Opdrachtgever:

RWS-DWW in samenwerking met WL | Delft Hydraulics

Uitbreiding Delft-FLS / koppeling met Sobek

Deelrapport:

Delft-1D2D: Systeemdocumentatie

Systeemdocumentatie

November 1999

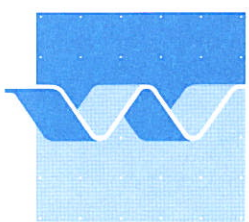
R3240/R3260

WL | delft hydraulics

Uitbreiding Delft-FLS / koppeling met Sobek

Deelrapport:

Delft-1D2D: Systeemdocumentatie



WL | delft hydraulics

WL | delft hydraulics

OPDRACHTGEVER:	Rijkswaterstaat, DWW Postbus 5044 2600 GA Delft
TITEL:	Uitbreiding Delft-FLS / koppeling met Sobek Deelrapport: Delft-1D2D: Systeemdocumentatie

SAMENVATTING:

In januari 1998 sloten RWS | Dienst Weg en Waterbouwkunde (DWW) en WL | Delft Hydraulics Samenwerkingsovereenkomst nummer DWW 1382. In het kader van deze overeenkomst hebben DWW en WL besloten om gezamenlijk een onderzoeks- en ontwikkelingsproject uit te voeren. Dit project is gericht op de uitbreiding van het pakket Delft-FLS en zijn koppeling met Sobek voor het berekenen van overstromingen veroorzaakt door dijkdoorbraak. Deze overeenkomst werd in 1999 vervangen door de overeenkomst DWW-1613.

Op basis van de resultaten van de eerste onderzoeksfase is besloten om Delft-FLS functionaliteiten in een nieuwe Oveland Flow module binnen het pakket Sobek-Lowland van WL| Delft Hydraulics te bouwen. De User Interface van Sobek Lowland is aangepast om de nieuwe 2D elementen te kunnen verwerken en visualiseren. De combinatie van de bestaande Sobek Lowland Channel Flow module en het nieuwe Overland Flow module wordt genoemd Delft-1D2D.

Dit deelrapport geeft technische details over software-structuur, nieuwe ontwikkelde software en aanpassingen aan bestaande software uitgevoerd ter behoefte van het nieuwe Overland Flow module en de communicatie met Delft-FLS en Sobek-LL/Pluv routines, uitvoering van berekeningen, routines voor voorbewerking en nabewerking van gegevens.

WL | Systeemontwikkelaars: J. Dhondia, J. Crebas, H. Kernkamp, G. Stelling, K.J. van Heeringen, P.J. van Overloop..

REFERENTIES:	Samenwerkingsovereenkomst DWW-1382 tussen DWW en WL| Delft Hydraulics, januari 1998 Opdracht nummer DWW-1613, 1999. Contactpersoon: Ir. M.Jak / Ir. R. Jorissen

VER.	AUTEUR		DATUM	OPMERK.	REVIEW		GOEDKEURING	
1	J. Crebas /J. Dhondia		11/11/99	al verwerkt	M. Laguzzi / 12/11/99		I. van Rooij	

PROJECTNUMMER:		R3240/ R3260						
TREFWOORDEN:		Sobek Lowland / Delft-FLS / Delft-1D2D / systeemdocumentatie / Delft-rekenschema						
INHOUD:	TEKST		TABELLEN	-	FIGUREN	-	APPENDICES	-
STATUS:	☐ VOORLOPIG		☒ CONCEPT		☐ DEFINITIEF			

Inhoud

1 Inleiding

1.1 Algemeen

Delft-1D2D (Sobek-CF-OF) is de naam gegeven aan de koppeling tussen Delft-FLS en Sobek-Lowland ontwikkeld door WL | Delft Hydraulics in samenwerking met RWS DWW in de periode 1998-1999 (Contributie DWW via externe opdracht R3240: Samenwerkings-overeenkomst DWW / vanaf 1999, overeenkomst DWW ; Contributie WL R3260: eigen speurwerkfondsen). De koppeling is tot stand gebracht door middel van het bouwen van een nieuwe Overland Flow module in Sobek-Lowland. De User Interface, het programma Netter, de Parser, het simulatieprogramma Sobeksim en de nabewerkingsprogrammatuur van Sobek LL/Urban zijn uitgebreid met een aantal opties benodigd voor het uitvoeren van 1D2D-simulaties en het presenteren van resultaten.

