series of positive and negative Fourier components

$$\psi^0 = Re\left(\sum_{k=-\infty}^{\infty}\breve{\psi}^0_k e^{ki\omega t}\right),$$

with $\breve{\psi}^0_k = \frac{1}{2}\sum_{m=0}^{\infty}\hat{A}^0_{\nu k-m}\hat{u}^0_{m,z}$. A series with positive Fourier components can be found by further rewriting to

$$\begin{aligned}\psi^0 &= \frac{1}{2}Re\left(\sum_{k=0}^{\infty}\sum_{m=0}^{\infty}\left(\hat{A}^0_{\nu k-m}\hat{u}^0_{m,z} + \hat{A}^0_{\nu -k-m}\hat{u}^0_{m,z}\right)e^{-ki\omega t}\right), \\ &= \frac{1}{2}Re\left(\sum_{k=0}^{\infty}\sum_{m=0}^{\infty}\left(\hat{A}^0_{\nu k-m}\hat{u}^0_{m,z} + \hat{A}^0_{\nu k+m}\overline{\hat{u}^0_{m,z}}\right)e^{ki\omega t}\right), \qquad (5.8)\end{aligned}$$

where for the last step it was used that $Re(a) = Re(\overline{a})$ for any complex number a. We then find $\hat{\psi}^0_n = \frac{1}{2}\sum_{m=0}^{\infty}\left(\hat{A}^0_{\nu k-m}\hat{u}^0_{m,z} + \hat{A}^0_{\nu k+m}\overline{\hat{u}^0_{m,z}}\right)$. The positive Fourier components $\hat{\psi}^0_n$ cannot be expressed as a linear combination of $\hat{u}^0_n$, but also require knowledge about the complex conjugate[1]. This observation is also expressed in Theorem below.

Since we are looking for linear solution methods for $\hat{u}^0$, we prefer the positive and negative components $\breve{\psi}^0$ over the positive components $\hat{\psi}^0$. As a consequence we also have to consider $\breve{u}^0$ instead of $\hat{u}^0$ for the calculations. We then find $\breve{\psi}^0_k = \frac{1}{2}\sum_{m=-\infty}^{\infty}\hat{A}^0_{\nu k-m}\breve{u}^0_{m,z}$. When considering the truncated series of $\breve{u}^0_n$ $n \in [-p,p]$, $\breve{\psi}^0_n$ is typically has non-zero for $n < -p$ and $n > p$. These components are neglected and the series for $\breve{\psi}^0_n$ is truncated to $n \in [-p,p]$.

Theorem 5.1.1 — Product of positive Fourier series. Let u and v be two real $\mathscr{L}^2$ signals and let their Fourier series be defined as $Re\left(\sum_{n=0}^{\infty}\hat{u}_n e^{nit}\right)$ and $Re\left(\sum_{n=0}^{\infty}\hat{v}_n e^{nit}\right)$ with $\hat{u},\hat{v} \in \mathbb{C}$. Then
1. $\psi = uv$ can be expressed as $Re\left(\sum_{n=0}^{\infty}\hat{\psi}_n e^{nit}\right)$ with $\hat{\psi} \in \mathbb{C}$.
2. However, $\hat{\psi}_n$ cannot be expressed as a linear combination of $\hat{u}_n$, $\forall n \geq 0$.

Proof. The first claim follows almost trivially from Fourier series theory. The signal ψ is real and $\psi \in \mathscr{L}^2$, as $\mathscr{L}^2$ is a Hilbert space. We can then apply Fourier series theory to find that $\exists!\hat{\psi}_n$ with $n \in \mathbb{Z}^+$ such that $\psi = Re\left(\sum_{n=0}^{\infty}\hat{\psi}_n e^{nit}\right)$.

For the second statement let us assume that we can write the coefficients $\hat{\psi}_k$ as linear

[1] Note that complex conjugation is not a linear operation

combination of the components of $\hat{u}$.

$$\hat{\psi}_k = \sum_{n=0}^{\infty} f_k\left(\left\{\hat{v}_m, m \in \mathbb{Z}^+\right\}\right)\hat{u}_n \qquad \forall k \in \mathbb{Z}^+. \tag{5.9}$$

By the uniqueness of Fourier coefficients we can compare 5.9 with the previously derived expression 5.8. This comparison yields:

$$\frac{1}{2}\sum_{n=0}^{\infty} \hat{u}_n\hat{v}_{k-n} + \overline{\hat{u}_n}\hat{v}_{k+n} = \sum_{n=0}^{\infty} f_k\left(\left\{\hat{v}_m, m \in \mathbb{Z}^+\right\}\right)\hat{u}_n \qquad \forall k \in \mathbb{Z}^+.$$

After eliminating the summation we see

$$\frac{1}{2}\hat{u}_n\hat{v}_{k-n} + \overline{\hat{u}_n}\hat{v}_{k+n} = f_k\left(\left\{\hat{v}_m, m \in \mathbb{Z}^+\right\}\right)\hat{u}_n \qquad \forall n,k \in \mathbb{Z}^+.$$

This equality can never be satisfied as the complex conjugate of $\hat{u}_n$ on the left-hand side of this equality can never be balanced by the right-hand side. ■

5.2 Leading-order equations

We substitute the Fourier series expansions into the leading-order equations 4.15, 4.31 and 4.33. We then obtain a set of equations for each frequency component. We can write this as a set of matrix equations using the vector notation of Expression 5.1.

5.2.1 Momentum equation

The above procedure results in the following form of the momentum equation:

$$D\underline{\breve{u}}^0 - \left(\mathcal{N}^0\underline{\breve{u}}^0_z\right)_z = -g\left(\underline{\breve{R}}^0_x + \underline{\breve{\zeta}}^0_x\right), \tag{5.10}$$

where D and $\mathscr{N}$ are $(2p+1)\times(2p+1)$ matrices of the form

$$D = \begin{bmatrix} -pi\omega & & & \emptyset \\ & -(p-1)i\omega & & \\ & & \ddots & \\ \emptyset & & & pi\omega \end{bmatrix}, \quad \underline{\breve{u}} = \begin{bmatrix} \breve{u}_{-p} \\ \breve{u}_{-(p-1)} \\ \vdots \\ \breve{u}_{p} \end{bmatrix},$$

$$\mathscr{N}^0 = \frac{1}{2}\begin{bmatrix} 2Re(\hat{A}^0_{v0}) & \overline{\hat{A}^0_{v1}} & \cdots & \overline{\hat{A}^0_{vp}} & & & \\ \hat{A}^0_{v1} & 2Re(\hat{A}^0_{v0}) & \ddots & \vdots & \ddots & \emptyset & \\ \vdots & \ddots & \ddots & \overline{\hat{A}^0_{v1}} & & \ddots & \\ \hat{A}^0_{vp} & \cdots & \hat{A}^0_{v1} & 2Re(\hat{A}^0_{v0}) & \overline{\hat{A}^0_{v1}} & \cdots & \overline{\hat{A}^0_{vp}} \\ & \ddots & & \hat{A}^0_{v1} & \ddots & \ddots & \vdots \\ & \emptyset & \ddots & \vdots & \ddots & 2Re(\hat{A}^0_{v0}) & \overline{\hat{A}^0_{v1}} \\ & & & \hat{A}^0_{vp} & \cdots & \hat{A}^0_{v1} & 2Re(\hat{A}^0_{v0}) \end{bmatrix}$$

The matrix $\mathscr{N}^0$ is a diagonal matrix only if all time-varying components of $A_v{}^0$ are zero. In such a case, Equation 5.10 reduces to a diagonal matrix equation. As a result all frequencies are uncoupled within the leading order. However, in general $A_v{}^0$ will contain time variations and all frequency components are coupled.

We will now adopt a more abstract notation of the above matrix equation using abstract linear operators. The abstract notation helps in deriving expressions for the continuity and depth-averaged continuity equations later. It also helps to show the similarities between the leading-order system and higher-order systems. An example of the notation that we will adopt is provided in Intermezzo 5.2.1

Intermezzo 5.2.1 — Abstract linear operators. We will illustrate the use of linear operators by a simple example. Consider the Poisson problem in one dimension in the domain $[0,1]$

$$u_{zz} = f, \qquad (z \in (0,1))$$

where f is an arbitrary real function and the problem is subject to homogeneous boundary conditions of a mixed type:

$$\alpha u(0) + \beta u_z(0) = 0, \qquad (\alpha, \beta \in \mathbb{R})$$
$$\gamma u(1) + \delta u_z(1) = 0, \qquad (\gamma, \delta \in \mathbb{R})$$

Let the linear function space $H^1_m(0,1)$ be the space of all once weakly differentiable functions between 0 and 1 (generally denoted by $H^1(0,1)$) that satisfy the above boundary conditions. We require our solution u to be in this space, i.e.

$$u \in H^1_m(0,1) = \left\{ H^1(0,1) \,|\, \alpha u(0) + \beta u_z(0) = 0,\ \gamma u(1) + \delta u_z(1) = 0,\ \alpha, \beta, \gamma, \delta \in \mathbb{R} \right\}$$

In the remainder of this manual we will not focus too much on the function spaces that the solutions are in. They are merely introduced here to explain how the abstract notation captures the boundary conditions.

We define the linear operator $\mathscr{A} : H^1_m(0,1) \to \mathbb{R}$ according to

$$\mathscr{A} = \frac{\partial^2}{\partial z^2}.$$

We then solve the problem

$$\mathscr{A} u = f.$$

Inhomogeneous boundary conditions cannot be incorporated in the function space, because this would make the space non-linear. In order to see this, consider the boundary conditions

$$\begin{aligned} \alpha u(0) + \beta u_z(0) &= a, & (\alpha, \beta, a \in \mathbb{R}) \\ \gamma u(1) + \delta u_z(1) &= b, & (\gamma, \delta, b \in \mathbb{R}) \end{aligned}$$

and let u_1 and u_2 be functions in $H^1_m(0,1)$, but now with these boundary conditions. The sum of these solutions is not in the same space, because it does not satisfy the boundary conditions.

The solution is therefore separated into two parts. The first is a solution to the inhomogeneous differential equation with homogeneous boundary conditions, which has been treated above. We will call this the *internally forced* part. The second is a solution to the homogeneous differential equation with inhomogeneous boundary conditions. We will call this the *externally forced* part. The latter will be denoted by $u_{bc1}(a)$ and $u_{bc2}(b)$. $u_{bc1}(a)$ is the solution that satisfies the inhomogeneous boundary condition at $z = 0$ and the homogeneous boundary condition at $z = 1$. $u_{bc2}(b)$ satisfies the inhomogeneous condition at $z = 1$ and the homogeneous condition at $z = 0$. The total solution to the problem reads

$$u = u_{bc1}(a) + u_{bc2}(b) + \mathscr{A}^{-1} f$$

Let us define the linear operator $\mathscr{A}$ as

$$\mathscr{A} = D - \mathscr{N}^0_z \frac{\partial}{\partial z} - \mathscr{N}^0 \frac{\partial^2}{\partial z^2}. \tag{5.11}$$

The momentum equation and its boundary conditions then rewrite to

$$\mathscr{A} \underline{\breve{u}}^0 = -g \left(\underline{\breve{R}}^0_x + \underline{\breve{\zeta}}^0_x \right), \tag{5.12}$$

- $\mathscr{N}^0 \underline{\breve{u}}^0_z(x,R) = \underline{0}$,
- $\begin{cases} \mathscr{N}^0 \underline{\breve{u}}^0_z(x,-H) - \mathscr{S}_f \underline{\breve{u}}^0(x,-H) = \underline{0}, & \text{or} \\ \underline{\breve{u}}^0(x,-H) = \underline{0}, \end{cases}$

where $\mathscr{S}_f$ is a matrix like $\mathscr{N}$ with components $\hat{s}_f$. The solution to the equation reads

$$\underline{\breve{u}}^0 = -g\mathscr{A}^{-1}\left(\underline{\breve{R}}_x^0 + \underline{\breve{\zeta}}_x^0\right). \tag{5.13}$$

The inverse of the abstract linear operator $\mathscr{A}$ can, in some cases, be calculated analytically. However, we will calculate this numerically using the method outlined in Chapter 6

5.2.2 Depth-averaged continuity equation

Next, the depth-averaged continuity equation will be rewritten to a vector form using the complex amplitudes of u and ζ. The resulting expression reads:

$$D\underline{\breve{\zeta}}^0 + \frac{1}{B}\left(\int_{-H}^{R} B\underline{\breve{u}}^0\,dz\right)_x = \underline{0}$$

The velocity components $\breve{u}$ can be eliminated from this equation by using Expression 5.13. This yields

$$D\underline{\breve{\zeta}}^0 - \frac{g}{B}\left(\int_{-H}^{R} B\mathscr{A}^{-1}\,dz\right)_x \underline{\breve{\zeta}}_x^0 - \frac{g}{B}\int_{-H}^{0} B\mathscr{A}^{-1}\,dz\underline{\breve{\zeta}}_{xx}^0 = \frac{g}{B}\left(\int_{-H}^{R} B\mathscr{A}^{-1}\underline{\breve{R}}_x^0\,dz\right)_x \tag{5.14}$$

This can be written in terms of an abstract operator $\mathscr{B}$, which is defined as

$$\mathscr{B} = D - \frac{g}{B}\left(\int_{-H}^{R} B\mathscr{A}^{-1}\,dz\right)_x \frac{\partial}{\partial x} - \frac{g}{B}\int_{-H}^{0} B\mathscr{A}^{-1}\,dz\frac{\partial^2}{\partial x^2}. \tag{5.15}$$

It is useful to note that the right-hand side of Equation (5.14) can be written as $-\mathscr{B}\underline{\breve{R}}^0$, since the reference level only has a sub-tidal component. The depth-averaged continuity equation and its boundary condition then reduce to

$$\mathscr{B}\underline{\breve{\zeta}}^0 = -\mathscr{B}\underline{\breve{R}}^0, \tag{5.16}$$

- $\underline{\breve{\zeta}}^0(0) = \underline{\hat{A}}^0$,
- $-gB(L)\int_{-H}^{R}\mathscr{A}^{-1}(L,z)\,dz\left(\underline{\breve{R}}_x^0(L) + \underline{\breve{\zeta}}_x^0(L)\right) = \underline{\hat{Q}}^0$.

Where $\underline{\hat{A}}^0$ contains the tidal forcing at all frequency components in metres and $\underline{\hat{Q}}^0$ is the river discharge in m^3/s. This vector is only non-zero is its subtidal component.

