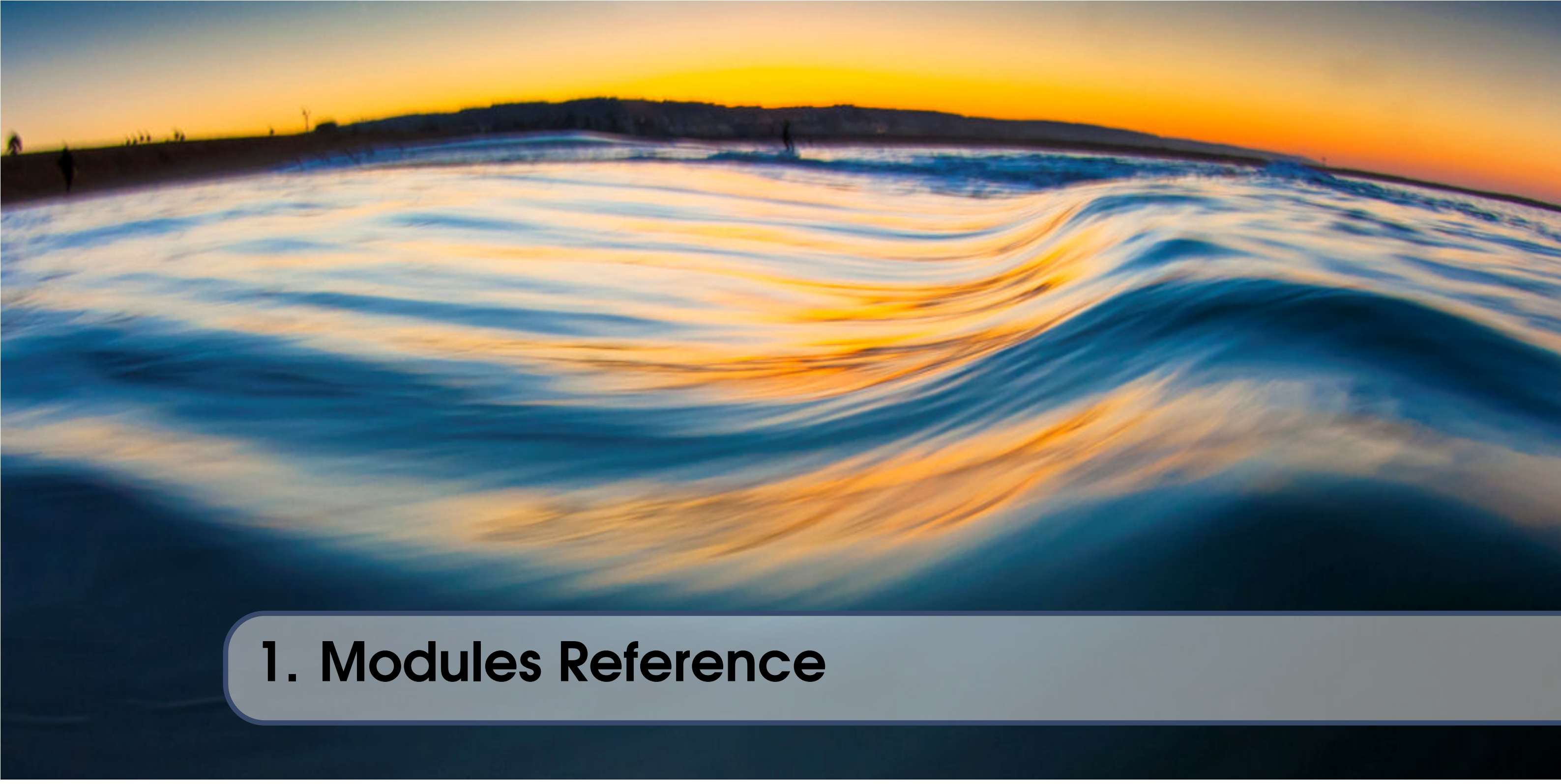
Analytical 2DV package for salinity and turbulence modules

package for iFlow

Yoeri Dijkstra

When using iFlow, please cite 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.
Additionally you may refer to this manual as Dijkstra, Y. M. (2017). *iFlow modelling framework. User manual & technical description*.

Contents



1. Modules Reference

This chapter provides a short overview of all modules in the package `analytical2DV` 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)

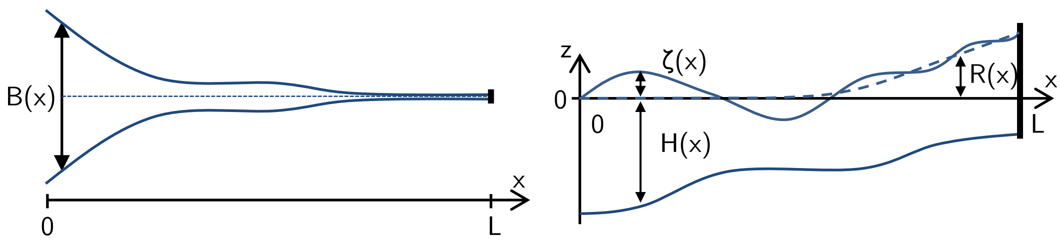
The cells with input variables have been colour-coded to indicate whether the variable is likely to be given in the input file, computed by another module or given in the configuration file. By the very nature of iFlow this is only indicative and depends on the modules used. As an example, almost any 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 Geometry

1.1.1 Geometry2DV

The model domain is a two-dimensional width-averaged area as sketched in Figure 1.1. 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 $z = 0$.



