User's Guide WAQWND

SIMONA report number 2004-01

Rijkswaterstaat
Ministerie van Infrastructuur en Waterstaat

User's Guide WAQWND

Version number	:	Version 12.03, June 2019
Maintenance	:	see www.helpdeskwater.nl/waqua
Copyright	:	Rijkswaterstaat

Log-sheet

document version	date	Changes with respect to the previous version	
10.44	01-04-2014	m4050:	Simona2014 release
10.45	08-12-2014	m4198:	converted to LaTeX
10.46	09-12-2014	m4198:	added chapters Quick Reference and Technical background
10.47	19-12-2014	m4198:	added section cumulative stress
10.48	05-01-2015	m4198:	update figure 5.1
10.49	31-03-2015	m4181:	added description of the header of Ascii wind input
10.50	08-07-2015	beheer:	small corrections
10.51	05-10-2015	c3925:	replaced emos library by GRIB-API ; added section scanning mode
10.52	07-10-2015	m4311:	time zone read from NetCDF
12.01	13-09-2016	m4351:	correction wind2str input
12.02	16-05-2017	m4564:	support for space varying Charnock coefficients
12.03	19-06-2019	m4767:	support for time step < 1 h in Grib files

Contents

Chapter 1

General directions for the use of WAQWND

1.1 Background information

WAQWND is used to centralize all preprocessing of wind and (atmospheric) pressure data. WAQWND converts the information from KNMI ASCII-files, binary GRIB-files, binary NetCDF-file, Timeseries file or the so-called Space Varying Wind and Pressure (SVWP) file to the standard SIMONA format and stores this information on the 'wind' SDS file. Some additional information that is not present on the binary SVWP file (eg. grid size) is obtained from the user's input (via perl procedure parameters and windidfile) and stored on the 'wind' SDS file in order to complete the data. The information from the 'wind' SDS file is used by the WAQUA pre-processor and processor.

1.2 Input

WAQWND has the following input:

- - NetCDF-input, stored in 1 or more physical files
  - GRIB-input, distributed over 1 or more physical files
  - ASCII-input, distributed over 1 or more physical files
  - SVWP-input, stored in 1 physical file
  - Timeseries-input, stored in 1 physical file
- optional Wind2str-input, stored in 1 physical file
- WindID-input, stored in 1 physical file
- perl procedure-parameters

1.3 NetCDF-input

The binary NetCDF-format (network Common Data Form) is a widely used format for the exchange of multidimensional data. See www.unidata.ucar.edu for more information. The reading of NetCDF-files is done by means of the Fortran-90-NetCDF-library. It uses the CF-names for wind, stress and pressure:

- surface_downward_eastward_stress,
- surface_downward_northward_stress,
- eastward_wind,
- northward_wind,
- air_pressure,
- air_pressure_at_sea_level and
- surface_air_pressure.

It is only implemented for spherical grids, including spherical grids with a shifted South Pole.

All timesteps on the NetCDF file are converted to the Wind-SDS file. The gridflag and WindID-input is read from the attributes on the NetCDF-file, so the perl-procedure will not ask for these parameters.

The following fields in the NetCDF are also recoqnized:

eastward_ext_force	eastward external forcing
northward_ext_force	northward external forcing
charnock	space and time varying Charnock coefficients

1.4 GRIB-input

The binary GRIB-format is used for the exchange of data between meteorological institutes. The reading of GRIB-files is done by means of the GRIB-API, which is developed by the European Centre for Medium-Range Weather Forecast (ECMWF).

For a detailed description, see: software.ecmwf.int/wiki/display/GRIB.

The GRIB-API needs a definition file. This definition file is part of the Simona installation. Therefore, the Perl procedure waqwnd.pl sets the enviroment variable GRIB_DEFINITION_PATH to $SIMONADIR/extern/grib-api/definitions .

1.5 ASCII-input

The ASCII input is an "old" format, but still recoqnized by WAQWND .

1.6 SVWP-input

The SVWP-input contains the "old" binary format, formerly used for the exchange of data.

1.7 Timeseries/Wind2str-conversion-input

Consists of timeseries of global wind (constant in space), similar to the time series of global wind used in WAQPRE , and wind-to-stress conversion data. Physically the Timeseries and Wind2str input resides in separate files, due to the fact that the scope of Wind2str is not limited to Timeseries, but is also applicable to ASCII/GRIB/NetCDF/SVWP input.

1.8 WindID-input

Consists of up to 3 text lines to identify the wind field. The text is stored on the output SDS-file.

1.9 Perl procedure-parameters

Are used to identify input/outputfiles, to indicate whether wind-to-stress conversion must take place and to add missing information such as: the timezone of the input, the coordinate system used, the type of grid used, etcetera. For a detailed description of the perl procedure-parameters: refer to the Quick Reference Guide WAQWND , also to be found in Chapter 2.

1.10 Global functionality

WAQWND is able to process five different types of input:

- NetCDF: the binary format, used to exchange multidimensional data.
- GRIB: the binary format, used for the exchange of data between meteorological institutes.
- SVWP: "old" binary format
- ASCII: "old" ascii format, described in section 5.3
- Timeseries/Wind2str-conversion: timeseries of global wind (constant in space), similar to the time series of global wind used in WAQPRE , and wind-to-stress conversion data.

WAQWND may convert wind speeds to wind stresses (controlled by perl procedure parameter "convert2stress"). All methods implemented in WAQUA are possible (refer to User's Guide WAQPRE , section SPACE_VAR_WIND). The timezone (runprocedure parameter) for the input is added to the SDS-file. The scanning mode specified in the input for ASCII and GRIB is sustained. Extra info with respect to the coordinates-system (run-procedure parameters) is added to the SDS-file.

Chapter 2

Quick reference

2.1 Function

purpose The subsystem converts the information from KNMI ASCII-files, binary GRIB-files, binary NETCDF-files, TIMESERIES file or the so-called Space Varying Wind and Pressure (SVWP) file, delivered by the KNMI, to the standard SIMONA format and stores this information on the 'wind' SDS file. Some additional information that is not present on the binary SVWP file (eg. grid size) is obtained from the user's input (via the perl procedure parameters) and stored on the 'wind' SDS file in order to complete the data. The information from the 'wind' SDS file is used by the WAQUA pre-processor and processor.