The solution to this abstract equation can be written as (see Intermezzo 5.2.1 for an explanation of the notation)

$$\underline{\breve{\zeta}}^0 = \breve{\zeta}^0_{\text{tide}} + \breve{\zeta}^0_{\text{river}} - \underline{\breve{R}}^0, \tag{5.17}$$

where the subscript 'tide' denotes the effect from the forcing at the seaward boundary condition (i.e. by $\underline{\hat{A}}^0$) and 'river' denotes the effect from the forcing at the landward boundary condition (i.e. by $\underline{\hat{Q}}^0$). If the reference level is non-zero, then the reference level is deducted from the river-induced water level amplitude.

5.2.3 Continuity equation

Finally, we rewrite the continuity equation and its boundary condition in terms of complex amplitudes as

$$\underline{\breve{w}}_z^0 = -\frac{1}{B}(B\underline{\breve{u}})_x^0, \tag{5.18}$$

- $\underline{\breve{w}}^0(x,-H,t) = -\underline{\breve{u}}^0(x,-H,t)H_x$.

This equation can be used to calculate $\breve{w}^0$ when $\breve{u}^0$ has been calculated from the momentum and depth-averaged continuity equations. The solution reads

$$\underline{\breve{w}}^0(x,z) = -\frac{1}{B}\int_{-H}^{z}(B\underline{\breve{u}})_x^0\,dz - \underline{\breve{u}}^0(x,-H,t)H_x \tag{5.19}$$

5.3 First-order equations

Similar to the leading order, we will write the first-order equations 4.28, 4.31 and 4.33 in terms of complex amplitude vectors. It will be shown that this results in equations that are largely similar to the leading order. The internal forcing terms defined in (4.23)-(4.27) are approximated by a truncated series of Fourier components

$$\begin{aligned}
\varsigma^1(x,z,t) &= Re\left(\sum_{n=-p}^{p}\breve{\zeta}_n^1(x,z)e^{ni\omega t}\right),\\
\eta^1(x,z,t) &= Re\left(\sum_{n=-p}^{p}\breve{\eta}_n^1(x,z)e^{ni\omega t}\right),\\
\psi^1(x,z,t) &= Re\left(\sum_{n=-p}^{p}\breve{\psi}_n^1(x,z)e^{ni\omega t}\right),\\
\gamma^1(x,t) &= Re\left(\sum_{n=-p}^{p}\breve{\gamma}_n^1(x)e^{ni\omega t}\right),\\
\chi^1(x,t) &= Re\left(\sum_{n=-p}^{p}\breve{\chi}_n^1(x)e^{ni\omega t}\right).
\end{aligned}$$

Using these expressions we can proceed by deriving the complex amplitude vector form of the momentum equation.

5.3.1 Momentum equation

The momentum equation in matrix notation is derived similar to the leading-order system. The equation reads

$$D\underline{\breve{u}}^1 - \left(\mathscr{N}^0\underline{\breve{u}}^1\right)_z = -g\left(\underline{R}_x^1 + \underline{\breve{\zeta}}_x^1\right) - \underline{\breve{\eta}}^1 + \underline{\breve{\varsigma}}^1 + \underline{\breve{\psi}}_z^1.$$

We see that this system is highly similar to the leading-order system. The difference is in the forcing components. This is even more clear when we use the abstract notation introduced above. We can again use the same linear operator $\mathscr{A}$ as in 5.11. We then obtain the system:

$$\mathscr{A}\underline{\breve{u}}^1 = -g\left(\underline{R}_x^1 + \underline{\breve{\zeta}}_x^1\right) - \underline{\breve{\eta}}^1 + \underline{\breve{\varsigma}}^1 + \underline{\breve{\psi}}_z^1 \tag{5.20}$$

- $\mathscr{N}^0\underline{\breve{u}}_z^1(x,R) = -\mathscr{N}\underline{\breve{\chi}}^1(x) - \underline{\breve{\psi}}^1(x,R),$
- $\begin{cases}\mathscr{N}^0\underline{\breve{u}}_z^1(x,-H) - \mathscr{S}_f\underline{\breve{u}}^1(x,-H) = -\underline{\breve{\psi}}^1(x,-H), & \text{or}\\ \underline{\breve{u}}^1(x,-H) = \underline{0},\end{cases}$

The solution then has the abstract form

$$\begin{aligned}
\underline{\breve{u}}^1 &= -g\mathscr{A}^{-1}\left(\underline{R}_x^1 + \underline{\breve{\zeta}}_x^1\right) - \mathscr{A}^{-1}\underline{\breve{\eta}}^1 + \mathscr{A}^{-1}\underline{\breve{\varsigma}}^1 + \mathscr{A}^{-1}\underline{\breve{\psi}}_z^1 + \underline{\breve{u}}^1_{\text{no-stress}} + \underline{\breve{u}}^1_{\text{mixing}},\\
&= -g\mathscr{A}^{-1}\left(\underline{R}_x^1 + \underline{\breve{\zeta}}_x^1\right) + \underline{\breve{u}}_{\text{other forcings}},
\end{aligned}$$

where the last equation is simply a convenient short-hand notation of the complete first equation.

5.3.2 Depth-averaged continuity and continuity equations

The first-order depth-averaged continuity equation becomes

$$D\underline{\breve{\zeta}}^1 + \frac{1}{B}\left(\int_{-H}^{R} B\underline{\breve{u}}^1\,dz\right)_x = -\frac{1}{b}\left(B\underline{\breve{\gamma}}^1\right)_x.$$

Substituting the expression for u^1 we find

$$D\underline{\breve{\zeta}}^1 - \frac{g}{B}\left(\int_{-H}^{R} B\mathscr{A}^{-1}\underline{\breve{\zeta}}_x^1\,dz\right)_x = -\frac{1}{b}\left(B\underline{\breve{\gamma}}^1\right)_x + \frac{g}{B}\left(\int_{-H}^{R} B\mathscr{A}^{-1}\underline{\breve{R}}_x^1\,dz\right)_x - \frac{1}{B}\left(\int_{-H}^{R} B\underline{\breve{u}}_{\text{other forcings}}\,dz\right)_x.$$

The depth-averaged continuity equation and its boundary condition then reduce to

$$\mathscr{B}\underline{\breve{\zeta}}^1 = -\frac{1}{B}\left(B\underline{\breve{\gamma}}^1\right)_x - \mathscr{B}\underline{\breve{R}}^1 - \frac{1}{B}\left(\int_{-H}^{R} B\underline{\breve{u}}_{\text{other forcings}}\,dz\right)_x, \tag{5.21}$$

- $\underline{\breve{\zeta}}^1(0) = \underline{\hat{A}}^1,$
- $-gB(L)\int_{-H}^{R}\mathscr{A}^{-1}(L,z)\,dz\left(\underline{Rb}_x^1(L) + \underline{\breve{\zeta}}_x^1(L)\right) = \underline{\hat{Q}}^1.$

The solution to this abstract equation can be written as

$$\underline{\breve{\zeta}}^1 = -\mathscr{B}^{-1}\frac{1}{B}\left(B\underline{\breve{\gamma}}^1\right)_x + \underline{\breve{\zeta}}^1_{\text{tide}} + \underline{\breve{\zeta}}^1_{\text{river}} - \underline{\breve{R}}^1 - \mathscr{B}^{-1}\frac{1}{B}\left(\int_{-H}^{R} B\underline{\breve{u}}_{\text{other forcings}}\,dz\right)_x. \tag{5.22}$$

The first-order water level thus consists of the externally forcing tide and river flow, the tidal return flow originating from the depth-averaged continuity equation and several terms originating from the momentum balance. The reference level is again subtracted from the river-induced water level.

The first-order continuity equation is identical in u^1 and w^1 to the leading-order continuity in u^0 and w^0. The solution therefore trivially follows from the solution of the leading-order continuity equation.

5.4 Generalisation

We have seen that the leading-order and first-order systems have a highly similar structure. We will exploit this in the numerical implementation by implementing a single solver for each equation (momentum, depth-averaged continuity), which can be used for both orders. We will not define a generalisation of the continuity equation as this equation has already been solved for explicitly above.

The momentum equation has the general form:

$$D\underline{\breve{u}} - \left(\mathscr{N}^0\underline{\breve{u}}_z\right)_z = -g\underline{\breve{\zeta}}_x + \underline{f},$$

- $\mathscr{N}^0\underline{\breve{u}}_z(x,R) = \underline{f}_{\text{surface}},$
- $\begin{cases}\mathscr{N}\underline{\breve{u}}_z(x,-H) - \mathscr{S}_f\underline{\breve{u}}(x,-H) = \underline{f}_{\text{bed}}, & \text{or}\\ \underline{\breve{u}}(x,-H) = \underline{f}_{\text{bed}},\end{cases}$

This equation takes input parameters $\mathscr{N}$, $\mathscr{N}_z$, $\mathscr{F}$, $\underline{f}_surface$ and $\mathscr{S}_f$.

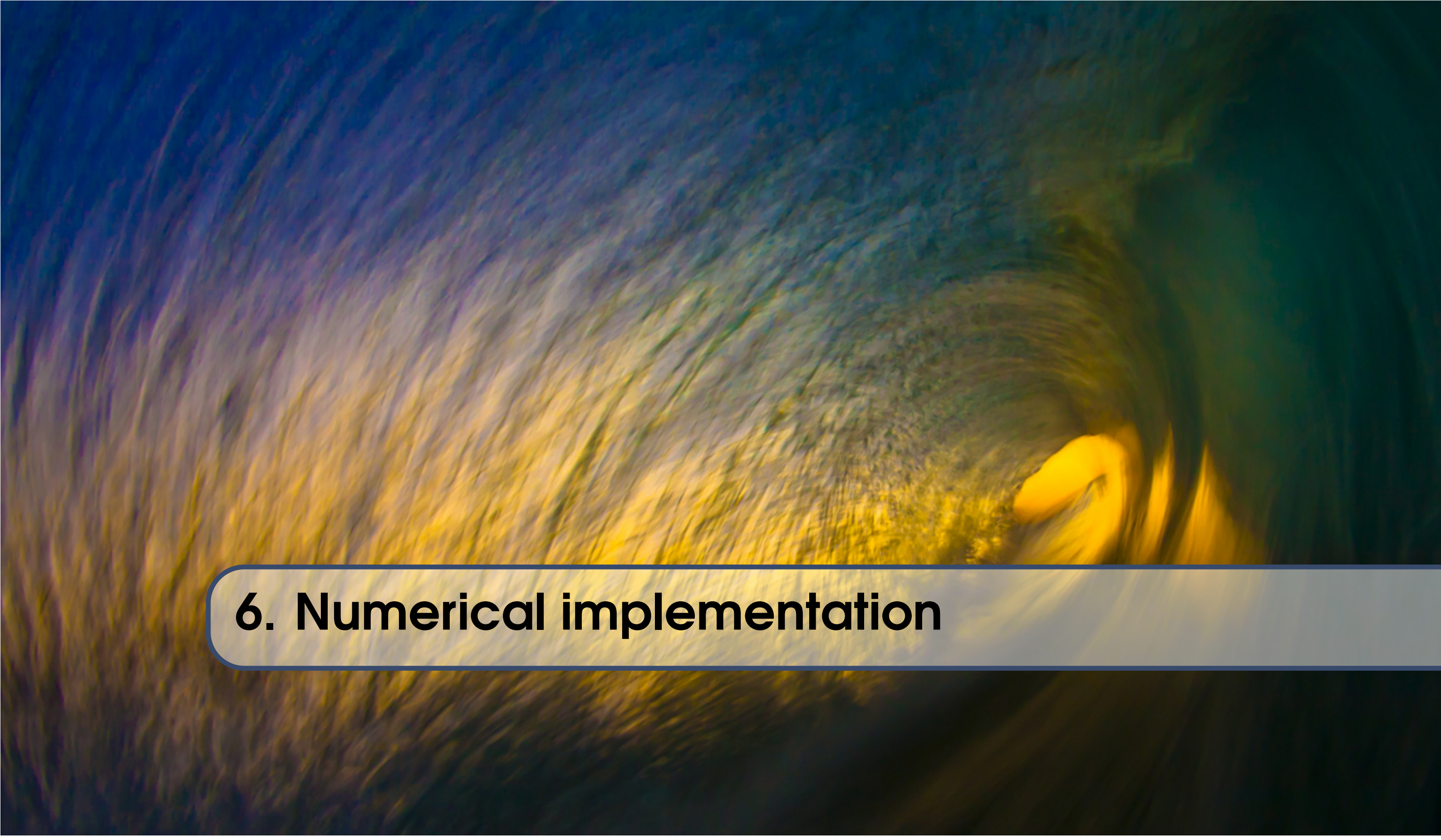
The depth-averaged continuity equation takes the form:

$$D\underline{\breve{\zeta}} + \frac{1}{B}\left(\mathscr{M}\underline{\breve{\zeta}}_x + \mathscr{F}\right)_x = \underline{0},$$

- $\underline{\breve{\zeta}}(0) = \underline{f}_{\text{open}},$
- $\mathscr{M}\underline{\breve{\zeta}}_x(L) = \underline{f}_{\text{closed}}$

The numerical implementations of these two equations is presented in the next chapter.

The general form of the continuity equation is given by (5.19)



6. Numerical implementation

This chapter describes the numerical implementation of the two general matrix equations that were described in Section 5.4. Since these general equations hold for all orders, the numerical implementation is the same for each order. For the momentum equation we will adopt a general discretisation method for $\breve{u}_z$ that guarantees the conservation of momentum. The solution u is then obtained by a second-order numerical integration of $\breve{u}_z$. The depth-averaged continuity equation is implemented using a strictly mass conservative scheme. One of the consequences is that the baroclinically induced velocity has a depth-averaged value that is strictly zero, regardless of the grid resolution. This property is especially useful when computing the transport of constituents in additional modules. This is because there is no net depth-averaged flow created by numerical errors.

The implementation of the continuity equation is not discussed in a separate section, since no discretisation is required. The solution of the continuity equation uses a second-order central numerical differentiation scheme, which reduces to a first-order upwind scheme at the boundaries. It also uses a first-order numerical integration scheme based on the trapezoidal rule.

The equations will be discretised on a collocated non-equidistant grid following the iFlow standard axis format. This format for a grid axis is sketched in Figure 6.1.

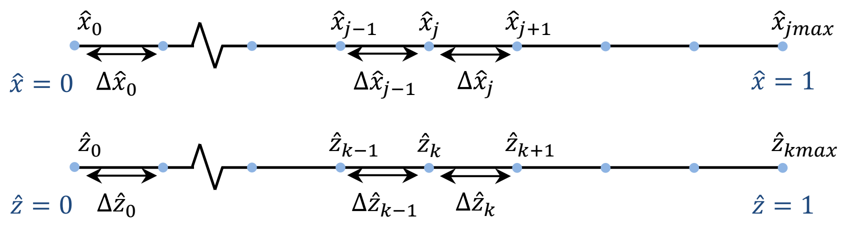


Figure 6.1: Definition of a standard dimensionless grid axis for $\hat{x}$ and $\hat{z}$.