(a) Top view (b) Side view

Figure 1.1: 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.

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.

The module Geometry2DV sets the length L, width B and depth H (without reference level R). The input for B and H is best illustrated using an example

■ Code sample 1.1

```
module  analytical2DV.Geometry2DV
L       160000
B0      type    functions.ExpRationalFunc
        C1      -0.027e-3 1.9
        C2      5.0e-11 -9.2e-6 1.0
H0      type    functions.Polynomial
        C       -2.9e-24 1.4e-18 -2.4e-13 1.7e-8 -5.2e-4 15.3
```

■

The width and depth are set by functions. Below the function specification follow the arguments for that function, preceded by at least one space or tab. Functions are specified in this input as if they are modules, in the form (package name).(function name). Standard functions in iFlow are specified in the package `functions`. These functions, for a variable $x \in [0, L]$, are

Constant		Constant function $f = C0$
	input	`C0`: number
Exponential		Exponential function $f = C0e^{-x/Lc}$
	input	`C0`: number
		`Lc`: number
ExpRationalFunc		Exponent of a ratio of two polynomials $f = 1000e^{\text{polyval}(C1,x)/\text{polyval}(C2,x)}$
	input	`C1`: Space-separated numbers representing polynomial coefficients
		`C2`: Space-separated numbers representing polynomial coefficients
HyperbolicTangent		Hyperbolic tangent function $f = C0 + C1\tanh\left(\frac{x-xc}{xl}\right)$
	input	`C0`: Number
		`C1`: Number
		`xc`: Number
		`xl`: Number
Linear		Linear function $f = C0\frac{L-x}{L} + CL\frac{x}{L}$
	input	`C0`: Number
		`CL`: Number
Polynomial		Polynomial function $f = \text{polyval}(C,x)$
	input	`C`: Space-separated numbers representing polynomial coefficients
PolynomialLinear		Combination of a polynomial and linear function. The linear function starts for $x > XL$ at the same level of the polynomial function there and with the same slope (i.e. continuous and once differentiable): $f = \begin{cases} \text{polyval}(C,x) & \text{if } x < XL \\ \text{polyval}(C,XL) + \text{polyder}(C,XL)(x-XL) & \text{if } x > XL \end{cases}$
	input	`C`: Space-separated numbers representing polynomial coefficients
		`XL`: Number

Although momentarily not included in iFlow, one can easily construct a function that retrieves numerical data of the depth or width from a file. This allows Geometry2DV to work with numerical data as well.

Module reference table:

Type		Normal
Submodules		None
Input	`L`	*Number*. Length of the system (in m).
	`B0`	*Function*. Width, see example above.
	`H0`	*Function*. Depth measured as the distance between the zero-reference (i.e. mean water level at the mouth) and the bed. See example input above
Output	`L`	*Number*. Length of the system (in m).
	`B0`	*Function*. Width, see example above.

1.2 Sediment

1.2.1 SedDynamicLead

Leading-order sediment model, see Chapter **??**.

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`.

1.2.2 SedDynamicFirst

First-order sediment model, see Chapter **??**.

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 $c^1 = \hat{c}^1_a a + \hat{c}^1_{a_x} a_x$. NB. this output variable consists of a main index `hatc1` and sub-indices `a` and `ax`.

1.2.3 SedDynamicSecond

Second-order sediment model, see Chapter **??**.

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`.

1.2.4 StaticAvailability

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

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).

1.3 Salinity

1.3.1 SaltExponential

Diagnostic (i.e. prescribed) along-channel salinity profile. The vertical is assumed to be fully mixed and the signal is assumed not to vary over the tidal time scale. The salinity profile follows an exponential profile of the form

$$s^0 = s_{\text{sea}} \exp\left(-\frac{x}{L_s}\right). \tag{1.1}$$

Type		Normal
Submodules		None
Input	ssea	*Number.* Salinity (in psu) at the seaward boundary $x = 0$.
	Ls	*Number.* Length-scale for salinity decay (in metres).
	L	*Number.* Length of the system (in metres). Value is output of the geometry module, but can also be prescribed in the input file.
Output	s0	*Analytical function 1-dimensional.* Leading-order salinity profile in x-direction.

1.3.2 SaltHyperbolicTangent

Diagnostic (i.e. prescribed) along-channel salinity profile. The vertical is assumed to be fully mixed and the signal is assumed not to vary over the tidal time scale. The salinity profile follows a hyperbolic tangent profile of the form (see also Warner et al. (2005); Talke et al. (2009))

$$s = \frac{s_{\text{sea}}}{2}\left(1 - \tanh\left(\frac{x - x_c}{x_L}\right)\right) \tag{1.2}$$

Type		Normal
Submodules		None
Input	ssea	*Number.* Salinity (in psu) at the seaward boundary $x = 0$.
	xc	*Number.* Length-scale (in metres). Denotes the position of the salinity value $\frac{s_{\text{sea}}}{2}$.
	xl	*Number.* Length-scale (in metres). Denotes the width of the salinity profile.
	L	*Number.* Length of the system (in metres). Value is output of the geometry module, but can also be prescribed in the input file.
Output	s0	*Analytical function 1-dimensional.* Leading-order salinity profile in x-direction.