2.2 Files

data flow	**Input files**	**Output files**
	- binary SVWP-file,	SIMONA data storage file
	- timeseries file	message print file
	- 1 or more ASCII/GRIB/NETCDF-file(s)	
file names	**Logical name**	**System name**
	SIMONA data storage file	SDS-<runid>
	message file	waqwnd-m.<runid>
file description	**Logical name**	**Purpose**
	- binary SVWP file, - ASCII-files, - a TIMESERIES-file, - GRIB-files - a NETCDF-file	Storage or description of Space Varying Wind and Pressure data delivered by the KNMI or made by user.
	SIMONA data-storage file	storage of permanent data in SIMONA
	message file	Contains error messages and diagnostics

test input Refer to the User's Guide WAQWND.

2.3 Run data

Format for Binary SVWP file

*waqwnd.pl [-inpfmt <inpfmt>] [-runid <runid>] [-exp <exp>] *
*[-KNMI <KNMI-file>][-stress_SVWP <Y/N>] *
*[-date_ref "<day> <month> <year>"] [-timzon <timzon>] *
*[-convert2stress <Y/N>] [-wnd2strfile <wnd2strfile>] *
*[-mmw <mmw>] [-nmw <nmw>] [-angle <angle>] *
*[-grid <P/S>] [-gridmode <gridmode>] *
*[-overwrite <overwrite>] [-windidfile <wind ident file>] *
[-coordsystem <coordsystem>] [-back <back>]

Format for ASCII and GRIB-file(s)

*waqwnd.pl [-inpfmt <inpfmt>] [-runid <runid>] [-exp <exp>] *
*[-KNMI <KNMI-file>] [-date_fc <date_fc>] *
*[-time <time>] [-timzon <timzon>] [-convert2stress <Y/N>] *
*[-wnd2strfile <wnd2strfile>] [-pgem <pgem>] [-stress <stress>] *
*[-overwrite <overwrite>] [-windidfile <wind ident file>] *
*[-coordsystem <coordsystem>] [-gridflag <gridflag>] *
[-date_unit <unit>] \[-back <back>]

Format for a NETCDF-file

*waqwnd.pl [-inpfmt <inpfmt>] [-runid <runid>] *
*[-KNMI <KNMI-file>] [-date_ref <date_ref>] *
*[-convert2stress <Y/N>] *
*[-wnd2strfile <wnd2strfile>] [-stress <stress>] *
[-overwrite <overwrite>] [-back <back>]

Format for a TIMESERIES file

*waqwnd.pl [-inpfmt <inpfmt>] [-runid <runid>] [-exp <exp>] *
*[-KNMI <KNMI-file>] [-timzon <timzon>] *
*[-wnd2strfile <wnd2strfile>] [-overwrite <overwrite>] *
[-windidfile <wind ident file>] [-back <back>]

parameters	**parameter**	**signification and value**
	-angle	angle of the Y-axis with the North-direction; default: 0.0
	-back	y(es): program will be started in the background n(o): program will be started in the foreground
	-convert2stress	Must waqwnd convert windspeeds from the input files into stresses

	(Y/N); default: N
-coordsystem	String with extra information on the coordinate system: USER: User defined (default); grid-type: Undefined INDEX: Model coordinates (M and N); grid-type: Undefined RDV: "Rijksdriehoeksstelsel-verschoven"; grid-type: Planar ED50: European Datum 1950; grid-type: Spherical WGS84: World Geodetic System 1984; grid-type: Spherical UTM31: Universal Transverse Mercator zone 31; grid-type: Planar UTM32: Universal Transverse Mercator zone 32; grid-type: Planar
-date_fc	Forecast date in format <*yyyymmdd*>; default: "current date". Input with reference time after "date_fc+time" is ignored.
-date_ref	reference date according to the WAQUA conventions (i.e. given in the form *dd mmm yyyy*, e.g. *15 SEP 1993*) The format is <yyyymmdd>; default: "current date" if input format is NETCDF.
-date_unit	Forecast unit; default: "h".
-exp	name of the experiment (simulation)
-inpfmt	Format of input file. Allowed values are SVWP (binary KNMI files), ASCII, GRIB, NETCDF or TIMESERIES
-grid	specification of grid type (either planar(P) or spherical(S))
-gridflag	Grid flag with values 1, 2 or 3 - 1: non-staggered grid - 2: auto detect (default) - 3: staggered grid
-gridmode	specification of grid mode (either staggered(S) or non-staggered(N))
-KNMI	- name of the binary space varying wind and pressure (SVWP) file delivered by KNMI; or - name of the TIMESERIES input file - name(s) of the ASCII/GRIB/NETCDF-file(s) with wind and pressure data (wildcard may be used)

	It can contain an explicit path name; the use of any indication of a parent directory ('..') is allowed.
-mmw	number of points in the rectangular wind grid in the M-direction
-nmw	number of points in the rectangular wind grid in the N-direction
-overwrite	an already existing experiment on the wind SDS file may be overwritten only if overwrite = yes is specified
-pgem	mean pressure in Pascal; (default: 101200 Pascal)
-runid	code to identify the output files of a run (eg. t02)
-stress	Indicates type of input to be used: -1 = autodetect (default) 0 = use wind speeds 1 = use wind stresses 2 = use cumm. wind stresses 3 = use cumm. wind stresses, times in between 4 = external forces
-stress_SVWP	Flag, indicating whether the KNMI file contains wind speeds (-stress_SVWP N) or wind stresses (-stress_SVWP Y); default: N
-time	time of start forecast in format <*hhmm*>; default: "current time" Input with reference time after "date_fc+time" is ignored.
-timzon	timezone GMT, MET or UNKNOWN; default: UNKNOWN
-windidfile	file containing text lines to identify the wind field (up to 3 lines allowed, maximal text length in each line: 80 characters); it can contain an explicit path name; the use of any indication of a parent directory ('..') is allowed.
-wnd2strfile	Name of input file for wind speed to stress conversion. Is only needed if -convert2stress== Y. if "-" is given default values will be used for the conversion.