De Sobek-1D2D programmatuur is ontwikkeld en getest volgens de ISO-9001 standaards voor het ontwikkelen en testen van software.

De ontwikkelde software is millennium bestendig (zie document Testing Sobek for Year 2000 compliance).

1.2 Sobeksim

In Sobeksim zijn de voor de 2D-simulaties benodigde routines van Delft-FLS opgenomen en gekoppeld met bestaande Sobeksim-routines (zie Hoofdstuk 2). De administratie van de data-arrays is aangepast om een efficiënte werking van 1D en 2D toepassingen te garanderen.

1.3 Parsers

De bestaande Sobek-Parser (interface-programma tussen de model database en de Sobeksim invoerbestanden) is uitgebreid met een routine om x/y-coordinaten te verwerken.

Er is een speciale parser (Parser2D) geschreven voor het lezen van de model database en het schrijven van het invoerbestand voor de 2D-informatie voor Sobeksim. De extra benodigde invoer is weergegeven in Hoofdstuk 3, Aanpassing/toevoeging Model Database Parser2D. In Hoofdstuk 4, Parser2D: modulebeschrijvingen, staat een beschrijving van de nieuw ontwikkelde routines van Parser2D.

1.4 Testen

Ten behoeve van het testen is een testplan ontwikkeld. Dit testplan is uitgevoerd en de resultaten zijn weergegeven in het testrapport "Delft-1D2D: tests implementatie en performance van de koppeling tussen Delft-FLS en Sobek-Lowland".

Tijdens de tests is een aantal problemen / fouten in de software ontdekt; meeste fouten zijn al verholpen. In de bijlage wordt een voorbeeld van de laatste"fout melding en verbeterings-aktie lijst" gepresenteerd.

2 Aanpassingen in SOBEKSIM

The next changes have been made to Sobeksim, in order to incorporate the coupling between 1D and 2D features within this calculation program.

ADVEC2D.FOR - module to calculate the advection within 2D domain.

FLS.FOR - module which defines the arrays and variable which are used to read the data for 2D part from the mdf file and to transfer this data to the 1D arrays and variable for the solver.

FLSCOMMON.FOR - module which define the arrays and variable which are used within the 2D subroutines and functions of SOBEKSIM.

FLSDATA.FOR - subroutine to read the mdf file, allocates the required arrays (as defined in FLS.FOR) and defines the numbers of 2D points (nodes) and 2D lines (branches).

FLSTOLINE.FOR - subroutine to transfer the read 2D grid information into the LINEV array (linev array keep the connectivity information for network - i.e. the linev in whole defines the network with its nodes as points and branch as line - thus the name).

Linev (1,*) - the line (branch number)
Linev (2,*) - internal used by CG solver
Linev (3,*) - left point (node) type
Linev (4,*) - right point (node) type
Linev (5,*) - left point (node) number
Linev (6,*) - right point (node) number

FLSINFL.FOR - subroutine to transfer the read 2D data into the point (which has level and depth) and line (which has the velocity) as defined by LINEV array.

FLSUTILS.FOR - collection of subroutines and functions of DELFT FLS. Further to these collections of subroutines and functions, there are few routines developed for the coupled version. They are:

SUBROUTINE flschzt - defines the friction for 2D line
SUBROUTINE CONNECT2DTOLINE - creates the linev array for the connections required for connecting the two 2D domains (grids).
SUBROUTINE CONNECT2D - sets the data for the connections as defined in subroutine CONNECT2DTOLINE into the point and line of LINEV array.
SUBROUTINE CONNECT1D2Ddata - identifies the number of points (nodes) and lines (branches) required to define a 2D domain as 1D network.
SUBROUTINE CONNECT1D2Dtoline- creates the linev array for the connections required for connecting the 1D to 2D domain points.
SUBROUTINE CONNECT1D2DINFL - sets the data for the connections as defined in subroutine CONNECT1D2Dtoline into the point and line of LINEV array.

REAFLS.FOR - subroutine to read the Delft FLS MDF file.