1.4 Two-parameter turbulence closures

The turbulence models compute the eddy viscosity and a roughness parameter belonging to the relevant boundary condition for the momentum equation.

1.4.1 Uniform

Eddy viscosity with uniform value in the vertical direction. The fitting boundary condition to the momentum equation is the partial slip condition. The eddy viscosity and partial slip roughness parameter can vary with the along-channel dimension as

$$A_v(x,f) = A_{v0}(f)\left(\frac{H(x)+R(x)}{H(0)}\right)^m,$$
$$s_f(x) = s_{f,0}\left(\frac{H(x)+R(x)}{H(0)}\right)^n$$

The input $A_{v0}(f)$ is allowed to be a function of time via the frequency dimension.

Type		Normal
Submodules		None
Input	`Av0amp`	*Space-separated numbers.* Leading-order reference eddy viscosity amplitude $\lvert A_{v0}\rvert$ (in m^2/s). The first value corresponds to subtidal. 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).
	`Av0phase`	*Space-separated numbers.* Leading-order reference eddy viscosity phase $\phi(A_{v0})$ (in deg). Input has the same structure as the amplitude. Note that the first element should equal zero.
	`sf0`	*Number.* Subtidal reference partial slip parameter $s_{f,0}$.
	`m`	*Number.* Depth-dependency parameter for A_v, see equations.
	`n`	*Number.* Depth-dependency parameter for s_f, see equations.
	`grid`	*iFlow grid.*
Output	`Av`	*Function 3-dimensional.* Leading-order eddy viscosity (in m^2/s). Function of x and f (length 1 in z dimension).
	`Roughness`	*Function 1-dimensional.* Partial slip parameter s_f (in m/s).
	`BottomBC`	*String* equal to `'PartialSlip'`. Indicates bottom boundary condition for the momentum equation.

1.4.2 Parabolic

Eddy viscosity with uniform value in the vertical direction. The fitting boundary condition to the momentum equation is the partial slip condition. The eddy viscosity and partial slip roughness parameter can vary with the along-channel dimension as

$$A_v(x,f) = A_{v0}(f)\,(z_s^* + \hat{z})\,(1 + z_0^* - \hat{z})\left(\frac{H(x)+R(x)}{H(0)}\right)^m,$$
$$z_0^*(x) = z_{00}^*\left(\frac{H(x)+R(x)}{H(0)}\right)^n,$$

where $\hat{z}$ is a dimensionless vertical axis between 0 (surface) and 1 (bed), z_0^* is a dimensionless bottom roughness equal to $z_0(x)/H(x)$ and z_s^* is a dimensionless surface roughness such that the subtidal eddy viscosity at the surface equals 10^{-6} m^2/s. The input $A_{v0}(f)$ is allowed to be a function of time via the frequency dimension and has dimension m^2/s.

Type		Normal
Submodules		None
Input	`Av0amp`	*Space-separated numbers.* Leading-order reference eddy viscosity amplitude $\|A_{v0}\|$ (in m^2/s), see Uniform module.
	`Av0phase`	*Space-separated numbers.* Leading-order reference eddy viscosity phase $\phi(A_{v0})$ (in deg), see Uniform module.
	`z0*`	*Number.* Subtidal reference dimensionless roughness height z_{00}^*.
	`m`	*Number.* Depth-dependency parameter for A_v, see equations.
	`n`	*Number.* Depth-dependency parameter for z_0^*, see equations.
	`grid`	*iFlow grid.*
Output	`Av`	*Function 3-dimensional.* Leading-order eddy viscosity (in m^2/s). Function of x, z and f.
	`Roughness`	*Function 1-dimensional.* Roughness height z_0 (N.B. not dimensionless) (in m).
	`BottomBC`	*String* equal to `'NoSlip'`. Indicates bottom boundary condition for the momentum equation.
Output		

1.5 $k-\varepsilon$ fitted turbulence closures

Set of turbulence closures that consist of algebraic equations that are fitted to the results of a $k-\varepsilon$ turbulence model. See also Chapter 2. Summarising the relations. The model uses the relation

$$A_v = 0.49 s_f (H + R + \zeta),$$

if `roughnessParameter` is set to `'sf0'` and uses

$$A_v = \frac{0.10}{0.636}\kappa^{-2} C_D |u| (H + R + \zeta), \tag{1.3}$$

$$s_f = \frac{0.22}{0.636}\kappa^{-2} C_D |u|, \tag{1.4}$$

if `roughnessParameter` is set to `'z0*'`. Here κ is the Von Karman constant of 0.4 and C_D equals

$$C_D = \left(\frac{U_*}{U}\right)^2 = \kappa^2 \left[(1 + z_0^*) \ln\left(\frac{1}{z_0^*} + 1\right) - 1 \right]^{-2}.$$

The modules `KEFittedLead`, `KEFittedFirst` and `KEFittedTruncated` use these equations in a scaling approach, while the module `KEFittedTruncated` uses a truncation approach.