- In case of a background run (with &) runid, exp, KNMI, date, mmw, nmw and grid are obligatory parameters and back has no effect.
- In case of an interactive start of the run the procedure will prompt for the parameters mentioned above (except those already given in the run call) including back. All parameters are

checked before the program is started.

- If an executable of the program (WAQWND.EXE) is present in the current directory, this executable will be used to run the program.

notes

Each set of space varying space and pressure data on the 'wind' SDS file is identified by means of an experiment name. More than one experiment on the 'wind' SDS file is allowed, unless the input format is NetCDF.

All the information required by the program WAQWND is passed to it by means of the run procedure.

A wind SDS file is different from a WAQUA SDS file created by WAQPRE and WAQPRO. It contains information which is additional to the information on a WAQUA SDS file.

If the wind grid must be rotated the value of angle should be nonzero. The wind grid will be then be rotated, counterclockwise when angle is positive and clockwise when angle is negative. However, the wind vectors are not rotated; they should be given - in the KNMI-file - , with reference to the coordinate system, in the definitive positions of the nodal points as if those were already rotated.

If planar grid is specified, the grid coordinates are expected to be given in meters.
If spherical grid is specified, the grid coordinates are expected to be given in degrees of longitude and latitude.

Chapter 3

Timeseries/Wind2str input description

3.1 General information

The input is based on SIMONA KEYWORD structure. Refer to "About SIMONA" in Section 1 "General Information".

Reminder: The input file is a structured ASCII-file. From the input file only the first 120 columns are read.

Reminder: The input file is a structured ASCII-file. From the input file only the first 258 columns are read.

Note: *If the last keyword block in the input file contains a sequential keyword, the* SIMONA *application independent preprocessor is not able to check the correctness of the block. This can result in incorrect processing of the input file!*

3.1.1 Conventions used

For the input definition the following conventions are used:

[val]	:	real value
[tval]	:	time specification in the form: *day hours:minutes* (e.g. 2 21:15). Times are given relative to midnight of a reference date, starting at 0 0:00.
[ival]	:	integer value
[iseq]	:	sequence number to indicate a point, curve, etc.
[text]	:	string (enclosed between quotes)
< . . . >	:	repetition group

| A
< : choice between A and B (A and B are mutually exclusive)

| B

& : continuation mark

In this document a part of the keywords is underlined (e.g, PRINT-OUTPUT). Only the underlined characters are significant. So the user must type at least PRINT in his input, but PRINTOUT is excepted as well.

The 'Explanation' part of the description of the various sections, subsections is divided in three columns:

Explanation:

KEYWORD	E	Explanation
		E can be O, M, D, S, R, X
	O	means keyword is optional
	M	means keyword is mandatory
	D	means keyword has a default value. When this keyword is omitted, the pre-processor will use the default value for the variable specified by means of this keyword
	S	means this keyword is a sequential keyword: a keyword followed by an integer (e.g. P4). A sequential keyword can be used repeatedly
	R	means keyword may occur more than once
	X	Exactly one of a series of keywords should be given

3.1.2 Data fields

Data field input is to be specified in two blocks:

SPACE_VARYING_DATA
GLOBAL
LOCAL

SPACE_VARYING_DATA stands for any keyword representing spatial data. In GLOBAL the data for the complete field is to be given, specifying function values at all grid points. In LOCAL however the user can specify rectangular boxes in which he can change the value of the space varying data. For the case of 3D this definition is extended in such a way that the input for separate layers is possible.

3.1.2.1 GLOBAL

Global data can be specified in two ways: first by giving one value for the complete computational grid, second by giving values for each grid point. The order in which these values are to be given is specified by the layout flag.

GLOBAL
LAYOUT = *[ival]*
| CONST_VALUE = *[val]*
<
| VARIABLE_VALUES = <*[val]*>

Explanation:

LAYOUT = [IVAL]	D	Layout-indicator specifying the order in which the values from input file are assigned to the function value in a grid point. Possible values for LAYOUT and their meaning are: [1] 1. function values at grid points: $[(m_1, n_1)$,$(m_1, n_1 + 1) \ldots (m_1, n_2)]$, $[(m_1 + 1, n_1) \ldots (m1 + 1, n_2)] \ldots [(m_2, n_1) \ldots (m_2, n_2)]$ columns; first column is left; column values from bottom to top 2. function values at grid points: $[(m_1, n_1)$,$(m_1 + 1, n_1) \ldots (m_2, n_1)]$, $[(m_1, n_1{+}1) \ldots (m_2, n_1{+}1)] \ldots [(m_1, n_2) \ldots (m_2, n_2)]$ rows; first row is bottom; row values from left to right 3. function values at grid points: $[(m_2, n_1)$, $(m_2, n_1 + 1) \ldots (m_2, n_2)]$, $[(m_2 - 1, n_1) \ldots (m_2{-}1, n_2)] \ldots [(m_1, n_1) \ldots (m_1, n_2)]$ columns; first column is right; column values from bottom to top 4. function values at grid points: $[(m_2, n_1)$, $(m_2 - 1, n_1) \ldots (m_1, n_1)]$, $[(m_2, n_1 + 1) \ldots (m_1, n_1{+}1)] \ldots [(m_2, n_2) \ldots (m_1, n_2)]$ rows; first row is bottom; row values from right to left

[1] Assume the limits of the box are given by (m_1, n_1) and (m_2, n_2) with $m_1 \leq m_2$ and $n_1 \leq n_2$. In the case of global input $n_1 = 1$, $n_2 = NMAX$, $m_1 = 1$ and $m_2 = MMAX$. The number of required function values is then $n_{tot} m_{tot}$, where :
$n_{tot} =$ (number of enclosed n grid points) = $n_2 - n_1 + 1$
$m_{tot} =$ (number of enclosed m grid points) = $m_2 - m_1 + 1$