6.1 Implementation of the generalised momentum equation

We choose for a numerical implementation of the momentum equations that starts from the conservative form of the differential equation. This form is not solved for the velocity u, but instead for u_z. The momentum equations are repeated here for convenience:

$$D\underline{\breve{u}} - \left(\mathcal{N}^0 \underline{\breve{u}}_z\right)_z = -g\underline{\breve{\zeta}}_x + \underline{f}, \tag{6.1}$$

- $$\mathcal{N}^0 \underline{\breve{u}}_z(x,R) = \underline{f}_{\text{surface}}, \tag{6.2}$$
- $$\begin{cases} \mathcal{N}\underline{\breve{u}}_z(x,-H) - \mathcal{S}_f \underline{\breve{u}}(x,-H) = \underline{f}_{\text{bed}}, & \text{or} \\ \underline{\breve{u}}(x,-H) = \underline{f}_{\text{bed}}, \end{cases}$$

For the following we define a numerical vector as $\underline{\underline{\breve{u}}}$ with elements $\breve{u}_{n,k}$ or $\underline{\breve{u}}_k$ at frequency component n and grid point k according to

$$\begin{aligned}\underline{\underline{\breve{u}}} &= [\breve{u}_{-p,0}, \dots, \breve{u}_{p,0}, \breve{u}_{-p,1}, \dots, \breve{u}_{p,1}, \dots, \breve{u}_{-p,\text{kmax}}, \dots, \breve{u}_{p,\text{kmax}}]^T \\ &= [\underline{\breve{u}}_0^T, \dots \underline{\breve{u}}_1^T, \dots, \underline{\breve{u}}_{\text{kmax}}^T]^T.\end{aligned} \tag{6.3}$$

6.1.1 Discretisation

Even though we will solve the equation on a collocated grid, during the derivation we will consider this equation on a staggered grid where u_z is given on the same points as the standard grid (Figure 6.1) and where u is given in the centre between two grid points. Later, we will convert u from the centre points back to the standard grid points. On the staggered grid with indices $k = 0, \dots,$kmax let us define $\underline{\breve{\phi}} = \breve{u}_z$ as

$$\underline{\breve{\phi}}_k = \frac{1}{\Delta z_{k-1/2}} \left(\underline{\breve{u}}_{k+1/2} - \underline{\breve{u}}_{k-1/2}\right),$$

where $\Delta z_{k-1/2} = \frac{1}{2}(\Delta z_k + \Delta z_{k-1})$. Equation (6.1) can then be discretised at point $z_{k+1/2}$ using a central discretisation. This yields

$$D\underline{\breve{u}}_{k+1/2} - \frac{1}{\Delta z_k}\left(\mathcal{N}^0_{k+1}\underline{\breve{\phi}}_{k+1} - \mathcal{N}^0_k \underline{\breve{\phi}}_k\right) = -g\underline{\breve{\zeta}}_x + \underline{f}_{k+1/2}. \tag{6.4}$$

We now subtract the equations in two consecutive cell centres

$$D\left(\underline{\breve{u}}_{k+1/2} - \underline{\breve{u}}_{k-1/2}\right) - \frac{1}{\Delta z_k}\left(\mathcal{N}^0_{k+1}\underline{\breve{\phi}}_{k+1} - \mathcal{N}^0_k \underline{\breve{\phi}}_k\right) + \frac{1}{\Delta z_{k-1}}\left(\mathcal{N}^0_k \underline{\breve{\phi}}_k - \mathcal{N}^0_{k-1}\underline{\breve{\phi}}_{k-1}\right) = \underline{f}_{k+1/2} - \underline{f}_{k-1/2}.$$

This equation rewrites to

$$D\underline{\breve{\phi}}_k \Delta z_{k-1/2} - \frac{1}{\Delta z_k}\left(\mathcal{N}^0_{k+1}\underline{\breve{\phi}}_{k+1} - \mathcal{N}^0_k \underline{\breve{\phi}}_k\right) + \frac{1}{\Delta z_{k-1}}\left(\mathcal{N}^0_k \underline{\breve{\phi}}_k - \mathcal{N}^0_{k-1}\underline{\breve{\phi}}_{k-1}\right) = \underline{f}_{k+1/2} - \underline{f}_{k-1/2}. \tag{6.5}$$

This expression requires the forcing $\underline{f}$ at the grid centres. This forcing is determined by linear interpolation using the adjacent grid points.

The boundary condition at the surface is easily discretised; Surface boundary condition

$$\mathcal{N}^0_0 \underline{\breve{\phi}}_0 = \underline{f}_{\text{surface}}.$$

Indeed, one of the reasons for choosing this method of implementation is because it guarantees that the velocity gradient at the surface is exact.

The partial slip condition at the bed is solved for by adapting the discrited equation (6.4) Partial slip in the cell centre closest to the bed. This discretised equation reads

$$D\underline{\breve{u}}_{kmax-1/2} - \frac{1}{\Delta z_{kmax-1}}\left(\mathscr{N}^0_{kmax}\underline{\breve{\phi}}_{kmax} - \mathscr{N}^0_{kmax-1}\underline{\breve{\phi}}_{kmax-1}\right) = -g\underline{\breve{\zeta}}_x + \underline{f}_{kmax-1/2}. \tag{6.6}$$

Next we use the boundary condition $\mathscr{N}^0_{kmax}\underline{\breve{\phi}}_{kmax} = \mathscr{S}_f\underline{\breve{u}}_{kmax} + \underline{f}_{\mathrm{bed}}$ to obtain an expression for $\underline{\breve{u}}$ at the bed

$$\underline{\breve{u}}_{kmax} = \mathscr{S}_f^{-1}\mathscr{N}^0\underline{\breve{\phi}}_{kmax} - \mathscr{S}_f^{-1}\underline{f}_{\mathrm{bed}}. \tag{6.7}$$

This can be converted to an approximation of $\underline{\breve{u}}_{kmax-1/2}$ by a first-order Taylor approximation

$$\underline{\breve{u}}_{kmax-1/2} = \underline{\breve{\phi}}_{kmax}\left(\mathscr{S}_f^{-1}\mathscr{N}^0 - \tfrac{1}{2}\Delta z_{kmax-1}\right) - \mathscr{S}_f^{-1}\underline{f}_{\mathrm{bed}}.$$

This condition is substituted to the left-hand side of (6.6). The thus obtained expression is left-multiplied by $\mathscr{S}_f$ and reads

$$D\left(\mathscr{N}^0 - \tfrac{1}{2}\Delta z_{kmax-1}\mathscr{S}_f\right)\underline{\breve{\phi}}_{kmax} - \frac{1}{\Delta z_{kmax-1}}\left(\mathscr{S}_f\mathscr{N}^0_{kmax}\underline{\phi}_{kmax} - \mathscr{S}_f\mathscr{N}^0_{kmax-1}\underline{\phi}_{kmax-1}\right)$$
$$= -g\mathscr{S}_f\underline{\breve{\zeta}}_x + \mathscr{S}_f\underline{f}_{kmax-1/2} + D\underline{f}_{\mathrm{bed}}.$$

After solving the above equation for ϕ, the integration to $\underline{\hat{u}}$ results in an integration constant equal to the velocity at the bed. The velocity at the bed follows from the bed boundary condition (see (6.7)) and is repeated here

$$\underline{\hat{u}}_{kmax} = \mathscr{S}_f^{-1}\mathscr{N}\underline{\phi}_{kmax} - \mathscr{S}_f^{-1}\underline{f}_{\mathrm{bed}}.$$

No-slip

Alternatively the no-slip boundary condition follows from combining (6.6) with a first-order Taylor approximation of the no-slip condition

$$\underline{\breve{u}}_{kmax-1/2} = \tfrac{1}{2}\Delta z_{kmax-1}\underline{\breve{\phi}}_{kmax}.$$

We then obtain

$$\tfrac{1}{2}\Delta z_{kmax-1}D\underline{\breve{\phi}}_{kmax}. - \frac{1}{\Delta z_{kmax-1}}\left(\mathscr{N}^0_{kmax}\underline{\breve{\phi}}_{kmax} - \mathscr{N}^0_{kmax-1}\underline{\breve{\phi}}_{kmax-1}\right) = -g\underline{\breve{\zeta}}_x + \underline{f}_{kmax-1/2}. \tag{6.8}$$

The bed velocity required in the integration follows directly from the no slip condition as $\underline{\breve{u}}_{\mathrm{kmax}} = 0$.

6.1.2 Numerical implementation

The discretised equation (6.5) can be abstractly written as

$$\mathbb{A}(x)\underline{\underline{\breve{\phi}}}(x) = -gI_b\underline{\underline{\hat{\zeta}}}_x(x) + \underline{\underline{\tilde{f}}}(x),$$

where the double underlining denotes a numerical vector, see (6.3). This equation solves for the vertical dimension and is evaluated at each horizontal grid point. The matrix $\mathbb{A}$ is a band matrix. The matrices $\mathscr{A}$ and I_b are defined below in terms of sub-matrices of dimension $(2p+1)\times(2p+1)$

$$\mathscr{A} = \begin{bmatrix} I & \emptyset & & & \\ A & B & C & \emptyset & \\ & \ddots & \ddots & \ddots & \\ & \emptyset & A & B & C \\ & & & \tilde{A} & \tilde{B} \end{bmatrix}, \quad I_b = \begin{bmatrix} \emptyset \\ \vdots \\ \vdots \\ \emptyset \\ I \end{bmatrix}. \tag{6.9}$$

where the sub-matrices $A, B, C, \tilde{A}$ and $\tilde{B}$ are formed from the coefficients in front of $\underline{\breve{\phi}}_{k-1}$, $\underline{\breve{\phi}}_{k}$ and $\underline{\breve{\phi}}_{k+1}$ in (6.5). The actual bandwidth of the matrix $\mathbb{A}$ is $4p+3$, because A, B and C themselves are band matrices. This is the most optimal matrix structure that can be obtained in terms of computational time to solve the system.

Since $\underline{\underline{\breve{\zeta}}}_x$ is still unknown, the solution $\underline{\underline{\breve{\phi}}}$ follows from separately solving $\mathbb{A}\Phi = I_b$ and $\mathbb{A}\underline{\underline{\breve{\phi}}}_f = \underline{\underline{f}}$, so that

$$\underline{\underline{\breve{\phi}}} = -g\Phi\underline{\underline{\breve{\zeta}}}_x + \underline{\underline{\breve{\phi}}}_f.$$

After $\underline{\underline{\breve{\zeta}}}_x$ is computed from the depth-averaged continuity equation, the velocity follows from the vertical velocity gradient using a second-order numerical integration routine based on the Simpson rule, see the NiFTy package.

6.2 Implementation of the generalised depth-averaged continuity equation

We will adopt an implementation of the generalised depth-averaged continuity equation that guarantees the net conservation of mass. We will first derive the criterion that the equation has to satisfy and then derive the mass conservative scheme. The generalised equations are repeated here for convenience:

$$D\underline{\breve{\zeta}} + \frac{1}{B}\left(\mathscr{M}\underline{\breve{\zeta}}_x + \mathscr{F}\right)_x = \underline{0}, \tag{6.10}$$

- $\underline{\breve{\zeta}}(0) = \underline{f}_{\text{open}}$,
- $\mathscr{M}\underline{\breve{\zeta}}_x(L) = \underline{f}_{\text{closed}}$.

The net conservation of mass concerns the residual flow. The residual component is therefore taken from the equation and for illustration purposes we assume that the boundary conditions are homogeneous:

$$\left\{\frac{1}{B}\left(\mathscr{M}\underline{\hat{\zeta}}_x + \mathscr{F}\right)_x\right\}_0 = 0.$$

This equation can be integrated over x to arrive at

$$\left\{\mathscr{M}\underline{\hat{\zeta}}_x + \mathscr{F}\right\}_0 = 0,$$

where the right-hand side is zero, because it is assumed that the boundary conditions are homogeneous. Non-homogeneous boundary conditions would add a net contribution on the right-hand side. The above equation shows that it is required to calculated $\hat{\zeta}_x$ directly, as this is the only way to guarantee that this equation is always satisfies in the numerical calculation.

6.2.1 Discretisation

In order to find a numerical scheme to solve for ζ_x, we will consider a staggered grid with the center-points defined in the middle of the grid points of our standard grid (Figure 6.1). The final scheme that we will derive will only use the points on the collocated standard grid, so that the staggered grid is only used for the derivation. On the staggered grid let us define $\xi = \zeta_x$ as

$$\breve{\xi}_i = \frac{1}{\Delta x_{i-1/2}}\left(\hat{\zeta}_{i+1/2} - \hat{\zeta}_{i-1/2}\right),$$

where $\Delta x_{i-1/2} = \frac{1}{2}(\Delta x_i + \Delta x_{i-1})$. We can then discretise Equation 6.10 at the point $j+1/2$ using a central discretisation. The result reads

$$D\underline{\zeta}_{j+1/2} + \frac{1}{B_{j+1/2}\Delta x_j}\left(\mathscr{M}_{j+1}\underline{\xi}_{j+1} + \mathscr{F}_{j+1} - \mathscr{M}_j\underline{\xi}_j - \mathscr{F}_j\right) = \underline{0}. \tag{6.11}$$

An equation for $\underline{\breve{\xi}}$ is obtained by subtracting two equations at neighbouring points;

$$D\left(\underline{\breve{\zeta}}_{j+1/2} - \underline{\breve{\zeta}}_{j-1/2}\right) + \frac{1}{B_{j+1/2}\Delta x_j}\left(\mathscr{M}_{j+1}\underline{\breve{\xi}}_{j+1} - \mathscr{M}_j\underline{\breve{\xi}}_j + \mathscr{F}_{j+1} - \mathscr{F}_j\right)$$
$$- \frac{1}{B_{j-1/2}\Delta x_{j-1}}\left(\mathscr{M}_j\underline{\breve{\xi}}_j - \mathscr{M}_{j-1}\underline{\breve{\xi}}_{j-1} + \mathscr{F}_j - \mathscr{F}_{j-1}\right) = \underline{0},$$

which rewrites to

$$D\underline{\breve{\xi}}_j\Delta x_{j-1/2} + \frac{1}{B_{j+1/2}\Delta x_j}\left(\mathscr{M}_{j+1}\underline{\breve{\xi}}_{j+1} - \mathscr{M}_j\underline{\breve{\xi}}_j + \mathscr{F}_{j+1} - \mathscr{F}_j\right) \tag{6.12}$$
$$- \frac{1}{B_{j-1/2}\Delta x_{j-1}}\left(\mathscr{M}_j\underline{\breve{\xi}}_j - \mathscr{M}_{j-1}\underline{\breve{\xi}}_{j-1} + \mathscr{F}_j - \mathscr{F}_{j-1}\right) = \underline{0}. \tag{6.13}$$

This expression only uses B on the staggered grid points. The values for B at these points will be either determined analytically or by linear interpolation of numerical data, depending on the way B is defined.