The roughness parameter r, i.e. s_f or z_0^*, may vary with x related to the depth according to

$$r(x) = r_0 \left(\frac{H(x) + R(x)}{H(0)}\right)^n.$$

1.5.1 KEFittedLead

Leading-order of the above equations, i.e

$$A_v^0 = 0.49 s_f (H+R),$$

or

$$A_v^0 = \frac{0.10}{0.636}\kappa^{-2}C_D|u|^0(H+R),$$
$$s_f^0 = \frac{0.22}{0.636}\kappa^{-2}C_D|u|^0.$$

Type		Iterative
Submodules		None
InputInit	`roughnessParameter`	*String*. Indicates what roughness parameter to use. May have values `sf0` or `z0*`.
	`@roughnessParameter` i.e. `sf0` or `z0*`	*Number*. Value of $s_{f,0}$ or z_{00}^*.
	`n`	*Number*. Depth-dependency parameter for the roughness parameter of choice, see above equations.
	`Avmin`	*Number*. Minimum value for the subtidal eddy viscosity (divided by the depth). The actual subtidal eddy viscosity equals the maximum of the computed A_v and $A_{v\mathrm{min}}(H+R)$.
	`lambda`	*Number between 0 and 1*. Fraction of time-dependency to account for. A value $\lambda = 0$ eliminates all computed time-dependency, while $\lambda = 1$ includes the full computed time-dependency. Values between 0 and 1 indicate that the computed time-dependence is only partially accounted for.
	`referenceLevel`	*Boolean, optional*. Include a reference level computation in the module. This allows for omitting the module ReferenceLevel, for a more optimised iteration loop.
	`ignoreSubmodule`	*Space-separated strings*. Names of submodules of the hydrodynamics (i.e. u) that may be ignored in the computation. Only relevant to leading-order if `z0*` is used as roughness parameter.
	`profile`	*String*. Vertical profile. Currently only `uniform` is allowed.
	`Q0, Q1`	*Numbers*. Only required if `referenceLevel` equals `True`. Leading- and first-order discharge (in m^3/s). Both are required, but only the first-order discharge is only used if $Q_0 = 0$.
	`H`	*General 1-dimensional*. Depth (in m).
	`B`	*General 1-dimensional*. Width (in m).
	`grid`	*iFlow grid*. Only required in initial run if `referenceLevel` equals False.
	`G`	*Number*. Gravitational acceleration (in m/s^2).
Input	`grid`	*iFlow grid*.
	`u0`	*General 3-dimensional*. Leading-order horizontal flow velocity.

Output	`Av`	*Array 3-dimensional.* Leading-order eddy viscosity (in m^2/s). Function of x and f (length 1 in z).
	`Roughness`	*Array 1-dimensional.* Roughness parameter, depending on choice in input.
	`BottomBC`	*String* equal to `'PartialSlip'`. Indicates bottom boundary condition for the momentum equation.

1.5.2 KEFittedFirst

First-order of the above equations, i.e

$$A_v^1 = 0.49 s_f \zeta^1,$$

or

$$A_v^1 = \frac{0.10}{0.636}\kappa^{-2} C_D \left(|u|^1 (H+R) + |u|^0 \zeta^0\right),$$
$$s_f^1 = \frac{0.22}{0.636}\kappa^{-2} C_D |u|^1.$$

Type		Iterative
Submodules		None
InputInit		Same as KEFittedLead, except for `Avmin`, `ReferenceLevel`, `B`, `Q0` and `Q1`.
	`u0`	*General 3-dimensional.* Leading-order horizontal flow velocity.
	`grid`	*iFlow grid.*
Input	`zeta0`	*General 3-dimensional, length 1 in z direction.* Leading-order water level elevation (in m)
	`u1`	*General 3-dimensional.* First-order horizontal flow velocity.
Output	`Av1`	*Array 3-dimensional.* First-order eddy viscosity (in m^2/s). Function of x and f (length 1 in z).
	`Roughness1`	*Array 1-dimensional.* First-order roughness parameter, depending on choice in input.

1.5.3 KEFittedHigher

Higher-order ($n > 1$) of the above equations, i.e

$$A_v^n = 0.49 s_f \zeta^n,$$

or

$$A_v^n = \frac{0.10}{0.636}\kappa^{-2} C_D \left(\sum_{p=0}^{n-1} \left(|u|^n \zeta^{n-1-p}\right) + |u|^n (H+R)\right),$$
$$s_f^n = \frac{0.22}{0.636}\kappa^{-2} C_D |u|^n.$$

May be used in combination with module HigherOrderIterator.