PLINCGC.FOR - subroutine which writes the output for 2D part into three forms -

- ARC INFO map files
- Incremental file
- Binary His file for HISTORY locations

Rest of the routines for SOBEKSIM are not modified except that following variables are set so that the division between the 1D point and 2D point or 1D line and 2D line is identified and taken into consideration in different routines of SOBEKSIM

nbranp = nbran
nbrtot = nbran + nbranfls
nodtot = max(nnodpl, ngridp) + nnodfls
ngridp = ngridp
ngridtot = ngridp + (nbranfls + 1)

where nbranp - number of 1D reaches
nbrtot - total number of 1D reaches and 2D reaches
nbranfls - number of 2D reaches (= branches) (= type = 100 internally)
nnodpl - number of 1D actual nodes
nnodfls - number of 2D nodes (= type = 100 internally)
nodtot - total number of 1D actual nodes and 2D nodes
ngridp - total number of 1D points (here the points refers to as number of actual nodes + the calculation points)
ngridtot - total number of 1D points and 2D points (here 2D points refers to the 2D cells and thus does not have any intermediate calculation points as in 1D)
Note: A reach is a line which has an actual begin node and end node and may or may not contain one or more calculation point or cross section.
A branch is line which has only a begin node (actual node or calculation point) and has an end node (actual node or calculation point).

the various routines of 1D thus loops over nbranp and nnodpl instead of nbrtot and nodtot respectively.

The flow of the various sequences (calling subroutines or functions of) SOBEKSIM remains the same except that in case the 2D grid or domain is present with the 1D network, the following flow of sequences is taken into consideration.

If 2D domain present -

CALL FLSDATA - which in turn calls the REAFLS routine
CALL FLSTOLINE
CALL FLSINFL

3 Parser2D: Aanpassing/toevoeging Model Database

3.1 Structure layer

Overzicht van files

[Structure layer]
NrOfFiles=4
net=..\work\network.st
def=..\work\struct.def
cnt=..\work\control.def
dat=..\work\struct.dat

Beschrijving van files

net=network.st

D2ST id '7' nm ' ' ci '-1' lc 0 px 106075.3 py 458695.3 d2st

Waarbij:

id	=	id van de structure
nm	=	naam van de structure
ci	=	id van de tak (reach)
lc	=	positie tov. begin van de tak (altijd 0)
px	=	X Coordinate within Grid
py	=	YCoordinate within Grid

dat=struct.dat

D2ST id 'S003' dd '6' ca 0 cj ' ' stru

Waarbij:

id	=	id van het kunstwerk
dd	=	id van de structure definitie (verwijst naar struct.def)
ca	=	geeft aan of er een controller is gedefinieerd (controller active) 1 = active: er is een controller gedefinieerd 0 = inactive: er is geen controller gedefinieerd
cj	=	controller id's

def=struct.def

Dit bestand bevat algemene definities van kunstwerken. De volgende typen kunstwerken worden onderscheiden:

112 = Breaking Dam

Voor een **Sobek weir** is de definitie als volgt:

```
D2SD      id 'dam' nm 'breakstruct' ty 112  t0   0.0  t1  4.0 dh lt 1
TBLE .... tble
d2sd
```

Waarbij:

id	=	id van de kunstwerk definitie
nm	=	naam van de kunstwerk definitie
ty	=	type kunstwerk (112 = 2D Breaking Dam)
t0	=	start time in hours from start of simulation
t1	=	end time (not if if table) in hours
dh lt 1	=	decrease in height with time table with time in hours and height decrease in m
dh lt 0 10.0	=	constant value - linear decrease in height m start at start time
wg 0 0	=	width growth at rate of 5% of gridsize
1 .6	=	width growth at rate specified in m/sec
wd 0 0	=	no increase in width growth with time (assumed total gridsize at begining of break)
wd 1 .5	=	increase in width with time in m till gridsize

3.2 Conditions layer

In deze laag worden randvoorwaarden en laterale lozingen/onttrekkingen beschreven.

Overzicht van files

```
[Condition layer]
NrOfFiles=3
net=..\work\network.cn
dat=..\work\boundary.dat
lat=..\work\lateral.dat
```

Beschrijving van files

net=network.cn

For Single Boundary Condition

```
D2PT id '1' nm '' ci '1' px 101991.8 py 458610.7 d2pt
```

For Multiple Boundaries

D2LI id 'l_1' nm '' ci 'l_1' bx 105683.4 by 454213.5 ex 107991.5 ey 454213.5 d2li

Waarbij:

id	=	id van de structure
nm	=	naam van de structure
ci	=	id van de tak (reach)
px	=	X Coordinate within Grid
py	=	YCoordinate within Grid
bx	=	Begin X Coordinate within Grid for Multiple boundaries
by	=	Begin YCoordinate within Grid for Multiple boundaries
ex	=	End X Coordinate within Grid for Multiple boundaries
ey	=	End YCoordinate within Grid for Multiple boundaries