Boundary conditions

The boundary condition on the open boundary can be discretised by using that $\underline{\breve{\xi}}_0 = \frac{1}{\frac{1}{2}\Delta x_0}\left(\underline{\breve{\zeta}}_{1/2} - \underline{\hat{A}}\right)$. This expression can be substituted into equation 6.11 to obtain a boundary condition for $\underline{\breve{\xi}}$:

$$D\left(\tfrac{1}{2}\Delta x_0\underline{\breve{\xi}}_0 + \underline{\hat{A}}\right) + \frac{1}{B_{1/2}\Delta x_0}\left(\mathscr{M}_1\underline{\breve{\xi}}_1 - \mathscr{M}_0\underline{\breve{\xi}}_0 + \mathscr{F}_1 - \mathscr{F}_0\right) = \underline{0}$$

The boundary condition at the landward boundary follows directly from the equation and reads

$$\mathscr{M}_{\text{jmax}}\underline{\breve{\xi}}_{\text{jmax}} = \underline{f}_{\text{closed}}.$$

6.2.2 Numerical implementation

The discretised depth-averaged continuity equation (6.13) can be written as

$$\mathbb{B}\underline{\underline{\breve{\xi}}} = \underline{\underline{f}}_{\text{open}} + \underline{\underline{f}}_{\text{closed}} + \tilde{\mathscr{F}},$$

where the right-hand side term $\tilde{\mathscr{F}}$ is a function containing the forcing term $\mathscr{F}$ and the matrix $\mathbb{B}$ and right-hand side vectors can be written as

$$\mathscr{B} = \begin{bmatrix} \tilde{E} & \tilde{F} & & & \\ E & F & G & \emptyset & \\ & \ddots & \ddots & \ddots & \\ & \emptyset & E & F & G \\ & & & \emptyset & I \end{bmatrix}, \quad \underline{\underline{f}}_{\text{open}} = \begin{bmatrix} \underline{f}_{\text{open}} \\ \underline{0} \\ \vdots \\ \underline{0} \\ \underline{0} \end{bmatrix}, \quad \underline{\underline{f}}_{\text{open}} = \begin{bmatrix} \underline{0} \\ \underline{0} \\ \vdots \\ \underline{0} \\ \underline{f}_{\text{closed}} \end{bmatrix},$$

with $2p+1$ square sub-matrices E, F and G following from the coefficients in the discretised equations. The water level $\underline{\underline{\zeta}}$ follows from the computed water level gradient by a second-order numerical integration routine based on the Simpson rule, see the NiFTy package.

7. Higher order model

We have seen that the leading-order and first-order momentum and depth-averaged continuity equations can be expressed in abstract notation as $\mathscr{A}u = f_1$ and $\mathscr{B}u = f_2$, where $\mathscr{A}$ and $\mathscr{B}$ are some linear operators and f_1 and f_2 represent the forcing terms. Similar forms can be derived for the second-order and higher-order systems. In this chapter we will derive these systems for a model up to any order.

7.1 Derivation of the equations

We will start again from the equations of motion (4.1)-(4.9) with the scaling of Section 4.1. Before making the ordering we will develop a series expansion around $z = R$ in the boundary conditions and depth-integrated quantities. Using Taylor series, we find the following expansions

$$A_v(x,R+\zeta,t)u_z(x,R+\zeta,t) = \sum_{n=0}^{\infty}\frac{1}{n!}\left(A_v(x,R,t)u_z(x,R,t)\right)^{(n)}\zeta^n,$$

$$\int_{-H}^{R+\zeta} u(x,z,t)\,dz = \int_{-H}^{R} u(x,z,t)\,dz + \sum_{n=1}^{\infty}\frac{1}{n!}u^{(n-1)}(x,R,t)\zeta^n,$$

$$\int_{z}^{R+\zeta} \rho_x(x,z,t)\,dz = \int_{z}^{R} \rho_x(x,z,t)\,dz + \sum_{n=1}^{\infty}\frac{1}{n!}\rho_x^{(n-1)}(x,R,t)\zeta^n,$$

where $[\cdot]^{(n)}$ denotes the n^{th}-order derivative in z-direction. Substituting this, our system of equations then reads

$$u_t - (A_V u_z)_z = -g\zeta_x - \eta - \varsigma - \varsigma_\delta,$$

- $A_V(x,R,t)u_z(x,R,t) = -\chi,$
- $A_V(x,-H,t)u_z(x,-H,t) - s_f u(x,-H,t) = 0,$ or,
- $u(x,-H,t) = 0.$

$$\zeta_t + \frac{1}{B}\left(B\int_{-H}^{0} u\,dz\right)_x = -\frac{1}{B}(B\gamma)_x,$$

- $\zeta(0) = \zeta_0,$
- $\displaystyle\int_{-H}^{R} u(L)\,dz = q_{\text{riv}} - \gamma(L,t)$

$$w_z + \frac{1}{B}(Bu_x) = 0,$$

- $w(x,-H,t) = -u(x,-H,t)H_x.$

Here we use the short-hand notation introduced in Expressions (4.23)-(4.27), with the addition of ς_δ defined as

$$\varsigma_\delta = \frac{g}{\rho_0}\sum_{n=1}^{\infty}\frac{1}{n!}\rho_x^{(n-1)}(x,R,t)\zeta^n, \qquad \text{(density drift)} \tag{7.1}$$

The system is scaled and ordered using the same procedure as in Sections 4.1 and 4.2. For clarity, we will adopt the following notation

$[\cdot]^n$	for n^{th} powers,
$[\cdot]^{(n)}$	for n^{th}-order derivatives with respect to z,
$[\cdot]^{<n>}$	for n^{th} order in ε.

For the ordering, we will use the formal expansions of Expressions (4.11)-(4.14), which now are written as e.g. $u(x,z,t) = \sum_{n=0}^{\infty} u^{<n>}(x,z,t)$. The ordering leads to a system of equations for order $n = 0,1,\ldots$ of the form

$$u_t^{<n>} - \left(A_V{}^{<0>}u_z^{<n>}\right)_z = -g\zeta_x^{<n>} - \eta^{<n>} - \varsigma^{<n>} - \varsigma_\delta^{<n>} + \psi_z^{<n>},$$

- $A_V{}^{<0>}(x,R,t)u_z^{<n>}(x,R,t) = -\chi^{<n>} - \psi^{<n>}(x,R,t) - \psi_\chi^{<n>},$
- $A_V{}^{<0>}(x,-H,t)u_z^{<n>}(x,-H,t) - s_f u^{<n>}(x,-H,t) = -\psi^{<n>}(x,-H,t),$ or,
- $u^{<n>}(x,-H,t) = 0.$

$$\zeta_t + \frac{1}{B}\left(B\int_{-H}^{R} u^{<n>}\,dz\right)_x = -\frac{1}{B}\left(B\gamma^{<n>}\right)_x,$$

- $\zeta^{<n>}(0) = \zeta_0^{<n>},$
- $\displaystyle\int_{-H}^{R} u^{<n>}(L)\,dz = q_{\text{riv}}^{<n>} - \gamma^{<n>}(L,t),$

$$w_z^{<n>} + \frac{1}{B}\left(Bu_x^{<n>}\right) = 0,$$

- $w^{<n>}(x,-H,t) = -u^{<n>}(x,-H,t)H_x.$

Here we find another new term $\psi_c hi$, which indicates the interaction between the higher-order eddy viscosity and depth variations, or *mixing-no-stress interaction*. The non-linear terms η, ς_δ, χ, ψ_χ and γ and the linear terms ς and ψ are all zero at the leading order. The density drift ς_δ and mixing-no-stress interaction ψ_χ are also zero at first order. At order n ($n \geq 1$) these forcing terms depend only on the flow velocity and water level up to order $n-1$. The system of equation is thus the same at each order, except for the forcing. The forcing terms at each order are expanded below.

$$\eta^{<n>} = \sum_{m=0}^{n-1} u^{<m>} u_x^{<n-m-1>} + w^{<m>} u_z^{<n-m-1>}, \qquad \text{(advection)}$$

$$\varsigma^{<n>} = \frac{g}{\rho_0} \int_z^0 \rho_x^{<n>}, \qquad \text{(baroclinic pressure)}$$

$$\varsigma_\delta^{<n>} = \frac{g}{\rho_0} \sum_{m=1}^{n-1} \sum_{k=0}^{n-1-m} \frac{1}{m!} \left(\rho_x^{<k>}(x,0,t)\right)^{(m-1)} \zeta^{<l_1>} \cdots \zeta^{<l_m>},$$

$$\forall l_1, \ldots, l_m \text{ s.t. } \sum_{r=1}^{m} l_r = n-1-m-k, \qquad \text{(density drift)}$$

$$\chi^{<n>} = \sum_{m=1}^{n} \sum_{k=0}^{n-m} \frac{1}{m!} \left(A_v(x,0,t) u_z^{<k>}(x,0,t)\right)^{(m)} \zeta^{<l_1>} \cdots \zeta^{<l_m>},$$

$$\forall l_1, \ldots, l_m \text{ s.t. } \sum_{r=1}^{m} l_r = n-m-k, \qquad \text{(no-stress term)}$$

$$\gamma^{<n>} = \sum_{m=1}^{n} \sum_{k=0}^{n-m} \frac{1}{m!} \left(u^{<k>}(x,0,t)\right)^{(m-1)} \zeta^{<l_1>} \cdots \zeta^{<l_m>},$$

$$\forall l_1, \ldots, l_m \text{ s.t. } \sum_{r=1}^{m} l_r = n-m-k, \qquad \text{(tidal return flow)}$$

$$\psi^{<n>} = \sum_{m=1}^{n} A_v{}^{<m>}(x,z,t) u_z^{<n-m>}(x,z,t). \qquad \text{(mixing)}$$

$$\psi_\chi^{<n>} = \sum_{m=1}^{n} \sum_{k=0}^{n-m} \sum_{i=1}^{n-m-k} \frac{1}{m!} \left(A_v{}^{<i>}(x,0,t) u_z^{<k>}(x,0,t)\right)^{(m)} \zeta^{<l_1>} \cdots \zeta^{<l_m>},$$

$$\forall l_1, \ldots, l_m \text{ s.t. } \sum_{r=1}^{m} l_r = n-m-k-i, \qquad \text{(mixing-no-stress interaction)}$$

Apart from these *internal* forcing mechanisms, we allow for external forcing by the tide and river. These external forcing mechanisms are only allowed on leading order and first order. This is because higher-order external forcing mechanisms have no added value to the interpretation of the model results.

7.2 Computation of required derivatives

Since the higher-order equations take the same form as the leading-order and first-order systems, they can be solved using the same methods for harmonic analysis and numerical implementation, see Chapters 5 and 6. The difference between the system at each order is in the forcing terms. A particular point of attention are the no-stress, tidal return flow and density drift terms, which need higher-order derivatives of the velocity and density at the surface. These derivatives cannot simply be computed using straight-forward numerical differentiation as one quickly looses accuracy, see also Appendix A. Therefore the derivatives are computed using an alternative method presented in this section.

7.2.1 Required derivatives for higher order hydrodynamics

Before deriving an alternative method for calculating derivatives, we will look closer at which derivatives are required at which order. The required derivatives for the n^{th}-order equation are listed below

Term	Derivative	Remarks
Advection	$u_z^{<k>}$	$(k \le n)$.
Density drift	$\left(-A_v \rho_z^{<k>}\right)^{(m-1)}(x,R,t)$	$(m \le n-1, k \le n-m-1)$.
No-stress	$\left(-A_v u_z^{<k>}\right)^{(m)}(x,R,t)$	$(m \le n, k \le n-m)$.
Tidal return flow	$\left(u^{<k>}\right)^{(m-1)}(x,R0,t)$	$(m \le n, k \le n-m)$.

The advection term thus requires the first derivatives to be available always and everywhere. The no-stress and tidal return flow terms require very specific higher-order derivatives at the surface. Especially for the first-order and second-order equations these specific derivatives can be calculated by using the momentum equation and the knowledge of the solutions. We will therefore look specifically at the first and second order before considering the general case.

First-order system

The first-order no-stress term only requires $\left(A_v u_z^0\right)_z(x,R,t)$. This term can be calculated by using the leading order momentum equation, according to

$$\left(A_v u_z^0\right)_z(x,R,t) = ni\omega u^0(x,R,t) + g\zeta_x^0(x,t).$$

This does not require taking any derivatives and thus has the same accuracy as the leading-order solution. The tidal return flow term only requires $u^0(x,R,t)$ and thus involves no vertical derivatives.

Second-order system

The second-order no-stress term requires the following terms: $\left(A_v u_z^0\right)_z(x,R,t)$, $\left(A_v u_z^1\right)_z(x,R,t)$, $\left(A_v u_z^0\right)_{zz}(x,R,t)$. The first of these terms has already been calculated in the first-order system. The second term can be calculated from the first-order momentum equation according to

$$\begin{aligned}\left(A_v u_z^1\right)_z(x,R,t) &= ni\omega u^1(x,R,t) + (u^0 u_x^0)(x,R,t) + (w^0 \underbrace{u_z^0}_{=0})(x,R,t) + g\zeta_x^1(x,t) + \underbrace{\frac{g}{\rho_0}\int_R^R \rho_x^0}_{=0}, \\ &= ni\omega u^1(x,R,t) + (u^0 u_x^0)(x,R,t) + g\zeta_x^1(x,t).\end{aligned}$$

This also does not require taking any vertical derivatives. The third term in the no-stress contribution is determined from taking the derivative of the leading-order momentum equation. This yields

$$\begin{aligned}\left(A_v u_z^0\right)_{zz}(x,0,t) &= ni\omega \underbrace{u_z^0(x,0,t)}_{=0} + \underbrace{(g\zeta_x(x,t))_z}_{=0}, \\ &= 0.\end{aligned}$$

The tidal return flow requires $u^0(x,0,t)$, $u^1(x,0,t)$, $u_z^0(x,0,t)$. The latter term is zero, so that there are no vertical derivatives involved in the second-order tidal return flow.