5. function values at grid points: $[(m_1, n_2), (m_1, n_2-1) \ldots (m_1, n_1)], [(m_1+1, n_2) \ldots (m_1+1, n_1)] \ldots [(m_2, n_2) \ldots (m_2, n_1)]$ columns; first column is left; column values from top to bottom
6. function values at grid points: $[(m_1, n_2), (m_1+1, n_2) \ldots (m_2, n_2)], [(m_1, n_2-1) \ldots (m_2, n_2-1)] \ldots [(m_1, n_1) \ldots (m_2, n_1)]$ rows; first row is top; row values from left to right
7. function values at grid points: $[(m_2, n_2), (m_2, n_2-1) \ldots (m_2, n_1)], [(m_2-1, n_2) \ldots (m_2-1, n_1)] \ldots [(m_1, n_2) \ldots (m_1, n_1)]$ columns; first column is right; column values from top to bottom
8. function values at grid points: $[(m_2, n_2), (m_2-1, n_2) \ldots (m_1, n_2)], [(m_2, n_2-1) \ldots (m_1, n_2-1)] \ldots [(m_2, n_1) \ldots (m_1, n_1)]$ rows; first row is top; row values from right to left

Default = 1

CONST_VALUE = [VAL]	D	Constant value for the complete field. Default = 0
VARIABLE_VALUES =< [VAL] >	O	It is possible to specify a function value at each grid point. The order in which the values are to be given is defined by means of layout-indicator. In the case of 3D the information must be specified as a set of KMAX separate layers, each layer given according to the global layout-indicator (i.e. MMAX*NMAX*KMAX values must be specified, beginning with the top layer).

3.1.2.2 LOCAL

In LOCAL the function values at grid points specified in GLOBAL can locally be overwritten by specifying boxes (i.e. rectangles). In the 3D-case a box is a rectangle drawn in the horizontal plane identified by the layer-index.

```
LOCAL
<       BOX : MNMN = ([ival],[ival])([ival],[ival])          LAYER = [ival]
          | CONST_VALUES = [val]
          <
          | CORNER_VALUES = [val],[val],[val],[val]
          <
```

| VARIABLE_VALUES = <*[val]*>
>

Explanation:

BOX	R	A BOX is defined by specifying its opposite corner points (m1,n1) and (m2,n2), where $m1 \leq m2$ and $n1 \leq n2$. In this rectangle the global function value of a "field" variable can be overwritten by new values. It is possible to define more than one box for one single "field" variable. When the rectangles defined in the boxes have common grid points, the latest values specified for those grid point will be used. The data can be specified either by means of a single value defining all points within the box or by means of a array of data. In the latter case the data should be given according to the following scheme:
MNMN = ([IVAL], [IVAL]) ([IVAL], [IVAL])	M	Corner points of the rectangular box, specifying $(m_1, n_1)(m_2, n_2)$, where $m_1 \leq m_2$ and $n_1 \leq n_2$.
LAYER = [IVAL]	O	Layer index , where 0 ≤ layer ≤ kmax. If layer is not specified or layer=0, a uniform vertical distribution is assumed. However, when the function values belong to a data-array which is defined for layers 0 until kmax, layer=0 is only valid for the upper layer and layer=-1 will define the uniform vertical distribution. As default, 3D-arrays are assumed to be defined for layers 1 until kmax, unless stated otherwise in their input description. LAYER is only relevant in the 3D-case.
CONST_VALUES = [VAL]	O	The function at all grid points in the box gets this value.
CORNER_VALUES = [VAL][VAL][VAL][VAL]	O	The function values at the corner points of the box are given in the following order (m_1, n_1), (m_2, n_1), (m_2, n_2), (m_1, n_2). The function values at the other grid points enclosed by the box will be determined by means of bilinear interpolation.
VARIABLE_VALUES = < [VAL] >	O	Inside the box for each grid point a function value is specified. The order in which the values are to be given is set by LAYOUT under keyword GLOBAL.

For example:

```
GLOBAL
    CONST_VALUES = 40.5
    LAYOUT = 4
LOCAL
    BOX: MNMN = (10, 5), (50,100)
    CONST_VALUES = 38
```

or

```
GLOBAL
    CONST_VALUES = 0
    LAYOUT = 3
LOCAL
    BOX: MNMN = (10, 5), (11,7)
    VARIABLE_VAL = 2 2.3 2.4 1.9 2.0 3.2
```

3.1.3 Time series

Time series are used for boundary conditions. There are two pos-sibilities in SIMONA to specify time series: 'regular' and 'irregular'.

Regular time series are given by using a time frame (FRAME), defining a time first, time interval and time last (all times in minutes elapsed from midnight of the reference date as specified in FLOW, PROBLEM, TIMEFRAME, DATE). The values must be given at constant time intervals.

In case of irregular time series a time can be specified together with the values related to this time, repeatedly. In this case the times are given in day hour:minute. A minute can be specified with a decimal value (e.g. 5.75). In this notation midnight of the reference date as specified in FLOW, PROBLEM, TIMEFRAME, DATE is 0 0:00. It is possible to specify negative times, but note that this only holds for the days. For example -1 23:00 means 1 hour before the reference date and -1 1:00 means 23 hours before the reference date.

All time series are interpolated during computation.

'Timeseries' is not a (sub)keyword, but the name of an input structure that may be embedded in other keyword structures described in this guide. Initial values for time series must be specified (see paragraph 2.9.1.3).

Examples:

1. Timeseries

    SERIES = 'REGULAR'
    FRAME = 100. 5. 125.
    VALUES = 5 6 7 7 3 4

2. Timeseries

    SERIES = 'IRREGULAR'
    TIME_AND_VALUES = (0 1:00.5) 2

TIME_AND_VALUES = (0 1:10.5) 8
TIME_AND_VALUES = (0 2:00.5) 12

3.2 Main keywords Timeseries/Wind2str input

The input has 3 optional main keywords: (M = mandatory, O = optional):

GENERAL (O)
TIMEFRAME (O)
GRID (O)

These keywords are described in the following sections.