Type		Iterative

Submodules		None
InputInit		Same as KEFittedFirst.
	zeta0	*General 3-dimensional, length 1 in z direction*. Leading-order water level elevation (in m)
	u1	*General 3-dimensional*. First-order horizontal flow velocity.
Input	maxOrder	*Integer*. Maximum order. May be passed by the module HigherOrderIterator.
	maxOrder	*Integer*. Current order. May be passed by the module HigherOrderIterator
	u+{0,@{maxOrder}+1}	*General 3-dimensional*. Higher-order horizontal flow velocity.
	zeta +{0,@{maxOrder}+1}	*General 3-dimensional, length 1 in z direction*. Higher-order water level elevation.
Output	Av+{2,@{maxOrder}+1}	*Array 3-dimensional*. Higher-order eddy viscosity (in m^2/s). Function of x and f (length 1 in z).
	Roughness +{2,@{maxOrder}+1}	*Array 1-dimensional*. Higher-order roughness parameter, depending on choice in input.

1.5.4 KEFittedTruncated

Solve the equations in full up to order `truncationOrder` (inclusive).

Type		Iterative
Submodules		None
InputInit		Same as KEFittedLead.
	truncationOrder	*Integer*. Truncate after this order (inclusive).
Input	u+{0, @{truncationOrder}+1}	*General 3-dimensional*. Horizontal flow velocity.
	zeta+{0, @{truncationOrder}+1}	*General 3-dimensional, length 1 in z direction*. Water level elevation.
Output	Av	*Array 3-dimensional*. Leading-order eddy viscosity (in m^2/s). Function of x and f (length 1 in z).
	Roughness	*Array 1-dimensional*. Roughness parameter, depending on choice in input.
	BottomBC	*String* equal to 'PartialSlip'. Indicates bottom boundary condition for the momentum equation.

2. $k-\varepsilon$ fitted closures

The $k-\varepsilon$ turbulence model is the state-of-the-art for 1DV, 2DV or 3D models of estuaries. This model is however highly non-linear and has only been tested in time-stepping methods. Idealised modelling methods that solve for harmonic components are therefore not directly compatible with the $k-\varepsilon$ model. Instead, such idealised models often settle with much simpler turbulence closures, where the eddy viscosity profile is typically assumed to be either vertically uniform or parabolic. This assumption on the eddy viscosity profile is probably not very restricting in cases of mild stratification. A more influential assumption in these models concerns the lack of a relation to the depth or flow. Also, the simplified turbulence closures typically depend on two fit-parameters. Often there is a band of parameter values with more-or-less equivalently accurate results. This property threatens to undermine the reliability of these turbulence models when it comes to modelling salt or sediment.

The goal of the `KEFitted` turbulence models is to resolve the above identified problems of idealised turbulence models concerning depth-flow-dependency and multiple equivalent parameter setting. This is done by fitting the idealised uniform profiles to solutions of the $k-\varepsilon$ model. Here, we restrict our attention to barotropic flows. This fitting procedure is done for a subtidal eddy viscosity only. Additionally the fitting procedure reduces the number of fit parameters to one.

2.1 Models and fitting conditions

2.1.1 Models and parameters

The idealised models that will be considered assume a uniform eddy viscosity profile, described as

$$A_v(x,z) = A_{v,0}(x),$$

which is accompanied by a partial-slip boundary condition for the momentum equation:

$$A_v u_z(x,-H) = s_f(x)u(x,-H) \qquad \text{(partial slip)}.$$

The parameters in this model are thus $A_{v,0}(x)$ and $s_f(x)$. The time-dependence of $A_{v,0}$ is allowed to consist of harmonic components with a period equal to the M_2 tide and its overtides. The components of $A_{v,0}$ will be denoted by $A_{v,0n}$, where $n=0$ corresponds to the subtidal component, $n=1$ to the M_2, $n=2$ to the M_4 etc.

The idealised model is fitted to the $k-\varepsilon$ model. This model can be described in abstract notation and without buoyancy as

$$A_v = f(u_z, H, z_0).$$

The fit is performed using the water column (i.e. 1DV) model as described by Dijkstra et al. (2016). The model is forced by a prescribed depth-averaged velocity U with one or several tidal components and a constant river discharge.

2.1.2 Fitting conditions

The idealised model is fitted to the $k-\varepsilon$ model using fit conditions. The required number of conditions depends on the number of parameters. Since we only consider a subtidal eddy viscosity, the model has two parameters $A_{v,0}$ and s_f. These are obtained by matching the amplitude and phase of the M_2 tidal water level gradient ζ_x obtained using the simple turbulence model to that obtained with the $k-\varepsilon$ model. Other fit conditions one could think of are the (subtidal) turbulent energy dissipation and the bed shear-stress. In Appendix B we will show that these conditions are automatically satisfied when fitting the water level gradients.

2.1.3 Regression

The $k-\varepsilon$ model depends on the depth-averaged velocity amplitude U (through u_z), the depth H and the roughness height z_0. This dependency is incorporated into the idealised models by using a non-linear regression on the fitted results. From a combination of theory and experimentation with different formulations for the regression, the following follows as the best:

$$\gamma_1 U^{\gamma_2}\left[\left(1+\frac{z_0}{H}\right)\ln\left(\frac{H}{z_0}+1\right)-1\right]^{\gamma_3} z_0^{\gamma_4} H^{\gamma_5}.$$

Equivalently this can be written as

$$\gamma_1 U^{\gamma_2}\left(\kappa^{-2}C_D\right)^{\gamma_3/2} z_0^{\gamma_4} H^{\gamma_5},$$

where C_D is a drag coefficient as specified by Burchard et al. (2011) as

$$C_D = \kappa^2\left[\left(1+\frac{z_0}{H}\right)\ln\left(\frac{H}{z_0}+1\right)-1\right]^{-2}.$$

The parameters γ_n, $n=1,\ldots,5$ are the regression parameters.