It thus follows that the first-order and second-order systems only require the first derivative of the velocity for the advection term, but otherwise do not require any vertical derivatives. The first-order derivative is solved for directly in the numerical solution procedure (see Chapter 6), so that its accuracy is guaranteed.

Third and higher orders

The new terms that appear in the third-order no-stress terms are $\left(A_\nu u_z^2\right)_z(x,0,t)$, $\left(A_\nu u_z^1\right)_{zz}(x,0,t)$ and $\left(A_\nu u_z^0\right)_{zzz}(x,0,t)$. The first term could be reconstructed from the second-order momentum equation in a similar way as for the first-order momentum equation in the second-order computation. As the second-order momentum equation contains many forcing terms, this is done by saving the forcing terms while calculating the second order and using this saved forcing for the reconstruction.

The second term is calculated by differentiating the first-order momentum balance. This requires calculating u_z^0z and u_z^1 at the surface. The third term can be obtained by differentiating the leading-order momentum balance twice. This yields

$$\left(A_\nu u_z^0\right)_{zzz}(x,0,t) = ni\omega u_{zz}^0(x,0,t),$$

which also requires u_{zz}^0 at the surface. Similarly the tidal return flow requires u_{zz}^0 at the surface.

Extending this to higher order it is found that the n^{th}-order equation requires $(u^{<k>})^(m)(x,0,t)$ with $m \leq n-1$ and $k \leq n-m-1$. Summarising, the first-order and second-order systems provide no problems. The third-order and higher-order systems require progressively higher derivatives at the surface, the accuracy of which should be investigated.

7.2.2 Computation of derivatives

At general order n we thus require $\left(A_\nu u_z^{<k>}\right)^{<n-k>}$ and $u^{<k>(n-k-1)}$ for $k=0,\dots,n-1$. As was shown, the first of these terms can be computed using the latter. In the derivation however, it is useful to compute both terms. This is done by taking derivatives of the momentum equation. In this section we will denote these to terms as follows:

$$\left(A_\nu{}^{<0>} u_z^{<n>}\right)^{(d+1)} \qquad \text{shear stress,}$$

$$u^{<n>(d)} \qquad \text{velocity derivatives.}$$

Derivatives of the shear stress term

Rewriting the momentum equation, for a general derivative d we find for $d \geq 0$

$$\left(A_\nu{}^{<0>} u_z^{<n>}\right)^{(d+1)} = u_t^{<n>(d)} + g\zeta_x^{<n>(d)} + \eta^{<n>(d)} + \varsigma^{<n>(d)} + \varsigma_\delta^{<n>(d)} - \psi^{<n>(d+1)}.$$

The right-hand side can be expanded according to

$$\zeta_x^{<n>(d)} = \begin{cases} \zeta_x^{<n>} & \text{if } d=0 \\ 0 & \text{if } d>0 \end{cases},$$

$$\eta^{<n>(d)} = \sum_{m=0}^{n-1}\sum_{i=0}^{d}\binom{d}{i}\left(u^{<m>(i)}u_x^{<n-m-1>(d-i)} + w^{<m>(i)}u^{<n-m-1>(d-i+1)}\right),$$

$$\varsigma^{<n>(d)} = \begin{cases} \frac{g}{\rho_0}\int_z^0 \rho_x^{<n>} & \text{if } d=0 \\ -\frac{g}{\rho_0}\rho_x^{<n>(d-1)} & \text{if } d>0 \end{cases},$$

$$\varsigma_\delta^{<n>(d)} = \frac{g}{\rho_0}\sum_{m=1}^{n-1}\sum_{k=0}^{n-1-m}\frac{1}{m!}\left(\rho_x^{<k>}(x,0,t)\right)^{(m-1+d)}\zeta^{<l_1>}\cdots\zeta^{<l_m>},$$

$$\forall l_1,\dots,l_m \text{ s.t. } \sum_{r=1}^{m} l_r = n-1-m-k,$$

$$\psi^{<n>(d+1)} = \sum_{m=1}^{n}\sum_{i=0}^{d+1}\binom{d+1}{i}A_\nu{}^{<m>(i)}u^{<n-m>(d-i+2)}.$$

The derivatives of the vertical velocity in the advection term is calculated according to

$$w^{<m>(d)} = \begin{cases} w & \text{if } d = 0 \\ -\frac{1}{B}\left(Bu^{<m>(d-1)}\right)_x & \text{if } d > 0 \end{cases}.$$

Restricting our attention to the terms involving the velocity, we require at most $u^{<n-1><d>}$ for advection and $u^{<n-1><d+2>}$ for the mixing term. We will show later that these quantities are indeed available.

Derivatives of the velocity

The derivative of the velocity can be derived from the derivative of the shear stress. First we expand

$$\left(A_v{}^{<0>}u_z^{<n>}\right)^{(d+1)} = \sum_{i=0}^{d+1}\binom{d+1}{i}A_v{}^{<0>(i)}u^{<n>(d-i+2)}.$$

We split-off the term with the highest velocity derivative and rewrite this term to

$$u^{<n>(d+2)} = \frac{1}{A_v{}^{<0>}}\left(\left(A_v{}^{<0>}u_z^{<n>}\right)^{(d+1)} - \sum_{i=1}^{d+1}\binom{d+1}{i}A_v{}^{<0>(i)}u^{<n>(d-i+2)}\right).$$

Order of calculation

Based on the calculation methods of the shear stress and velocity derivatives, we can determine when each of these terms can be computed. The tables below show, for each order of ε (n) and each order of derivation (d), when these terms can be computed. For example, the first table shows a 4 for $n = 3$, $d = 1$. This means that the shear stress $\left(A_v{}^{<0>}u_z^{<3>}\right)^{(2)}$ can be computed **at the beginning of** the fourth-order computation, i.e. before $u^{<4>}$, $w^{<4>}$ and $\zeta^{<4>}$ are computed.

As an example, the quantities needed at the fourth order are marked red. Clearly, all red-marked entries can be calculated at the beginning of the fourth-order computation or earlier. This means that we have a closed system for computing derivatives.

shear stress		0	1	2	3	4	5	$d \rightarrow$
$n \downarrow$	0	1	2	3	4	5	6	
	1	2	3	4	5	6		
	2	3	4	5	6			
	3	4	5	6				
	4	5	6					

velocity		0	1	2	3	4	5	6	$d \rightarrow$
$n \downarrow$	0	1	1	1	2	3	4	5	
	1	2	2	2	3	4	5		
	2	3	3	3	4	5			
	3	4	4	4	5				
	4	5	5	5					

Order of convergence: proof of concept

The method for computing derivatives presented above allows us to compute derivatives accurately, without losing numerical order of convergence. We will show a proof of concept of this statement by using a simple example. Let us consider a model with a uniform, time-independent eddy viscosity. We calculate the water motion up to tenth order and look at the numerical convergence of the leading-order velocity and its derivatives at the surface $z = 0$. The result is presented in Figure 7.1. It only presents the even derivatives, as the odd derivatives are all zero. The figure shows that the velocity and its derivatives display second-order convergence. The error is about an order of magnitude larger for the derivatives than for the velocity itself, but the error does not grow when increasing the order of derivation. The method of taking derivatives presented here thus seems robust.

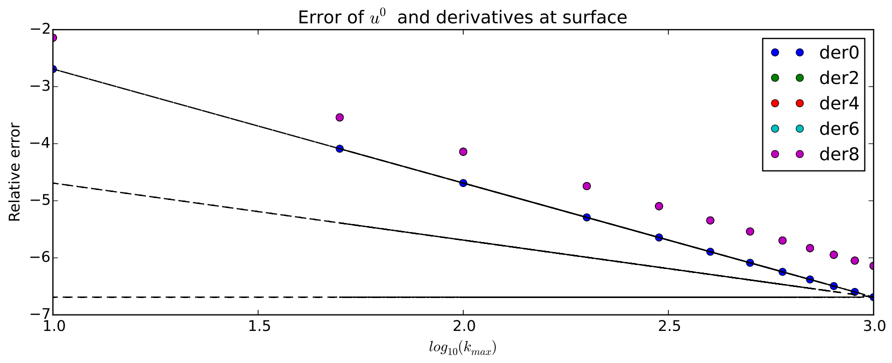


Figure 7.1: Relative error of u^0 and its derivatives at the surface ($z = 0$) plotted for vertical grid resolutions ranging from 10 to 1000 cells. The dashed lines mark no convergence (flat line), first-order convergence and second-order convergence (steepest sloping line).

8. Reference level

The reference level R can be predefined or computed using iFlow's `ReferenceLevel` module. This module determines the reference level on the basis of the river-induced sub-tidal water level set-up. To this end the reference level can be expressed as either R^0 following the leading-order river discharge or R^1 following the first-order river discharge. Since the equations are identical in either case, we will present a general equation for R, without any ordering markers. Since an estimate is sufficient, the reference level is computed using only the sub-tidal components of the leading-order eddy viscosity and partial slip parameter. The appropriate sub-tidal equations follow from the leading-order and first-order system of equations and read (omitting unnecessary ordering marks and subscripts denoting the riverine component)

$$-gR_x + \left(\langle {A_v}^0 \rangle u_z\right)_z = 0, \tag{8.1}$$

- $$\langle {A_v}^0 \rangle (x,R) u_z(x,R) = 0, \tag{8.2}$$
- $$\begin{cases} \langle {A_v}^0 \rangle (x,-H) u_z(x,-H) = \langle s_f(x) \rangle u(x,-H), & \text{or} \\ u(x,-H) = 0, \end{cases} \tag{8.3}$$

$$\int_{-H}^{R} u(x,z)\, dz = \frac{Q}{B}, \tag{8.4}$$

- $$R(0) = 0, \tag{8.5}$$

(8.6)

The solution to Equation (8.1) follows from integrating twice

$$u = -gR_x \left(\int_{-H}^{z} \frac{R-\tilde{z}}{A_v} d\tilde{z} + \begin{cases} \frac{R+H}{s_f} & \text{(partial slip)} \\ 0 & \text{(no-slip)} \end{cases} \right).$$

Inserting this into Equation (8.4) we find

$$-gR_x \underbrace{\int_{-H}^{R}\left(\int_{-H}^{z}\frac{R-\tilde{z}}{A_v}\,d\tilde{z}+\begin{cases}\frac{R+H}{s_f} & \text{(partial slip)}\\ 0 & \text{(no-slip)}\end{cases}\right)dz}_{\mathscr{U}(R)}=\frac{Q}{B}.$$

This equation is non-linear in R and reminds us of the classical equation for stationary backwater curves. Following a similar method as for backwater curves, the solution can be computed numerically using an explicit finite differences approach. Starting at $x=0$, it is known that $R=0$ from the boundary condition. This can be used to compute R_x by rewriting the above equation to

$$R_x(0)=-\frac{Q}{g\mathscr{U}(0)B(0)}.$$

The reference level R at a point $x=\Delta x$ then follows from

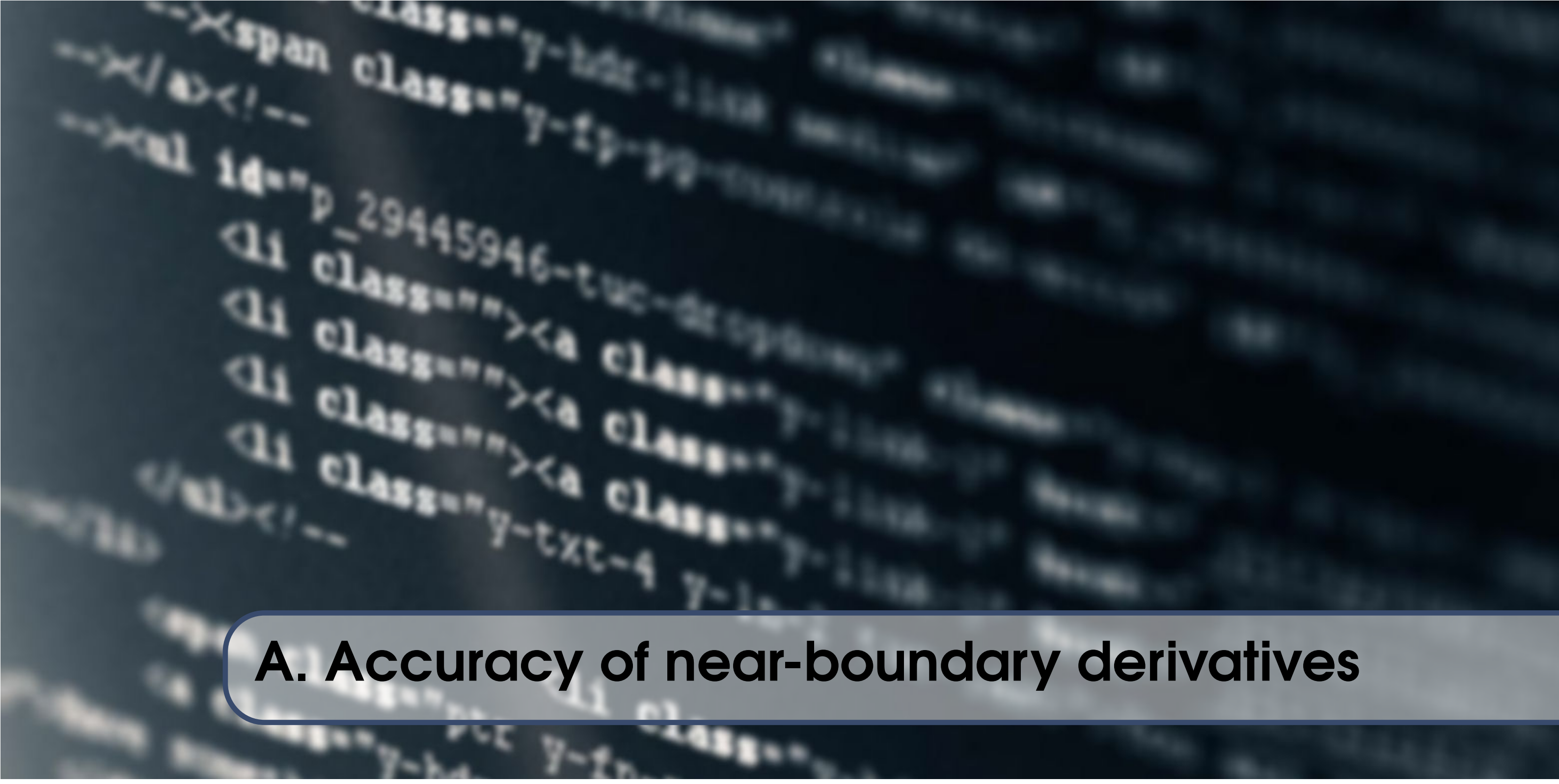
$$R(\Delta x)=R(0)+\Delta x R_x(0).$$

Continuing this approach, the following two equations are solved at every grid point

$$R_x(x)=-\frac{Q}{g\mathscr{U}(x)B(x)}, \tag{8.7}$$

$$R(x+\Delta x)=R(x)+\Delta x R_x(x). \tag{8.8}$$

The thus computed reference level provides an estimate of the sub-tidal water level set-up directly caused by the river discharge. Any effect of time-varying eddy viscosity or partial slip parameter as well as non-linear interactions involving the river discharge are incorporated in the water motion and will not be neglected. Likewise, the water level term ζ^0_{river} or ζ^1_{river} corrects for the difference in numerical errors between the first-order numerical scheme used here and any other method used to solve the hydrodynamics equations.