Chapter 4

The Timeseries/Wind2str input

The Timeseries/Wind2str input of the WAQWND program is described in this chapter. Physically the Timeseries and Wind2str input resides in separate files, due to the fact that the scope of Wind2str is not limited to Timeseries, but is also applicable to ASCII/GRIB/NetCDF/SVWP input.

Note: Timeseries input always contains wind speeds, which are always converted to wind stresses.

4.1 General Information

For general information about the conventions being used for the data fields the reader is referred to section 3.1.1 of this user's guide.

4.2 General (optional)

This main keyword has 3 optional sub-keywords:

GENERAL ([O])
    PHYSICALPARAM [O]
    WIND [O]
    SPACE_DEP_CD [O]

The General keyword is used for

- the specification of speed to stress conversion
- the definition of timeseries

The sub-keywords are described in the following sections.

4.2.1 Physical parameters (optional)

The physical parameters gravity, water density, air density and dynamic viscosity of water can be defined in this subsection.

PHYSICALPARAMETERS
GRAVITY = *[val]*
WATDENSITY = *[val]*
AIRDENSITY = *[val]*
DYNVISCOSITY = *[val]*

Explanation:

GRAVITY	D	Gravity $(m\ s^{-2})$ Default = 9.813
WATDENSITY	D	Water density $(kg\ m^{-3})$ Default = 1023.0
AIRDENSITY	D	Air density $(kg\ m^{-3})$ Default = 1.205
DYNVISCOSITY	D	Dynamic viscosity of water $(kg\ m^{-1}\ s^{-1})$ Default = 0.001

4.2.2 Wind (optional)

In this section the effect of uniform (constant is space) wind can be taken into account.

WIND
WSPEED = *[val]*
WANGLE = *[val]*
WCONVERSIONFACTOR = *[val]*
WUNIT = text
| CONST_CD wstresscoefficient=[val]
$<$
| VARIABLE_CD cda=[val] cdb=[val] wind_cda=[val] wind_cdb=[val]
$<$
| CHARNOCK beta=[val] height=[val]
SERIES =[text]
FRAME =[val1] [val2] [val3]
| VALUES = < ([val1] [val2]) > (i.c. series='regular')
$<$
| <TIME_AND_VALUES =[tval] [val1] [val2] > (i.c. series='irregular')

Explanation:

WSPEED=[VAL]	D	Global wind speed in a dimension specified by wunit. Default = 0.0
WANGLE=[VAL]	D	Global wind direction, in degrees from 0 to 360. Wind direction is measured clockwise from north, where (wind coming from) North equals to 0^o, (wind coming from) East equals 90^o and so on. Default = 0

Note: - Remark the difference in specifying the angles for wind direction (wangle) and the model (anglegrid). The first must be given in degrees from the North where the wind is coming from, while the model angle is measured from the positive Y-direction to the North, both clockwise.

Explanation:

WCONVERSIONFACTOR=[VAL]	D	Wind conversion factor, converts the dimension of the wind speed specified by wunit to $m\ s^{-1}$. Thus if wind speed is given in knots, then wconversionfactor must be set to 0.5144. Default = 1.0
WUNIT=[TEXT]	O	Name of wind speed unit to display. The maximum length of text is 4 characters.
CONST_CD	D	const_cd is a flag-keyword. If this keyword is specified, a wind speed-independent C_d-coefficient, defined with the keyword wstresscoefficient will be used in the computation of the force due to wind.
WSTRESSCOEFFICIENT=[VAL]	D	Coefficient used in the computation of the force due to wind. Should be specified together with the keyword const_cd. Default = 0.0026
VARIABLE_CD	O	variable_cd is a flag-keyword. If this keyword is specified, a wind speed-dependent C_d-coefficient, defined with the keywords: CdA, CdB, Wind_CdA and Wind_CdB will be used in the computation of the force due to wind.
CDA=[VAL] , CDB=[VAL]	O	Two coefficients used in the computation of the force due to wind. Should be specified together with the keyword Variable_Cd.
WIND_CDA=[VAL] , WIND_CDB=[VAL]	O	Two wind speed-values used to calculate the C_d-coefficient. Should be specified together with the keyword Variable_Cd.

Note: For the computation of the C_d-coefficient that depends piecewise linearly on wind speed, we

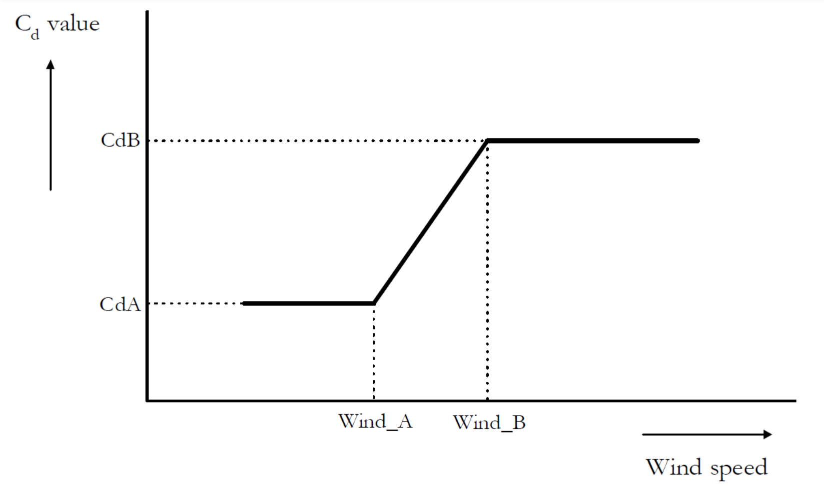

Figure 4.1: C_d coefficient related to wind speed

need both lower and upper bounds of the coefficients and speeds. The following rules are applied when calculating the wind drag coefficient:

```
if wind speed < WIND_CdA: Cd = CdA
if wind speed > WIND_CdB: Cd = CdB
```

For the wind speed-values between Wind_CdA and Wind_CdB the C_d-coefficient is obtained by means of linear interpolation between CdA and CdB, refer to Figure 4.1.