2.2 Fitting to M_2 tidal flows

For the first case we impose a simple single constituent M_2 flow. The models have been tested with a wide range of the parameters U, z_0 and H in order to have trustworthy results for the regression. The tested parameter values are given below

M_2 tide only	U M_2	0.2, 0.4, 0.6, 0.8, 1.0 m/s
	z_0	0.1, 0.01, 0.001, 0.0001, 0.00001 m
	H	6, 8, 10, 12, 14, 16, 18, 20 m

All permutations of settings are tested, so that there are 200 cases. The fitting conditions have been applied to find the parameters $A_{v,00}$ (i.e. only subtidal) and s_f for each of the 200 parameter settings and forcing only by the M_2 tide. A fit has been found for all of these 200 simulations. A non-linear regression is applied to the results of the fitted cases. Figure 2.1 shows the best regressive fit of $A_{v,00}$ and s_f.

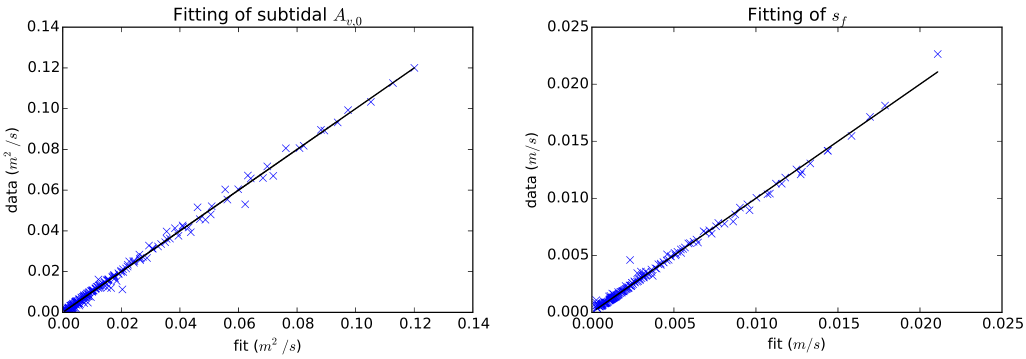


Figure 2.1

The fitted expressions read:

$$A_{v,0} = 0.09U^{1.1}\left[\left(1+\frac{z_0}{H}\right)\ln\left(\frac{H}{z_0}+1\right)-1\right]^{-1.8} z_0^{0.053}H^{1.0},$$
$$s_f = 0.20U^{0.98}\left[\left(1+\frac{z_0}{H}\right)\ln\left(\frac{H}{z_0}+1\right)-1\right]^{-1.9} z_0^{-0.0011}H^{-0.040}.$$

These fitting relations are simplified somewhat by rounding the powers. After rounding the powers, the factor γ_1 in front of the relation is refitted to arrive at the following simplified relations:

$$A_{v,0} = 0.10U\left[\left(1+\frac{z_0}{H}\right)\ln\left(\frac{H}{z_0}+1\right)-1\right]^{-2} H, \tag{2.1}$$
$$s_f = 0.22U\left[\left(1+\frac{z_0}{H}\right)\ln\left(\frac{H}{z_0}+1\right)-1\right]^{-2}. \tag{2.2}$$

Alternative to relating $A_{v,0}$ and s_f to z_0 as done above, we can also eliminate z_0 and relate $A_{v,0}$ to s_f. Simply rewriting (2.1) and (2.2) yields $A_{v,0} = 0.45s_fH$. However, a more accurate result is found by making a new fit using s_f in place of z_0. The results are presented in Figure 2.2a and the regression formula below.

$$A_{v,0} = 0.60U^{0.16}s_f^{1.1}H^{1.1}.$$

As the dependency on U is only weak, we choose to eliminate this dependency altogether. We round the powers and fit the factor in front of the expression again to arrive at

$$A_{v,0} = 0.49s_fH. \tag{2.3}$$

This relation between $A_{v,0}$ and s_f is plotted in Figure 2.2b together. Even though the relation is simple, it fits the data points quite well.