A. Accuracy of near-boundary derivatives

The higher-order model presented in Chapter 7 requires vertical derivatives of arbitrary order of the horizontal velocity at the surface. In Chapter 7 it is explained how these derivatives can be computed with a high degree of accuracy. The method developed there is motivated by the loss of accuracy when simply taking the derivative using a numerical finite difference method or a spectral method. This loss of accuracy is demonstrated below.

A.1 Finite difference method

While it is possible to solve the higher-order system equation using finite differences, this method encounters several problems. The first problem is one of general accuracy. As the higher-order forcing terms require the multiplication of many previously numerically calculated quantities, there is a risk of a build-up of the numerical error. This problem is however encountered with any approximating solution method and is best identified by trying different grid resolution. The second, more imminent problem is in taking high-order derivatives at the surface. The higher-order terms require taking arbitrary order derivatives of numerical entities. We will show below that this is possible, but encounters serious accuracy problems.

Let us consider the n$^{\text{th}}$ derivative of a numerical function $u(z)$ at the boundary $z=0$ (i.e. $R=0$ for simplicity). This derivative can be determined by using a Taylor series, which in general reads

$$u(z) = \sum_{m=0}^{\infty} \frac{1}{m!} u^{(m)}(0) z^m.$$

This series can be used to distil $u^{(n)}(0)$ by taking a set of $N+1$ points $0 = z_0 > z_1 > \ldots > z_N$, evaluating

$$
\begin{aligned}
u^{(n)}(0) &= \sum_{k=0}^{N} \alpha_k u(z_k) \\
&= \sum_{m=0}^{\infty} \frac{1}{m!} u^{(m)}(0) \sum_{k=0}^{N} \alpha_k z_k^m
\end{aligned}
$$

and requiring that

$$
\begin{cases}
\sum_{k=0}^{N} \alpha_k z_k^n = n! \\
\sum_{k=0}^{N} \alpha_k z_k^m = 0 \quad \forall m \neq n,
\end{cases}
$$

This set of requirements can be solved when taking as many equations as variables, which means solving for $m = 0, \ldots, N$. Note that this implies that $N \geq n$. The order of accuracy of the method is simply given by $N - n + 1$, so that first order accuracy is obtained when $N = n$. The coefficients α_k can be solved from the matrix equation

$$
\begin{bmatrix} z_0^0 & \cdots & z_N^0 \\ \vdots & \ddots & \vdots \\ z_N^0 & \cdots & z_N^N \end{bmatrix}
\begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_N \end{bmatrix} = n! \underline{e}_n
$$

This matrix equation can lead to an erroneous estimate for $u^{(n)}(0)$ in two ways. First, the error depends on the numerical order of accuracy, i.e. $N - n + 1$. The error decreases with increasing order of accuracy and increases with increasing order of the derivative. The latter is because higher-order derivatives require more z-points, so that the the first and last point are spaced further apart. This error can be decreased again by refining the grid. Second, the error depends on the condition number of the above matrix. The condition quickly becomes worse for increasing N, so that a higher order of accuracy comes with a larger condition number. The condition is also worse for finer grids. The two sources of errors thus pose a strict trade-off between the condition number and the order of accuracy and the grid spacing. As a result, the error on the direct numerical computation of higher-order derivatives cannot be bounded. Finite difference differentiation for higher-order derivatives is therefore not feasible.

A.2 Spectral methods

As higher-order derivatives are a problem in finite difference methods, we try a solution using spectral methods. As the momentum equation with a parabolic eddy viscosity profile looks a lot like the Legendre differential equation, the solution is expressed in terms of Legendre polynomials. Consider the arbitrary-order momentum equation assuming $R = 0$ for simplicity

$$ni\omega\hat{u}(x,z) - \left(A_v(x,z)\hat{u}_z(x,z)\right)_z = -g\hat{\zeta}_x(x) + F(x,z), \qquad -H(x) \leq z \leq 0 \tag{A.1}$$

- $$A_v(x,0)\hat{u}_z(x,0) = F_{\text{surf}}, \tag{A.2}$$
- $$u(x,-H) = 0. \tag{A.3}$$

using the usual notation. We assume here that the eddy viscosity has a parabolic profile according to

$$A_v(x,z) = A_{v,0}(x)(z_s^* H(x) - z)(z + (z_0^* + 1)H(x)). \tag{A.4}$$

Note that this parabolic eddy viscosity uses a dimensionless roughness z_0^* at the bed and z_s^* at the surface. The latter is required in order to have a non-zero eddy viscosity at the surface and thus allow for the surface boundary condition. Substituting (A.4) we rewrite Equation (A.1) to

$$ni\omega\hat{u} - A_{v,0}\left((z_s^* - z)(z + (z_0^* + 1)H)\hat{u}_z\right)_z = -g\hat{\zeta}_x + F, \qquad -H \le z \le 0 \tag{A.5}$$

Equation (A.5) is further transformed to a new coordinate $\hat{z}$, which is defined as

$$\hat{z} = \frac{2z}{H} + 1.$$

This transforms Equation (A.5) to

$$ni\omega\hat{u} - A_{v,0}\left((2z_s^* + 1 - \hat{z})(2z_0 + 1 + \hat{z})\hat{u}_{\hat{z}}\right)_{\hat{z}} = -g\hat{\zeta}_x + F, \qquad -1 \le \hat{z} \le 1,$$

which is further rewritten to

$$ni\omega\hat{u} - A_{v,0}\left((1-\hat{z}^2)\hat{u}_{\hat{z}} + 2z_s^*(1+\hat{z})\hat{u}_{\hat{z}} + 2z_0^*(1-\hat{z})\hat{u}_{\hat{z}} + 4z_0^* z_s^* \hat{u}_{\hat{z}}\right)_{\hat{z}} = -g\hat{\zeta}_x + F, \tag{A.6}$$

- $2HA_{v,0}z_s^*(z_0^*+1)\hat{u}_{\hat{z}}(x,1) = F_{\mathsf{surf}},$ (A.7)
- $\hat{u}(x,-1) = 0.$ (A.8)

This equation can be simplified by making use of the Legendre eigenvalue problem

$$\left((1-\hat{z}^2)P_{n,\hat{z}}\right)_{\hat{z}} + n(n+1)P_n = 0 \qquad -1 \le \hat{z} \le 1, \tag{A.9}$$

The solutions of this eigenvalue problem are Legendre polynomials. The velocity signal $\hat{u}$ can now be expanded in terms of the Legendre polynomials found above. The expansion is truncated after $N+1$ terms so that the series reads

$$\hat{u}(x,z) = \sum_{k=0}^{N} c_k(x)P_k(\hat{z}). \tag{A.10}$$

Expressions (A.9) and (A.10) are substituted in Equation (A.6) to find

$$\sum_{k=0}^{N}(ni\omega + A_{v,0}k(k+1))c_kP_k - A_{v,0}\left((2z_s^*(1+\hat{z}) + 2z_0^*(1-\hat{z}) + 4z_0^*z_s^*)c_kP_{\hat{z},k}\right)_{\hat{z}} = -g\hat{\zeta}_x + F.$$

This rewrites to

$$\sum_{k=0}^{N}(ni\omega + A_{v,0}k(k+1))c_kP_k - A_{v,0}p_{\hat{z}}c_kP_{\hat{z},k} - A_{v,0}pc_kP_{\hat{z}\hat{z},k} = -g\hat{\zeta}_x + F, \tag{A.11}$$

- $\sum_{k=0}^{N} 2HA_{v,0}z_s^*(z_0^*+1)c_kP_{\hat{z},k}(1) = F_{\mathsf{surf}},$ (A.12)
- $\sum_{k=0}^{N} c_kP_k(-1) = 0,$ (A.13)

where $p(\hat{z}) = 2z_s^*(1+\hat{z}) + 2z_0^*(1-\hat{z}) + 4z_0^*z_s^*$. These equations are evaluated on a Legendre-Gauss-Lobatto grid. This is the optimal grid for solving the above problem in such a way that the boundary points are included. The grid is determined as the zeros of the polynomial $P_{\hat{z},N}$ supplemented by the boundary points $\hat{z} = -1$ and $\hat{z} = 1$. Resulting is a matrix-vector equation $A\underline{c} = b$, which can be solved for the vector of coefficients $\underline{c}$.

For a first simple test we take the leading-order balance, i.e. $F = 0$, $F_{\mathsf{surf}} = 0$. The resulting velocity profile for $z_0^* = z_s^* = 0.001$ and 50 Legendre polynomials (i.e. $N = 49$) is plotted in Figure A.1.

Next considering the derivatives, we present the values of the first 10 derivatives of the complex amplitude $\hat{u}$ at the surface in Table A.1. The results are unrealistically large for the

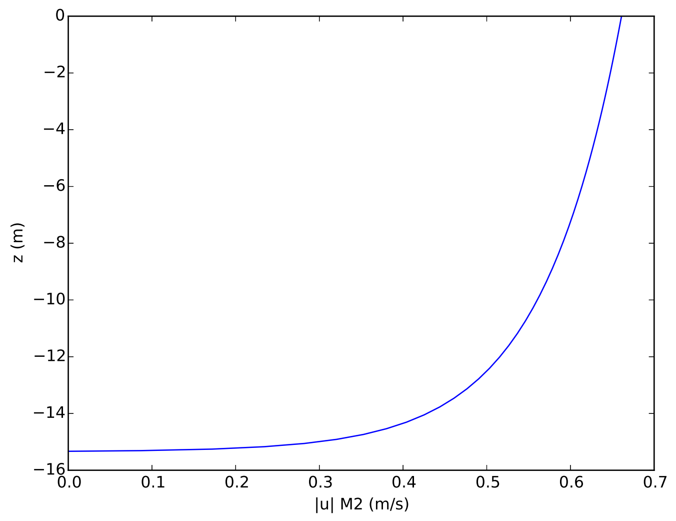


Figure A.1: Vertical profile of the leading-order M_2 velocity amplitude.

	$N = 49$	$N = 50$
Value	$0.331398590135 + 0.571918657833i$	$0.331354073632 + 0.571862224246i$
Derivative 1	$-3.24740234703e^{-15} - 2.88657986403e^{-15}i$	$1.85962356625e^{-15} + 4.4408920985e^{-15}i$
Derivative 2	$-9.17274192145 - 14.2384109211i$	$-13.7628423338 - 22.2530758058i$
Derivative 3	$-1503.97105291 - 2052.06633488i$	$-5709.71494084 - 9434.83785597i$
Derivative 4	$100188.053184 + 324613.346467i$	$-1929493.09078 - 3246682.90378i$
Derivative 5	$142776778.279 + 282246571.605i$	$-538844780.156 - 918695754.093i$
Derivative 6	$50570037196.1 + 94572152629.9i$	$-126308916525 - 217338867418i$
Derivative 7	$1.22127010365e^{13} + 2.23737573114e^{13}i$	$-2.526203683e^{13} - 4.37498283231e^{13}i$
Derivative 8	$2.33008369769e^{15} + 4.22452310456e^{15}i$	$-4.37707835081e^{15} - 7.61546724019e^{15}i$
Derivative 9	$3.72121394705e^{17} + 6.70544468807e^{17}i$	$-6.65801737027e^{17} - 1.16230875729e^{18}i$
Derivative 10	$5.13158186669e^{19} + 9.21027174533e^{19}i$	$-8.99136132495e^{19} - 1.57360619242e^{20}i$

Table A.1: Surface derivatives of $\hat{u}$.

second and higher derivatives. Tests with N larger does only make the results worse. For example the second derivative for $N = 100$ is equal to approximately $-154 - 245i$.

Looking closer at the second derivative. This is computed according to $\hat{u}_{\hat{z}\hat{z}}(1) = \sum_{k=0}^{N} c_k P_{\hat{z}\hat{z},k}(1)$. The values of $P_{\hat{z}\hat{z},k}(1)$ diverge, while the absolute values of the coefficients c_k converge. The resulting cumulative sums $\sum_{k=0}^{n} c_k P_{\hat{z}\hat{z},k}(1)$ for $n = 1, 2, \ldots, N$ converge, but only after strongly oscillating between values much larger than the final converged result, see Figure A.2. The cumulative sums thus involve adding and subtracting very large numbers. As this adding and subtracting needs to be done numerically, this leads to an error of the converged value of the derivative which is larger than the actual value of the derivative. Spectral methods therefore also cannot solve the problem of taking high-order derivatives of u at the surface.

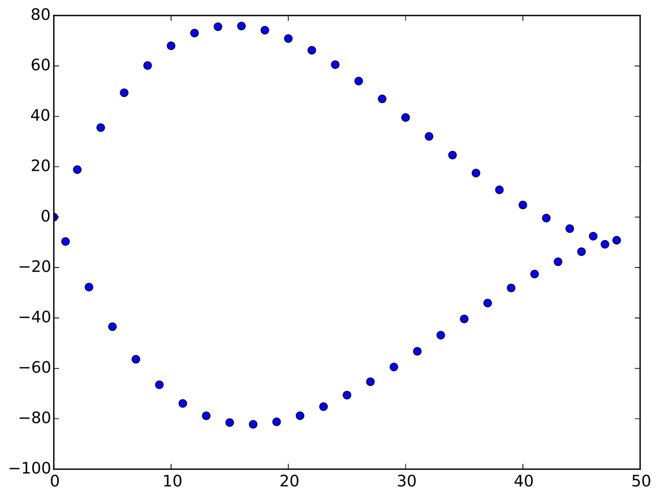


Figure A.2: Value of the cumulative sums $\sum_{k=0}^{n} c_k P_{\hat{z}\hat{z},k}(1)$ for $n = 1, 2, \ldots, N = 49$, with n on the horizontal axis. For $n = N$, this sum equals the second derivative of the velocity u at the surface. However, this computation is erroneous, since the sum involves numerically adding and subtracting large numbers.