Explanation:

CHARNOCK	O	Charnock is a flag keyword. If this keyword is specified, a wind drag coefficient C_d depending on wind speed in an implicit manner based on the Charnock drag formulation will be used in the computation of the force due to wind.
BETA=[VAL]	D	The dimensionless Charnock coefficient β. Default = 0.032
HEIGHT=[VAL]	D	The height (m) above the free surface where the wind speed has been measured. Default = 10.0

Note: Based on the mixing length theory, the velocity of the wind in the turbulent layer above the free surface follows a logarithmic velocity profile in which the friction velocity u_* and the roughness height z_0 have to be determined. Charnock (1955) proposed the following relation for the roughness height: $z_0 = u_*^2/g$ with β the dimensionless Charnock coefficient and g the gravity acceleration.

Explanation:

SERIES=[TEXT]	O	Series can have two possible values: 'regular' or 'irregular'. When Series = 'regular' keywords Frame and Values are expected. When Series = 'irregular' keyword Time_and_wind_values is expected.
FRAME=[VAL1] [VAL2] [VAL3]	O	[val1] is the first time for which wind speed and angle are given. [val2] is the time interval at which wind speed and angle are given. [val3] is the last time at which wind speed and angle are given. (All these times are given in minutes)
VALUES= < ([VAL1] [VAL2]) >	O	The values for wind speed (=[val1]) (dimension: See wunit) and angle (=[val2]) (dimension: See wangle) are given for the times as defined at the keyword frame.
TIME_AND_WINDVALUES=[TVAL] [VAL1] [VAL2]	O	In this case it is possible to give wind speed and angle at non-equidistant times.

4.2.3 Space_dep_cd (optional)

In this section space-dependent wind drag coefficient (C_d) can be given. Both global and local spatial values for wind speed-independent and wind speed-dependent (in piecewise linear manner) wind drag coefficients can be taken into account. This section consists of two subsections:

```
SPACE_DEP_CD
      | WSPEED_INDEPENDENT
     <
      | WSPEED_PIECE_LINEAR
```

Wspeed_independent (optional)

Wind speed-independent wind drag coefficient C_d to be used in the computation of the force due to wind can be defined here.

WSPEED_INDEPENDENT
GLOBAL (see paragraph 3.1.2.1)
LOCAL (see paragraph 3.1.2.2)

Wspeed_piece_linear (optional)

Wind drag coefficient C_d depending on wind speed in a piecewise continuous linear manner to be used in the computation of the force due to wind can be defined here. This keyword should be followed by the subkeywords Cd_low and Cd_high, which are the lower and upper bounds of C_d, respectively.

WSPEED_PIECE_LINEAR
    CD_LOW
    CD_HIGH

Explanation:

CD_LOW	O	The lower bound used to calculate the wind drag coefficient.

CD_LOW
    GLOBAL
    LOCAL

Global (mandatory)

GLOBAL
    LAYOUT = [ival]
    WSPEED_LOW = [val]
    | CONST_VALUES = [val]
  <
    | VARIABLE_VALUES = < [val] >

Explanation:

CONST_VALUES = [VAL]	D	See paragraph 3.1.2.1 Default = 0.0
VARIABLE_VALUES = < [VAL] >	O	See paragraph 3.1.2.1
LAYOUT = [IVAL]	D	See paragraph 3.1.2.1 Default = 1
WSPEED_LOW = [VAL]	M	The lower bound of wind speed used to calculate the C_d-coefficient. Currently ignored.

Local (optional)

See paragraph 3.1.2.2 for this subsection.

Explanation:

CD_HIGH	O	The upper bound used to calculate the wind drag coefficient.

CD_HIGH
    GLOBAL
    LOCAL

Global (mandatory)

GLOBAL
    LAYOUT = [ival]
    WSPEED_HIGH = [val]
    | CONST_VALUES = [val]
  <
    | VARIABLE_VALUES = < [val] >

Explanation:

CONST_VALUES = [VAL]	D	See paragraph 3.1.2.1 Default = 0.0
VARIABLE_VALUES = < [VAL] >	O	See paragraph 3.1.2.1
LAYOUT = [IVAL]	D	See paragraph 3.1.2.1 Default = 1
WSPEED_HIGH = [VAL]	M	The upper bound of wind speed used to calculate the C_d-coefficient. Currently ignored.

Local (optional)

See paragraph 3.1.2.2 for this subsection.

Note: For the computation of the C_d-coefficient that depends piecewise linearly on wind speed, we need both lower and upper bounds of the coefficients and speeds. The following rules are applied when calculating the wind drag coefficient:

```
if wind speed < WSPEED_LOW: Cd = Cd_LOW
if wind speed > WSPEED_HIGH: Cd = Cd_HIGH
```

For the wind speed-values between WSPEED_LOW and WSPEED_HIGH the C_d-coefficient is obtained by means of linear interpolation between Cd_Low and Cd_High, refer to Figure 4.1.

4.3 Timeframe (optional)

In this subsection the start and end time of the simulation are given.

TIMEFRAME
    DATE =[text]
    TSTART =[val]
    TSTOP =[val]
    TSTEP =[val]

Explanation:

DATE=[TEXT]	M	Reference date in the form *dd mmm yyyy*, e.g. 12 oct 1987. *mmm* can be: jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec. Midnight starting this date is time zero for a simulation. Times in minutes, such as TSTART, are elapsed minutes from midnight beginning the simulation start date.
TSTART=[VAL]	D	Start time of the simulation in elapsed minutes from midnight at the beginning of the simulation start date. Default = 0.0
TSTOP=[VAL]	D	End time of the simulation in elapsed minutes from midnight at the beginning of the simulation start date.
TSTEP=[VAL]	M	Time step used in the computation (minutes).