For Initial Boundary Condition

D2IN id '1' ci '1' px 101991.8 py 458610.7 d2in

dat=boundary.dat

D2PT id 'pntbound' nm 'Multiple Boundary' ty 0 h_wd 0 10.0 d2pt

and

D2LI id 'mulbound' nm 'Multiple Boundary' ty 1 q_ dw 0 0.5 d2li

id	=	identificatie
ty	=	type randvoorwaarde 0 = water level , 1 = flow/Flux , 2 = velocity
h_ wd 0 10.0	=	constante waterstand (uiteraard alleen voor ty 0)
h_ wd 1	=	waterstand als functie van Q (tabel: 1e kolom Q, 2e kolom h) (uiteraard alleen voor ty 0) **** not implemented
h_ wt 1	=	variabele waterstand als functie van de tijd (TBLE ... tble) (tabel: 1e kolom time in hours from start of simulation, 2e kolom water level in m) (uiteraard alleen voor ty 0) .
q_ dw 0 0.5	=	constant debiet cumecs (uiteraard alleen voor ty 1)
q_ dw 1	=	Q=Q(H) volgens Q-H tabel (1e kolom=h, tweede kolom=Q) (uiteraard alleen voor ty 1) **** not implemented
q_ dt 1	=	variabel debiet als functie van de tijd (TBLE ... tble). (tabel: 1e kolom time in hours from start of simulation, 2e kolom flow in cumecs) (uiteraard alleen voor ty 1)
u_st 1	=	variabel velocity als functie van de tijd (TBLE ... tble). (tabel: 1e kolom time in hours from start of simulation, 2e kolom velocity in m/s) (uiteraard alleen voor ty 2)

3.3 Initial Conditions layer

Overzicht van files

[Initial Conditions layer]
NrOfFiles=1
dat=..\work\initial.dat

Beschrijving van files

dat=initial.dat

D2IN id '11' ci '-1' lc 0 ty 1 lv ll 0 9.88 d2in

Waarbij:

id	=	id van de initial condition point
ci	=	id van de tak (reach)
lc	=	0
ty	=	type waterstand 1=waterstand (altijd 1)
lv ll 0 9.88	=	constante waarde voor waterstand

3.4 Measured data layer

Overzicht van files

[Measurements]
NrOfFiles=0
net=..\work\network.me

Beschrijving van files

net=network.st

History stations

MEPT id '11' nm '' px 102791.8 py 459887.6 mept

Waarbij:

id	=	id van de history station
nm	=	naam van de history station
px	=	X Coordinate within Grid
py	=	YCoordinate within Grid

3.5 Friction layer

Overzicht van files

[Friction layer]
NrOfFiles=4
dat=friction.dat
wnd=windfric.dat
exr=exresist.dat
glf=globfric.dat

bed=bedfrict.dat

Dit bestand bevat de bed friction gegevens per tak.

D2FR id '1' nm 'Frictie1' ci '1' mf 0 mt cp 0 45 0 mw cp 0 45 0 d2fr

Waarbij:

id	=	id van bed frictie definitie
nm	=	naam van de bed frictie definitie (kan nu niet worden opgegeven)
ci	=	carrier id = id van de tak
mf	=	main friction type (main=main channel) 0 = Chezy 1 = Manning 4 = White Colebrooks
mt	=	Bodem frictie mt cp 0 60 0 = alle overige gevallen (Chezy/Manning etc. als constante of van de lokatie) 0 = constant mw cp 2 'd:\sob_lite\fls_files\bedfrict.asc' ' ' 2 = file 2 = file
mw	=	Wall friction mw cp 0 60 0 = alle overige gevallen (Chezy/Manning etc. als constante of van de lokatie) 0 = constant mw cp 2 'd:\sob_lite\fls_files\walfrict.asc' ' ' 2 = file

3.6 Grid layer

Overzicht van files

[Grid layer]
NrOfFiles=1
net=..\work\network.d12

Beschrijving van files

net=network.d12

by	=	begin YCoordinate
ex	=	end X Coordinate
ey	=	end YCoordinate
bc	=	begin Column position within Grid (m) in X Direction
br	=	begin Row position within Grid (n) in Y Direction
ec	=	end Column position within Grid (m) in X Direction
er	=	end Row position within Grid (n) in Y Direction

4 Parser2D: Modulebeschrijvingen

Bestand	Subroutine/function

CONVDATE.FOR> convert_date
This function takes a character string and
calculates the date and time as integers.
Date is written: YYYY/MM/DD;HH:MM:SS:hh

CONVTIME.FOR> convert_time
This function takes a character string and
calculates the time as a real value.
Time is written: HH:MM:SS:hh

countd12.for > count_d12_objects
Estimates array dimensions for FLS data arrays

countkey12.fo>count_keys_d12
Counts possible present keys in a MDB-file

GENMES.FOR > generate_message
Generates a program message

GETFIL.FOR > getfil
When the parser is started as runtime argument the
names of the CMT file list must be passed.