II

Sediment dynamics

9. Sediment dynamics

This chapter provides a comprehensive mathematical derivation of a numerical sediment module for iFlow. Starting point is the semi-analytical sediment model of Brouwer et al. (2016b), which is similar to the model by Chernetsky et al. (2010), with the addition of the sediment advection term. In the present model, we will add the following features

1. Vertical variation of K_v and w_s,
2. Leading-order time variation of K_v and w_s,

9.1 Equation

We start from the two-dimensional width-averaged sediment equation as presented by e.g. Chernetsky et al. (2010), which reads

$$c_t + uc_x + wc_z - (w_s c)_z = (K_v c_z)_z + (K_H c_x)_x, \tag{9.1}$$

- $$w_s c + K_v c_z = 0 \qquad \text{at } z = R + \zeta, \tag{9.2}$$
- $$w_s c + K_v c_z = D - E \qquad \text{at } z = -H. \tag{9.3}$$

Here c is the suspended sediment mass concentration, w_s is the fall velocity, K_v is the vertical eddy diffusivity, K_H is the horizontal eddy diffusivity and D and E denote the deposition and erosion. Due to our solution method, the horizontal boundary conditions are not yet required. These will be provided in the availability equation (Section 10).

At the bed we assume a single sediment layer without considering bed stratigraphy. The deposition D is given by settling, i.e.

$$D = w_s c.$$

The erosion is formulated as

$$E = \mathscr{E} f(a)$$

where $\mathscr{E}$ is the potential erosion, i.e. the erosion rate irrespective of the sediment availability under conditions of abundant sediment supply. The function $f(a)$ is the erodability as a function of the sediment availability. The erodability is assumed to be a number between 0 and 1 and is assumed to be a constant in time. The formulation for $\mathscr{E}$ is discussed in Section 9.1.1.

9.1.1 Bed erosion

We will compare two bed erosion formulations. The first is the expression used by Chernetsky (2012). His expression is adapted to use the erodability instead of availability (also see Brouwer et al. (2016a))

$$E = w_s c_*, \tag{9.4}$$

where

$$c_* = \frac{\rho_s}{\rho_0 g' d_s} f_{\text{inf}} |\tau_b| f(a),$$

where ρ_s is the bulk dry sediment density, $g' = g(\rho_s - \rho_0)/\rho_0$ is reduced gravity, d_s is a typical grain size, f_{inf} is some calibration constant and τ_b is the bed shear stress. Chernetsky (2012) uses the availability a instead of $f_{\text{inf}} f(a)$ in his original erosion formulation. The magnitude of his a is controlled by a calibration parameter a^*, so that he has the same number of calibration parameters.

Alternatively, we consider the standard Partheniades model (Kandiah, 1974), slightly rewritten

$$\mathscr{E} = \begin{cases} M|\tau_b - \tau_c| & \text{if } |\tau_b| > \tau_c, \\ 0 & \text{if } |\tau_b| \le \tau_c. \end{cases}$$

where M is some erosion parameter and τ_c is the critical shear stress. Although almost never mentioned explicitly, this expression holds for the potential erosion $\mathscr{E}$ assuming there is sufficient sediment to erode (also see above). It thus requires additional bookkeeping of the sediment availability to compute the actual erosion. For simplicity we will assume that $\tau_c = 0$ to find

$$\mathscr{E} = M|\tau_b|. \tag{9.5}$$

Expressions (9.4) and (9.5) are equivalent if $M = w_s \frac{\rho_s}{\rho_0 g' d_s} f_{\text{inf}}$. Typically, however, M is regarded as a calibration parameter that has no apparent relation to the sediment density, size or fall velocity. As both expressions contain the same number of calibration parameters they can be regarded as fully equivalent. Momentarily the sediment model only implements the Chernetsky et al. (2010) model.

9.2 Scaling and ordering

For the scaling of the sediment model the reader is referred to the iFlow manual for the package `semi_analytical` (Brouwer, 2017). We then make an ordering of the concentration, fall velocity and eddy diffusivity as

$$\begin{aligned} c &= c^0 + c^1 + c^2 + \ldots, \\ w_s &= w_s^0 + w_s^1 + w_s^2 + \ldots, \\ K_v &= K_v{}^0 + K_v{}^1 + K_v{}^2 + \ldots, \end{aligned}$$

where c^0, w_s^0, $K_v{}^0$ are of leading order, c^1, w_s^1, $K_v{}^1$ are of order ε etc. Similarly we make an ordering of the erosion E, which then breaks down into an ordering of the bed shear stress and fall velocity. The erodability f is not ordered.

The ordering of c, w_s and K_v are inserted into the sediment equation (9.1). This surface boundary condition is then linearised around the reference level $z = R$. Using the ordering and the scaling we find the following equation at the leading leading order

$$c_t^0 - \left(w_s^0 c^0 + {K_v}^0 c_z^0\right)_z = 0, \tag{9.6}$$

- $$w_s^0 c^0 + {K_v}^0 c_z^0 = 0 \qquad \text{at } z = R, \tag{9.7}$$
- $$ {K_v}^0 c_z^0 = -E^0 \qquad \text{at } z = -H. \tag{9.8}$$

And at the first order

$$c_t^1 - \left(w_s^0 c^1 + {K_v}^0 c_z^1\right)_z = -u^0 c_x^0 - w^0 c_z^0 - \left(w_s^1 c^0\right)_z - \left({K_v}^1 c_z^0\right)_z, \tag{9.9}$$

- $$w_s^0 c^1 + {K_v}^0 c_z^1 = -w_s^1 c^0 - {K_v}^1 c_z^0 - \underbrace{\left(w_s^0 c^0 + {K_v}^0 c_z^0\right)_z}_{c_t^0} \zeta^0 \qquad \text{at } z = R, \tag{9.10}$$
- $$ {K_v}^0 c_z^1 = -E^1 - {K_v}^1 c_z^0 \qquad \text{at } z = -H. \tag{9.11}$$

The equality $c_t^0 = \left(w_s^0 c^0 + {K_v}^0 c_z^0\right)_z$ used in (9.10) is derived from (9.6). The second-order sediment dynamics will be largely disregarded, except for the sediment concentration induced by resuspension by the river flow. The equation for this reads

$$c_t^2 - \left(w_s^0 c^2 + {K_v}^0 c_z^2\right)_z = 0, \tag{9.12}$$

- $$w_s^0 c^2 + {K_v}^0 c_z^2 = 0 \qquad \text{at } z = R, \tag{9.13}$$
- $$ {K_v}^0 c_z^2 = -E^2_{\text{river-river}} \qquad \text{at } z = -H. \tag{9.14}$$

For ease of writing, we define the following quantities

$$\eta_c^1 = u^0 c_x^0 + w^0 c_z^0, \tag{9.15}$$
$$\psi_c^1 = {K_v}^1 c_z^0, \tag{9.16}$$
$$\xi_c^1 = w_s^1 c^0, \tag{9.17}$$
$$\chi_c^1 = c_t^0 \zeta^0, \tag{9.18}$$
$$\tag{9.19}$$

The resulting equation with boundary conditions reads

$$c_t^0 - \left(w_s^0 c^0 + {K_v}^0 c_z^0\right)_z = 0, \tag{9.20}$$

- $$w_s^0 c^0 + {K_v}^0 c_z^0 = 0 \qquad \text{at } z = R, \tag{9.21}$$
- $$ {K_v}^0 c_z^0 = -\mathscr{E}^0 f \qquad \text{at } z = -H. \tag{9.22}$$

And at the first order

$$c_t^1 - \left(w_s^0 c^1 + {K_v}^0 c_z^1\right)_z = -\eta_c^1 - \xi_c^1 - \psi_{c,z}^1, \tag{9.23}$$

- $$w_s^0 c^1 + {K_v}^0 c_z^1 = -\xi_c^1 - \psi_c^1 - \chi_c^1 \qquad \text{at } z = R, \tag{9.24}$$
- $$ {K_v}^0 c_z^1 = -\mathscr{E}^1 f - \psi_c^1 \qquad \text{at } z = -H. \tag{9.25}$$

Finally, the second-order equation, with only river-induced resuspension, reads

$$c_t^2 - \left(w_s^0 c^2 + {K_v}^0 c_z^2\right)_z = 0, \tag{9.26}$$

- $$w_s^0 c^2 + {K_v}^0 c_z^2 = 0 \qquad \text{at } z = R, \tag{9.27}$$
- $$ {K_v}^0 c_z^2 = -\mathscr{E}^2_{\text{river-river}} f \qquad \text{at } z = -H. \tag{9.28}$$

9.3 Abstract solution

The sediment equation can be seen as an abstract linear equation in c forced by a number of components. We identify five forcing components:

1. erosion ($\mathscr{E}^0 f$, $\mathscr{E}^1 f$, $\mathscr{E}^2 f$),
2. sediment advection (η_c^1),
3. the no-flux surface boundary condition (χ_c^1),
4. higher-order fall velocity variations (ξ_c^1),
5. higher-order eddy viscosity variations (ψ_c^1).

Only the erosion is found at the leading order, while all five appear at the first order.

The remaining unknown in the forcing components is the erodability $f(a)$. It will be shown here that all the forcing components can be expressed in terms of $f(a)$ and we will use this to define the abstract form of the solution to the sediment equation in terms of $f(a)$. First, the leading-order erosion was already written in terms of $f(a)$, see (**??**). We can therefore express the leading-order solution as

$$c^0 = \hat{\mathscr{C}}^0 f(a), \tag{9.29}$$

where $\hat{\mathscr{C}}^0$ is some linear operator that is computed numerically.

The abstract leading-order solution (9.29) can then be used to rewrite the forcing components for the first order. Substituting this expression yields

$$\begin{aligned}
\eta_c^1 &= u^0 \hat{\mathscr{C}}_x^0 f(a) + u^0 \hat{\mathscr{C}}^0 f_x(a) + w^0 \hat{\mathscr{C}}_z^0 f(a), \\
\psi_c^1 &= K_v{}^1 \hat{\mathscr{C}}_z^0 f(a), \\
\xi_c^1 &= w_s^1 \hat{\mathscr{C}}_z^0 f(a), \\
\chi_c^1 &= \left(\zeta^0 \hat{\mathscr{C}}_t^0\right) f(a), \\
E^1 &= \mathscr{E}^1 f.
\end{aligned}$$

We thus see that the first-order forcing scales with either $f(a)$ and $f_x(a)$. The first-order solution can then be written as

$$c^1 = \hat{\mathscr{C}}_a^1 f(a) + \hat{\mathscr{C}}_{a_x}^1 f_x(a). \tag{9.30}$$

Similarly, the second-order equation is forced by $E^2 = \mathscr{E}^2 f$ and can be written as

$$c^2_{\text{river-river}} = \hat{\mathscr{C}}_a^2 f(a). \tag{9.31}$$

9.4 Harmonic analysis and vector notation

We will write the concentration, fall velocity and vertical eddy diffusivity in terms of harmonic components of the form

$$\begin{aligned}
c^i(x,z,t) &= Re\left(\sum_{n=0}^{p} \hat{c}^i(x,z) e^{ni\omega t}\right), \\
w_s^i(x,z,t) &= Re\left(\sum_{n=0}^{p} \hat{w}_s^i(x,z) e^{ni\omega t}\right), \\
K_v{}^i(x,z,t) &= Re\left(\sum_{n=0}^{p} \hat{K}_v^i(x,z) e^{ni\omega t}\right).
\end{aligned}$$

The concentration is most easily solved for when the negative Fourier components are taken into account as well. We will therefore use the alternative notation

$$c^i(x,z,t) = Re\left(\sum_{n=-p}^{p} \breve{c}^i(x,z)e^{ni\omega t}\right), \tag{9.32}$$

with $\hat{c}^i_n = \breve{c}^i_n + \overline{\breve{c}}^i_{-n}$ for $n = 1,\ldots,p$ and $\hat{c}^i_0 = \breve{c}^i_0$. We will also define the following vectors:

$$\begin{aligned}
\underline{\breve{c}}^i &= \left[\breve{c}^i_{-p},\ldots,\breve{c}^i_0,\ldots,\breve{c}^i_p\right]^T,\\
\underline{\breve{w}_s}^i &= \left[\overline{\hat{w}_{sp}}^i,\ldots,\overline{\hat{w}_{s1}}^i,\hat{w}^i_{s0},\ldots,\hat{w}^i_{sp}\right]^T,\\
\underline{\breve{K}_v}^i &= \left[\overline{\hat{K}_{vp}}^i,\ldots,\overline{\hat{K}_{v1}}^i,\hat{K}^i_{v0},\ldots,\hat{K}^i_{vp}\right]^T,\\
\underline{\breve{\eta}_c}^i &= \left[\overline{\breve{\eta}_{c,p}}^i,\ldots,\overline{\breve{\eta}_{c,1}}^i,\breve{\eta}^i_{c,0},\ldots,\breve{\eta}^i_{c,p}\right]^T,\\
\underline{\breve{\psi}_c}^i &= \left[\overline{\breve{\psi}_{c,p}}^i,\ldots,\overline{\breve{\psi}_{c,1}}^i,\breve{\psi}^i_{c,0},\ldots,\breve{\psi}^i_{c,p}\right]^T,\\
\underline{\breve{\xi}_c}^i &= \left[\overline{\breve{\xi}_{c,p}}^i,\ldots,\overline{\breve{\xi}_{c,1}}^i,\breve{\xi}^i_{c,0},\ldots,\breve{\xi}^i_{c,p}\right]^T,\\
\underline{\breve{\chi}_c}^i &= \left[\overline{\breve{\chi}_{c,p}}^i,\ldots,\overline{\breve{\chi}_{c,1}}^i,\breve{\chi}^i_{c,0},\ldots,\breve{\chi}^i_{c,p}\right]^T.
\end{aligned}$$

Substituting the above vectors in the ordered equations (9.6)-(9.11), we obtain the following matrix equations

$$D\underline{\breve{c}}^0 - \left(\mathscr{W}_s^0\underline{\breve{c}}^0 + \mathscr{K}^0\underline{\breve{c}}^0_z\right)_z = 0, \tag{9.33}$$

- $\mathscr{W}_s^0\underline{\breve{c}}^0 + \mathscr{K}^0\underline{\breve{c}}^0_z = 0$ at $z = R$, (9.34)
- $\mathscr{K}^0\underline{\breve{c}}^0_z = -\mathscr{E}^0 f$ at $z = -H$. (9.35)

And at the first order

$$D\underline{\breve{c}}^1 - \left(\mathscr{W}_s^0\underline{\breve{c}}^1 + \mathscr{K}^0\underline{\breve{c}}^1_z\right)_z = -\underline{\breve{\eta}_c}^1 - \underline{\breve{\xi}}^1_z - \underline{\breve{\psi}}^1_{c,z}, \tag{9.36}$$

- $\mathscr{W}_s^0\underline{\breve{c}}^1 + \mathscr{K}^0\underline{\breve{c}}^1_z = -\underline{\breve{\xi}}^1 - \underline{\breve{\psi}}^1_c - \underline{\breve{\chi}}^1_c$ at $z = R$, (9.37)
- $\mathscr{K}^0 c^1_z = -\mathscr{E}^1 f - \underline{\breve{\psi}}^1_c$ at $z = -H$. (9.38)

Here the matrix D is a diagonal matrix with components $-pi\omega t,\ldots,pi\omega t$. The matrices $\mathscr{W}_s$, $\mathscr{K}$ and $\mathscr{E}^i$ contain the components of the fall velocity, eddy diffusivity and erosion respectively. The sediment balance is solved in terms of the unknown f, which is a function of x. The second-order model is of a similar form as the leading-order model and is omitted here for brevity.