The TIMEFRAME section is obligatory in case of timeseries

4.4 Grid (optional)

In this section the grid may be specified, by specifying the corners of the grid (spherical coordinates). The defaults are chosen such that a spherical grid is made, somewhat larger than the CSM8-model. This grid will do for all models currently in use (except for DCSM6). The grid defines the points at which the TIMESERIES are valid.

GRID
    LONG_FIRST
    LATT_FIRST

LONG_LAST
LATT_LAST
LONG_INC_DIR
LATT_INC_DIR

Explanation:

LONG_FIRST	D	Longitude of one of the corners of the grid Default = -15.0
LATT_FIRST	D	Latitude of corner "Long_first" Default = 45.0
LONG_LAST	D	Longitude of the opposite corner of "Long_first" Default = 15.0
LATT_LAST	D	Latitude of corner "Long_last" Default = 65.0
LONG_INC_DIR	D	Direction (±1) of Long_last relative to Long_first Default = 1
LATT_INC_DIR	D	Direction (±1) of Latt_last relative to Latt_first Default = 1

The GRID section is used in case of timeseries.

Chapter 5

Technical background

5.1 Forecast time

In GRIB files it is common that we have to deal with overlapping times. WAQWND selects the wind, stress and pressure data with the shortest forecast time.

Suppose we have each 6 hours new meteorological output with data written each 3 hours, WAQWND we use the timesteps as given in the table here below:

file nr	simulation date	forecast time	selected by WAQWND
1	1 jan 0:00	0 h	yes
1	1 jan 0:00	3 h	yes
1	1 jan 0:00	6 h	no
1	1 jan 0:00	9 h	no
1	1 jan 0:00	12 h	no
2	1 jan 6:00	0 h	yes
2	1 jan 6:00	3 h	yes
2	1 jan 6:00	6 h	no
2	1 jan 6:00	9 h	no
2	1 jan 6:00	12 h	no
3	1 jan 12:00	0 h	yes
3	1 jan 12:00	3 h	yes
3	1 jan 12:00	6 h	only if last file
3	1 jan 12:00	9 h	only if last file
3	1 jan 12:00	12 h	only if last file

In this case, 15 timesteps are available on the GRIB-files. 6 of them are overlapping, so WAQWND writes 9 timesteps on the SDS-file. From the last file, larger forecast times are used.

If larger forecast times are undesirable, use the command line options -fc_date and -time.

5.2 Cumulative stress

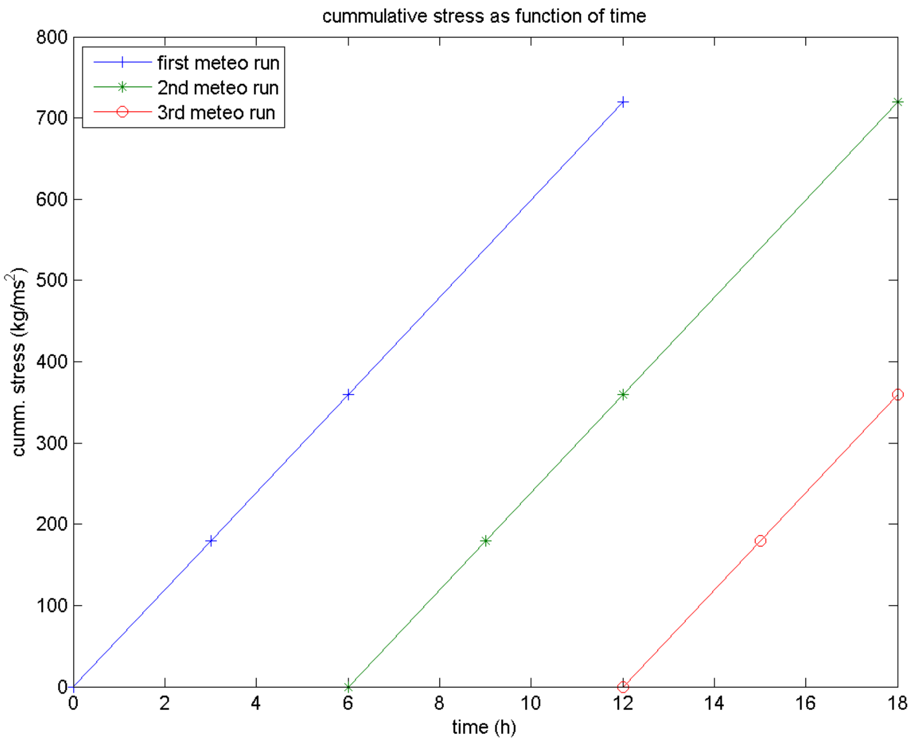

Figure 5.1: Cumulative wind stress given at certain forecast times

If the meteorological input consists of cumulative stresses, see Figure 5.1, the user have to use stress = 2 or stress = 3. WAQWND will convert the cumulative stress into instanteous stress. This conversion is taking the derivative. It is implemented by dividing the difference in cumulative stress by the difference in time at two timesteps. Two options are implemented.

Note that the value for $9h$ may be not available, and the value for $0h$ is actually the value after one timestep of the meteorological model (e.g. $300s$.)

5.2.1 Cumulative stress, option stress = 2

With the option stress = 2, WAQWND will convert stress as follows:

$$s(0h) = \frac{S(3h) - S(0h)}{\Delta t} \quad (5.1)$$

$$s(3h) = \frac{S(6h) - S(0h)}{2\Delta t} \quad (5.2)$$

The implementation is done in subroutine wawg8a.

5.2.2 Cumulative stress, option stress = 3

With the option stress = 3, WAQWND will convert stress at times in between.

$$s(1.5h) = \frac{S(3h) - S(0h)}{\Delta t} \tag{5.3}$$

$$s(4.5h) = \frac{S(6h) - S(3h)}{\Delta t} \tag{5.4}$$

This also have to be done for the pressure:

$$p(1.5h) = \frac{p(0h) + p(3h)}{2} \tag{5.5}$$

$$p(4.5h) = \frac{p(3h) + p(6h)}{2} \tag{5.6}$$

The implementation is done in subroutine wawg8b.