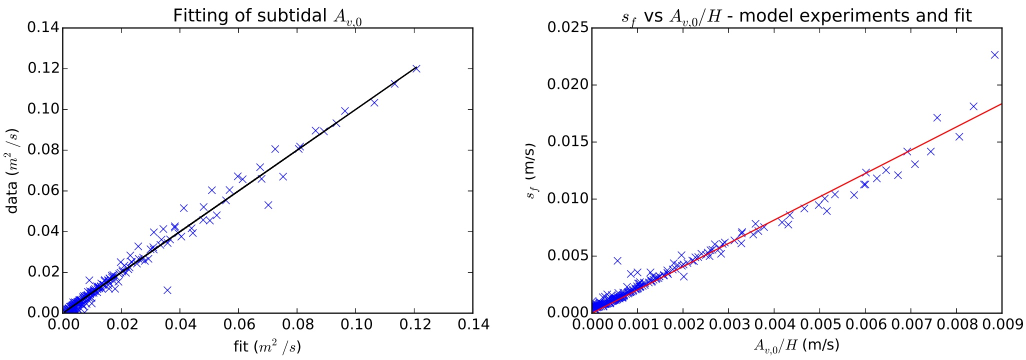


Figure 2.2

2.3 General formulation

It is found that $A_{v,0}$ can be expressed as

$$A_{v,0} = \gamma_1 U H \left[\left(1 + \frac{z_0}{H}\right) \ln\left(\frac{H}{z_0} + 1\right) - 1 \right]^{-2}.$$

This relation can be interpreted using scaling arguments and literature. A scaling of the $k-\varepsilon$ model (see Appendix A) reveals that the eddy viscosity should scale with $|u_*|H$, i.e. with the absolute value of the bed friction velocity multiplied by the local depth. The bed friction velocity can be related to the velocity through a shape function, which depends on the roughness height and depth. Letting $f(z_0, H)$ be this shape factor, we have $A_v \sim |u|Hf(z_0,H)$.

The shape factor should follow from the relation between the depth-averaged velocity and bed friction velocity. Such a relation follows from the logarithmic velocity profile (Burchard et al., 2011)

$$C_D = \left(\frac{U_*}{U}\right)^2 = \kappa^2 \left[\left(1 + \frac{z_0}{H}\right) \ln\left(\frac{H}{z_0} + 1\right) - 1 \right]^{-2}.$$

Our results show that the absolute value $|u_*|/|u|$ scales dominantly with $\left(\frac{U_*}{U}\right)^2$.

Following the scaling relation we will assume that the dependency on U can be generalised to a dependence on $|u|$. Note that U and the subtidal part of $|u|$ are related as $\langle|u|\rangle = 0.636U$ (where $\langle\cdot\rangle$ denotes time-averaging) for a flow with only a single harmonic components and no residual flow. Additionally, the eddy viscosity scales with the depth, which is fixed at H in the water column module, but in the width-averaged model reads $H+R+\zeta$. We will thus write our result as

$$A_{v,0} = \frac{\gamma_1}{0.636} \kappa^{-2} C_D \langle |u|(H+R+\zeta) \rangle.$$

A similar relation was found for s_f, with the general form

$$s_f = \frac{\gamma_1}{0.636} \kappa^{-2} C_D \langle |u| \rangle,$$

which is a form consistent with the derivation by Zimmerman (1982).

2.4 Approximating |u| for computations with general flows

The temporal variation of the eddy viscosity parameter $A_{v,0}$ is generated by the absolute value of the velocity times the depth $|u|(H+R+\zeta)$, while the partial slip parameter scales with $|u|$. Using this notion, the model derived for pure M_2 tidal flows can be extended to flows with combined subtidal, M_2, M_4 and higher overtidal flows. These (sub)tidal components will generate a signal of $|u|$ and $|u|(H+R+\zeta)$ that can be approximated using a subtidal part and tidal components. Here we will only consider the subtidal components $\langle|u|\rangle$ and $\langle|u|(H+R+\zeta)\rangle$, i.e. the subtidal value of $|u|$ and $|u|(H+R+\zeta)$ resulting from a combined subtidal and multi-frequency tidal flow.

However, it is not directly clear how to derive the subtidal part of these variables. One way would be to convert the harmonic components of the velocity and depth to a time series, taking the absolute value and computing the mean. However, this is an indirect technique inconsistent with the approach of using harmonic components and does not lead to unambiguous results if ordering is used. Therefore we make a Chebyshev polynomial expansion of $|u|$. This expansion yields, for the subtidal component,

$$\begin{aligned}\langle|u|\rangle &= \left\langle a_0 + a_2u^2 + a_4u^4 + \ldots \right\rangle, \\ \langle|u|(H+R+\zeta)\rangle &= \left\langle (a_0 + a_2u^2 + a_4u^4 + \ldots)(H+R+\zeta) \right\rangle,\end{aligned}$$

where $\langle\cdot\rangle$ denotes the tidal average and a_i are coefficients that follow from de expansion and depend on the number of components taken into account. In a demonstration below we will take all components up to a_4 into account. In this case the coefficient values are 0.127, 1.527 and -0.679. The above expansion is helpful, because it is clear how to compute the subtidal contribution of the product of two or more harmonic components (using the `NiFTy` tool `complexAmplitudeProduct`). Additionally, the above expression can be used to make an unambiguous ordering in the velocity (demonstrated below).