9.5 Numerical solution method

The general form of the sediment balance reads, omitting $\breve{\cdot}$ and order notation for generality,

$$D\underline{c} - \left(\mathscr{W}_s\underline{c} + \mathscr{K}\underline{c}_z\right)_z = \underline{f}, \tag{9.39}$$

- $\mathscr{W}_s\underline{c} + \mathscr{K}\underline{c}_z = \underline{f}_{\text{surf}}$ at $z = R$, (9.40)
- $\mathscr{K}\underline{c}_z = \underline{f}_{\text{bed}}$ at $z = -H$. (9.41)

We use a second-order central method for the diffusive term $\left(\mathscr{K}\underline{c}_z\right)_z$. This method uses small grid spacing by acting over half grid cells. A first-order upwind method over full grid cells is used for the settling term $(\mathscr{W}_s \underline{c})_z$. Since the settling is directed downwards, the upwind direction is known and constant. The resulting discretisation for interior grid cells reads

$$D\underline{c}_k - \frac{\left(\mathscr{W}_{s,k-1}\underline{c}_{k-1} - \mathscr{W}_{s,k}\underline{c}_k\right)}{\Delta z_{k-\frac{1}{2}}} - \frac{\left(\mathscr{K}_{k-\frac{1}{2}}\underline{c}_{z,k-\frac{1}{2}} - \mathscr{K}_{k+\frac{1}{2}}\underline{c}_{z,k+\frac{1}{2}}\right)}{\Delta z_{k-\frac{1}{2}}} = \underline{f}_k,$$

which further rewrites to

$$D\underline{c}_k - \frac{\left(\mathscr{W}_{s,k-1}\underline{c}_{k-1} - \mathscr{W}_{s,k}\underline{c}_k\right)}{\Delta z_{k-\frac{1}{2}}} - \frac{\left(\mathscr{K}_{k-\frac{1}{2}}\frac{\underline{c}_{k-1}-\underline{c}_k}{\Delta z_{k-1}} - \mathscr{K}_{k+\frac{1}{2}}\frac{\underline{c}_k-\underline{c}_{k+1}}{\Delta z_k}\right)}{\Delta z_{k-\frac{1}{2}}} = \underline{f}_k.$$

The eddy diffusivity at the points $k-\frac{1}{2}$ are defined as

$$\mathscr{K}_{k-\frac{1}{2}} = \frac{\mathscr{K}_{k-1} + \mathscr{K}_k}{2}.$$

A naive and simple way to discretise the boundary conditions is according to a first-order scheme

$$\mathscr{W}_{s,0}\breve{\underline{c}}_0 + \frac{\mathscr{K}_0 - \mathscr{K}_1}{\Delta z_0} = \underline{f}_{\mathrm{surf}},$$

$$\frac{\mathscr{K}_{k_{\mathrm{kmax}-1}} - \mathscr{K}_{k_{\mathrm{kmax}}}}{\Delta z_{k_{\mathrm{kmax}-1}}} = \underline{f}_{\mathrm{bed}}.$$

10. Bed exchange

10.1 Derivation of the equation

The exchange of sediment with the bed is modelled through the sediment availability. This is modelled using the balance equation

$$\beta a_t = D - E, \tag{10.1}$$

where a is the sediment availability (dimensionless) and D and E are the erosion and deposition per metre width of the estuary. The sediment availability can have different interpretations and the parameter β is included to convert between some different interpretations. If $\beta = \rho_s(1-n)H_0$, with ρ_s the dry bed density, n the bed porosity (i.e. $\rho_s(1-n)$ equals ρ_{bed}, the actual bed density) and H_0 a reference thickness (in metre), the availability equals the thickness of the sediment layer relative to H_0. If $H_0 = 1$ m, (10.1) corresponds to the Exner equation. Here we will take β equal to a a reference mass per bottom surface area M_0 (in kg/m^2). By default we will choose $\beta = M_0 = 1$ kg/m^2.

Following Chernetsky (2012) we rewrite this equation by integrating the sediment balance (9.1)

$$\int_{-H}^{R+\zeta} c_t \, dz + \int_{-H}^{R+\zeta} uc_x + wc_z \, dz - \int_{-H}^{R+\zeta} (w_s c + K_v c_z)_z \, dz = \frac{1}{B} \int_{-H}^{R+\zeta} (BK_H c_x)_x \, dz$$

This is rewritten using the Leibniz rule for integration

$$\begin{aligned}
&\left(\int_{-H}^{R+\zeta} c \, dz \right)_t - c\zeta_t + \frac{1}{B} \left(\int_{-H}^{R+\zeta} Buc \, dz \right)_z - uc(R+\zeta)_x + uc(-H)_x \\
&- \int_{-H}^{R+\zeta} (Bu)_x c + w_z c \, dz + wc_z|_{R+\zeta} - wc|_{-H} + (w_s c + K_v c_z)|_{R+\zeta} - (w_s c + K_v c_z)|_{-H} \\
&\qquad\qquad = \frac{1}{B} \left(\int_{-H}^{R+\zeta} BK_H c_x \, dz \right)_x - K_H c_x (R+\zeta)_x + K_H c_x (-H)_x.
\end{aligned}$$

Here

$$
\begin{aligned}
c\,(w-\zeta_t-u(R+\zeta)_x) &= 0 \text{ at } z=R+\zeta && \text{kinematic BC}\\
c\,(w+uH_x) &= 0 \text{ at } z=-H && \text{kinematic BC}\\
w_s c+K_v c_z-K_H c_x (R+\zeta)_x &= 0 \text{ at } z=R+\zeta && \text{surface BC}\\
w_s c+K_v c_z+K_H c_x (H)_x &= D-E \text{ at } z=-H && \text{bottom BC}\\
\int_{-H}^{R+\zeta} (Bu)_x c+w_z c\,dz &= 0 && \text{continuity.}
\end{aligned}
$$

We then obtain

$$\left(\int_{-H}^{R+\zeta} c\,dz\right)_t+\frac{1}{B}\left(\int_{-H}^{R+\zeta} Buc\,dz\right)_z-\frac{1}{B}\left(\int_{-H}^{R+\zeta} BK_H c_x\,dz\right)_x=-D+E$$

Substituting this in (10.1) we find

$$\left(\underbrace{\beta a+\int_{-H}^{R+\zeta} c\,dz}_{=S}\right)_t=-\frac{1}{B}\left(B\int_{-H}^{R+\zeta} uc-K_H c_x\,dz\right)_x.$$

We will only consider the subtidal part of this equation, meaning that the left-hand side vanishes and we find

$$\left\langle B\int_{-H}^{R+\zeta} uc-K_H c_x\,dz\right\rangle_x=0,$$

where $\langle\cdot\rangle$ denotes the tidal-average. This equation requires one horizontal boundary condition, for which we describe that he advective and diffusive transport vanishes in time-average sense at the landward boundary, i.e.

$$\left\langle B\int_{-H}^{R+\zeta} uc-K_H c_x\right\rangle=0 \text{ at } x=L, \tag{10.2}$$

Therefore we can integrate the equation, use the boundary condition to find

$$\left\langle B\int_{-H}^{R+\zeta} uc-K_H c_x\,dz\right\rangle=0, \tag{10.3}$$

10.2 Scaling and ordering

Using the same scaling arguments as before we find that the equation vanishes at leading-order. At first-order we find

$$\left\langle B\int_{-H}^{R} u^0c^0\,dz\right\rangle_x. \tag{10.4}$$

It is assumed that the velocity u^0 only contains an M_2 component and the concentration c^0 only contains M_0 and M_4 components. In that case, the time-average of u^0c^0 vanishes, so that the first-order equation vanishes. Therefore the second-order equation is required as a closure condition (this is explained below). The second-order equation reads

$$\left\langle B\int_{-H}^{R} u^1c^0+u^0c^1-K_H c_x^0\,dz+B\zeta^0\,u^0c^0\big|_{z=R}\right\rangle=0. \tag{10.5}$$

In practice it is found that, near the landward end, the transport by the river flow is dominant and needs to be included to prevent sediment from accumulating near the boundary. Therefore we include the fourth-order transport terms $B\int_{-H}^{R} u_t^1 extriverc^2_{\text{river-river}} + K_H c^2_{\text{river-river, x}}\, dz$. By including this term and not any other fourth-order terms, we consciously break the scaling and consistency. Whenever this fourth-order river-induced transport term becomes the dominant mechanism explaining the sediment dynamics, the model results need to be treated with care.

The concentration computed in Section **??** depends on the erodability $f(a)$, see (9.29), (9.30) and (9.31). In the following we will assume $f = a$. Future versions of the model will also allow for other formulations. We then find that the solutions for the concentration become

$$\begin{aligned} c^0 &= \hat{\mathscr{C}}^0 a, \\ c^1 &= \hat{\mathscr{C}}^1_a a + \hat{\mathscr{C}}^1_{a_x} a_x, \\ c^2_{\text{river-river}} &= \hat{\mathscr{C}}^2_a a. \end{aligned}$$

The equation can then be rewritten to

$$\left\langle B\int_{-H}^{R} u^1\hat{\mathscr{C}}^0 a + u^0\hat{\mathscr{C}}^1_a a + u^0\hat{\mathscr{C}}^1_{a_x} a_x + u^1_{\text{river}}\hat{\mathscr{C}}^2_a a - K_H\left(\hat{\mathscr{C}}^0 a\right)_x - K_H\left(\hat{\mathscr{C}}^2 a\right)_x dz + B\zeta^0\, u^0\hat{\mathscr{C}}^0\big|_{z=R}\, a \right\rangle = 0.$$

Gathering terms that scale with a and a_x we find

$$\underbrace{\left\langle B\int_{-H}^{R} u^1\hat{\mathscr{C}}^0 + u^0\hat{\mathscr{C}}^1_a + u^1_{\text{river}}\hat{\mathscr{C}}^2_a - K_H\hat{\mathscr{C}}^0_x - K_H\hat{\mathscr{C}}^2_x\, dz + B\zeta^0\, u^0\hat{\mathscr{C}}^0\big|_{z=R} \right\rangle}_{=T} a + \underbrace{\left\langle u^0\hat{\mathscr{C}}^1_{a_x} - K_H\hat{\mathscr{C}}^0_x - K_H\hat{\mathscr{C}}^2_x \right\rangle}_{=F} a_x = 0 \tag{10.6}$$

In short this reads

$$Ta + Fa_x = 0. \tag{10.7}$$

This expression still requires a boundary condition. We will follow Chernetsky et al. (2010) and use

$$a_* = \frac{\int_0^L Ba\, dx}{\int_0^L B\, dx}, \tag{10.8}$$

where a_* is the domain-average availability. This availability needs to be prescribed externally.

10.3 Solution method

The subtidal closure equation (10.7) is a first-order differential equation in a. The general solution to this reads

$$a = ke^{-\int_0^x \frac{T}{F}\, dx'}.$$

The constant of integration k follows from the boundary condition

$$k = \frac{a_* \int_0^L B\, dx}{\int_0^L Be^{-\int_0^x \frac{T}{F}\, dx'}\, dx}, \tag{10.9}$$

The transport terms T and F can be decomposed into several contributions related to the forcing of the velocity or concentration. Details on this decomposition are discussed in the manual of the package `semi_analytical` (Brouwer, 2017).

Bibliography

Brouwer, R. L. (2017). *Semi-analytical 2DV perturbation model. Package for iFlow.*

Brouwer, R. L., Schramkowski, G. P., de Swart, H. E., de Mulder, T., Verwaest, T., and Mostaert, F. (2016a). Geïdealiseerde proceprocess van systeemovergangen naar hypertroebelheid. WP 1.5 Niet-lineair sedimsediment. Technical report, Waterbouwkundig Laboratorium Borgerhout/Flanders Hydraulics Research, Antwerp. In Dutch.

Brouwer, R. L., Schramkowski, G. P., Verwaest, T., and Mostaert, F. (2016b). Geïdealiseerde proceprocess van systeemovergangen naar hypertroebelheid. WP 1.4 Basismodel sediment. Technical report, Waterbouwkundig Laboratorium Borgerhout/Flanders Hydraulics Research, Antwerp. In Dutch.

Chernetsky, A. S. (2012). *Trapping of sediment in tidal estuaries.* PhD thesis, TU Delft.

Chernetsky, A. S., Schuttelaars, H. M., and Talke, S. A. (2010). The effect of tidal asymmetry and temporal settling lag on sediment trapping in tidal estuaries. *Ocean Dynamics*, 60:1219–1241.

Dijkstra, Y. M. (2017). *iFlow modelling framework. User manual & technical description.*

Dijkstra, Y. M., Brouwer, R. L., Schuttelaars, H. M., and Schramkowski, G. P. (Manuscript submitted to Geoscientific Model Development). The iFlow Modelling Framework v2.4. A modular idealised process-based model for flow and transport in estuaries.

Ensing, E., De Swart, H. E., and Schuttelaars, H. M. (Manuscript in preparation). The role of flow-turbidity ffeedback in highly turbid estuaries.

Ianniello, J. P. (1977). Tidally induced residual currents in estuaries of constant breadth and depth. *Journal of Marine Research*, 35:755–786.

Ianniello, J. P. (1979). Tidally induced residual currents in estuaries of variable breadth and depth. *Journal of Physical Oceanography*, 9:962–974.

Kandiah, A. (1974). *Fundamental aspects of surface erosion of cohesive soils.* PhD thesis, University of California, Davis.

McCarthy, R. K. (1993). Residual currents in tidally dominated, well-mixed estuaries. *Tellus*, 45A:325–340.

Wei, X., Schramkowski, G. P., and Schuttelaars, H. M. (2016). Salt dynamics in well-mixed estuaries: Importance of advection by tides. *Journal of Physical Oceanography*, 46:1457–1475.