5.3 Format description ASCII wind files

An example of an ASCII wind file looks like this:

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	19	11	4										
2	99	4	255	128	1	103	0	2004	01	01	0	0	1
3	0	0	1	0	0	21							
4	0	2	2	47500	-12500	128	62500	13500	26000	15000	96		
5	103000.	103000.											
6	103000.	103000.											

The first 4 lines is the header of the file, the rest is the actual data. The header has three parts: a line with number of elements, section 1 which is splitted over 2 lines and section 2. Not all values of the header are used by WAQWND . The meaning of the values of header which are used, is as follows:

row	column	example value	description
1	1	19	in1 = number of elements in section 1 of KNMI-ASCII file
1	2	11	in2 = number of elements in section 2 of KNMI-ASCII file
1	3	4	inxny = number of data points in section of KNMI-ASCII file
2	5	1	ifldtp = Type of the field values according the WMO table 2. used are: 001 Pressure Pa 002 Pressure reduced to MSL Pa 033 u-component of wind m/s 034 v-component of wind m/s 081 land cover (1=land, 0=sea) Proportion 124 Momentum flux (stress) u component N/m^2 125 Momentum flux (stress) v component N/m^2 151* Pressure (of some kind, some unit) 165* wind speed u component m/s 166* wind speed v component m/s
2	8	2004	iyear =year (YYYY) of reference time
2	9	01	imonth = month (MM) of reference data
2	10	01	iday = day (DD) of reference time
2	11	0	ihour = hour (HH) of reference time
2	12	0	imin = minutes (MM) of reference date
2	13	1	ifunit = Unit of forecast leadtime (WMO table 4), used are: 0 : Minutes 1 : Hours 2 : Day
3	1	0	ifclt = forecast lead time in time unit ifunit
4	1	0	igdtp = Data representation type (WMO table 6) used are: 0: Latitude/longitude grid - equidistant cylindrical or Plate Carree projection 1: Mercator projection 10: rotated/shifted Latitude/longitude grid
4	2	2	mmw =number of nodes in grid in m-direction
4	3	2	nmw = number of nodes in grid in n-direction
4	4	47500	iang1y = latitude of first grid point in millidegrees
4	5	-12500	iang1x = longitude of first grid point in millidegrees
4	7	62500	iangy = longitude of last grid point in millidegrees
4	8	13500	iangx = latitude of last grid point in millidegrees
4	11	96	iscmod = scanning mode (see 5.4)

5.4 Scanning mode iscmod

Both ASCII and Grib files have a so called scanning mode. It is defined as follows.

The scanning mode iscmod consists of three bits with meaning:

iscmb1 = 0 → the first point has m = 1
iscmb1 = 1→ the first point has m = mmax
iscmb2 = 0→ the first point has n = nmax
iscmb2 = 1→ the first point has n = 1
iscmb3 = 0 → adjacent points in i-direction are consecutive
iscmb3 = 1 → adjacent points in j-direction are consecutive

Only bits number 5, 6 and 7 of iscmod are used:

iscmb1 = bit (iscmod , 7)
iscmb2 = bit (iscmod , 6)
iscmb3 = bit (iscmod , 5)

Chapter 6

Examples

6.1 Timeseries input

```
GENERAL

  WIND
#   wind gegevens:
    WSTRESScoefficient=  0.00280, WCONVersionfactor=  1.00, WUNIT='m/s '
#   opgave global windsnelheid en richting:

  WSPEED=  0.00, WANGLE=  0.00, SERIES='irregular'
  TIME_and_WINDVALUES=        0  0: 0      8.00    340.00
  TIME_and_WINDVALUES=        0  1: 0      6.50    350.00
  TIME_and_WINDVALUES=        0  2: 0      7.00    340.00
  TIME_and_WINDVALUES=        0  3: 0      6.50    330.00
  TIME_and_WINDVALUES=        0  4: 0      8.00    340.00
  TIME_and_WINDVALUES=        0  5: 0      5.00    350.00
  TIME_and_WINDVALUES=        0  6: 0      5.00    340.00
  TIME_and_WINDVALUES=        0  7: 0      6.00    350.00
  TIME_and_WINDVALUES=        0  8: 0      6.00    360.00


TIMEFRAME
  DATE='02 OCT 1975', TSTART= 0, TSTOP= 10000, TSTEP= 10

#GRID
# LONG_FIRST= -16.0
#GRID
#   LONG_FIRST= -16.0
```

6.2 Wind2str input 1

```
# input file for Cd coefficient that is based on the Charnock formulation

GENERAL
   WIND
      CHARNOCK
```

6.3 Wind2str input 2

```
# input file for Cd coefficient that is picewise linear on wind speed

GENERAL
   WIND
      VARIABLE_CD: WIND_CdA = 10.20, CdA = .00114,
                   WIND_CdB = 15.90, CdB = .00258,
```

6.4 Wind2str input 3

```
# input file for Cd coefficient that is picewise linear on wind speed,
# and space varying.

GENERAL
   WIND
      VARIABLE_CD: WIND_CdA = 10.20, CdA = .0,
                   WIND_CdB = 15.90, CdB = .0,

   SPACE_DEP_CD
      WSPEED_PIECE_LINEAR
         CD_LOW
            GLOBAL
               LAYOUT = 1 # WSPEED_LOW = 10.20
            LOCAL
               BOXMNMN = (1,1) (201,173)
               CONST_VALUES .00114
               BOXMNMN = (1,1) (75,75)
               CONST_VALUES .00014
         CD_HIGH
            GLOBAL
               LAYOUT = 1 # WSPEED_HIGH = 15.90
            LOCAL
               BOXMNMN = (1,1) (201,173)
```

```
CONST_VALUES .00258
BOXMNMN = (1,1) (75,75)
CONST_VALUES .00358
```

References

1. Drs. C. van Velzen (21-08-2002)

Technisch Rapport TR02-11

Uitbreiding wind verwerking in WAQWND , WAQPRE en WAQPRO.

2. Quick Reference Guide WAQWND

3. User's Guide WAQUA General Information

4. User's Guide WAQPRE