The results can be exemplified analytically for the combination of tidal M_2 and river flow, without other flow components. Let U_{M_2} be the depth-averaged M_2 velocity amplitude and U_{riv} be the depth-averaged river velocity amplitude. Additionally let α be defined as

$$\alpha = \frac{U_{\text{riv}}}{U_{M_2}}.$$

We then find for $|u|$

$$\begin{aligned}|u|_0 =& (1+\alpha)U_{M_2}\left(a_0 + a_2\left(\frac{\alpha}{1+\alpha}\right)^2 + \frac{1}{2}a_2\left(\frac{1}{1+\alpha}\right)^2 + a_4\left(\frac{\alpha}{1+\alpha}\right)^4\right. \\ &\left. + \frac{3}{8}a_4\left(\frac{1}{1+\alpha}\right)^4 + \frac{6}{2}a_4\frac{\alpha^2}{(1+\alpha)^4}\right).\end{aligned}$$

This expression conveniently describes the magnitude of $|u|$ in terms of the relative importance of the river flow. The limit values for the above expressions are

	only tide ($\alpha = 0$)	only river ($\alpha = \infty$)
$\lvert u\rvert_0$	$0.636U_{M_2}$	U_{riv}

The above analytical expression is useful to explore a deeper understanding of the effect a flow has on $|u|$ and therefore on the eddy viscosity and partial slip parameters via the turbulence model. Analyses like this can also be done for combinations of two tidal flows with different frequencies. It is outside the scope of this manual to further extend this analysis. Within the `KEFitted` modules, the Chebyshev expansion is used directly up to the a_8 component.

2.5 Ordering of velocity and depth

The fitted turbulence model depends on the velocity and depth, which are order quantities in the standard iFlow hydrodynamic modules. The `KEFitted` turbulence models come in two forms: ordered (`KEFittedLead, KEFittedFirst, KEFittedHigher`), which use the ordering of the velocity and depth to compute an ordered eddy viscosity and partial slip parameter, and truncated (`KEFittedTruncated`), which adds all velocity and depth contribution and computes a single eddy viscosity and partial slip parameter.

The velocity enters the model through the absolute value. The ordering of this is computed through the Chebyshev polynomial approximation outlined above. For the leading and first orders (denoted by superscripts) this yields

$$|u|_0^0 = a_0 + a^2\left(u^0\right)^2 + a^4\left(u^0\right)^4 + \ldots,$$
$$|u|_0^1 = 2a^2u^0u^1 + 4a^4\left(u^0\right)^3u^1 + \ldots.$$

An automated script allows the ordered `KEFitted` models to compute all components up to a_8 and up to any order.

The ordering of the depth dependence is through a factor $H+R+\zeta$ and is simply governed by the ordering of ζ. Since ζ^0 is regarded as an order ε contribution relative to $H+R$, the leading-order depth equals $H+R$, the first-order depth equals $H+R+\zeta^0$ etcetera.

2.6 Summary of relations

The uniform model with partial slip boundary condition is only applied for an eddy viscosity and partial slip parameter that is constant in time. We find the following expressions for the parameters $A_{v,0}$ and s_f:

$$A_{v,0} = \frac{0.10}{0.636}\kappa^{-2}C_D\left\langle|u|(H+R+\zeta)\right\rangle, \tag{2.4}$$
$$s_f = \frac{0.22}{0.636}\kappa^{-2}C_D\left\langle|u|\right\rangle. \tag{2.5}$$

Here

$$C_D = \left(\frac{U_*}{U}\right)^2 = \kappa^2\left[\left(1+\frac{z_0}{H}\right)\ln\left(\frac{H}{z_0}+1\right)-1\right]^{-2}.$$

The two parameters can also be related to one-another, yielding

$$A_{v,0} = 0.49s_f(H+R+\zeta). \tag{2.6}$$

This relation produces a good fit with the data from the $k-\varepsilon$ and is considered to be accurate for the whole range of roughness values that might be encountered in estuaries that are not or only weakly stratified. The dependencies on $|u|$ and $H+R+\zeta$ are resolved either through ordering (`KEFittedLead`, `KEFittedFirst`, `KEFittedHigher`) or truncation (`KEFittedTruncated`). The subtidal part of $|u|$ and $|u|(H+R+\zeta)$ is computed through Chebyshev polynomials.

Bibliography

Burchard, H., Hetland, R. D., Schulz, E., and Schuttelaars, H. M. (2011). Drivers of residual estuarine circulation in tidally energetic estuaries: Straight and irrotational channels with parabolic cross section. *Journal of Physical Oceanography*, 41:548–570.

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.

Dijkstra, Y. M., Uittenbogaard, R. E., Van Kester, J. A. T. M., and Pietrzak, J. D. (2016). Improving the numerical accuracy of the $k-\varepsilon$ model by a transformation to the $k-\tau$ model. *Ocean Modelling*, 104:129–142.

Talke, S. A., De Swart, H. E., and De Jonge, V. N. (2009). An idealized model and systematic process study of oxygen depletion in highly turbid estuaries. *Estuaries and Coasts*, 32:602–620.

Warner, J. C., Geyer, W. R., and Lerczak, J. A. (2005). Numerical modeling of an estuary: A comprehensive skill assessment. *Journal of Geophysical Research*, 110(C05).

Zimmerman, J. T. F. (1982). On the lorentz linearization of a quadratically damped forced oscillator. *Physics Letters A*, 89A:123–124.