GKWINI.FOR > gkwini
Reads a keyword from a "windows" type ini file
> Gettko
Check for group separator
> Charfo
Create character format string

Hashbran.for> Hashfun (int)
Modified version of the hashing system used
in SOBEK_LITE/PLUVIUS. It has been adapted
to generate a hash table for the branch_id
array in the new parser.
> Hashfill
Fill hashing arrays
> Hashsearch (int)
Search in hashing arrays
> Lentrim(STRING) (int)
Determine length of string, less trailing blanks
> Upperc(STRING)
convert string to uppercase characters

INIARR.FOR > iniai1
Initialise integer array of one dimension
> iniai2
Initialise integer array of two dimensions
> iniar1

Initialise real array of one dimension
> iniar2
Initialise real array of two dimensions
> iniar3
Initialise real array of three dimensions
> iniac1
Initialise character array of one dimension

mafina.for >mafina
Reads file containing names of files.
If this file list is not present construct
file names .

pbfric.for >parse_friction
Read and write friction data

pbound.for >parse_bound
Read and write boundary data

pdomain.for >parse_domain
main routine for the parsing of the FLS(2D)data

Pflout.for >parse_flow_output
This module reads the output request list for the
flow module. The integer function 'flopts' is used
to return the integer codes.

pflrun.for >parse_flowrun_data
This routine parses the runtime data for the flow
module

pinclass.for >parse_inclass
Read and write classes of incremental files data

pinitdat.for >parse_initial_data
Read and write initial data

pmapfile.for >parse_mapfile
Read and write quantities in map-files

pmeas.for >parse_meas
Read and write measurement data

pstruc.for >parse_struc
Read and write structure data

ptext.for >ptext
read mdftext.lst and
write fixed text lines (key words) in MDF-file

readgeo.for >read_geo
Read and store topography arrays

readomglb.for>read_domain_glb
Read and write global domain data

reafil.for >read_sobek_filelist

Reads file containing names of input files per layer

SOCNAM.FOR >SOCNAM
Create a new file name.

TIMEDIFF.FOR >timediff
This function calculates the difference between the start date/time and the end date/time for sobek, including leap years. The result is returned as a double precision real value in seconds. The date/time inputs are read as character strings

versistr.for >versi(iprint,sbkrel)
print version number

ZOEK.FOR >ZOEK
Routine modified to accelerate search procedure. In order to avoid problems with the filenames, the existing routine has been kept and renamed ZOEKEN. ZOEK has now been simplified to compare strings directly.
>ZOEKEN
De routine maakt gebruik van ICHAR()
a t/m z hebben codes 97 t/m 122
a t/m z hebben codes 65 t/m 90

BIJLAGE: SOBEK Problem reports / Action requests

Entry-number	Name	Subsystem	Create-da	Priority	Status	Assigned to	Short description
000000000003495	Laguzzi M M	CMT	22 Septem	High	fixed	Melger	FLS functionalities, ARS 2059.
000000000002059	Melger E.R.	Flow (Sewer/Channel)	29 Januar	High	assigned to	Stelling	Functionality 1-2 Dimensional flow, SOBEK-FLS.
000000000003673	Melger E.R.	Flow (Sewer/Channel)	18 Octobe	High	new	melger	Integration DIWA - HYDRA and FLS, ARS 1842, 2059.
000000000003789	Dhondia J F	Flow (Sewer/Channel)	2 Novemb	High	new	melger	Water Balance not correct when only Flow 1D, FLS, ARS 2059.
000000000003500	Dhondia J F	MODELEDT	23 Septem	High	fixed	melger	Add functionality for FLS (1d2d) part in MODELEDT.EXE.
000000000003527	Dhondia J F	MODELEDT	29 Septem	High	fixed	melger	Option FLS grid in Levels wrt NAP or in Depth wrt NAP.
000000000003786	Dhondia J F	PARSER2D	1 Novemb	High	fixed	melger	PARSER2D crashes for testcase CHEZY2D, FLS, ARS 2059.
000000000002968	Laguzzi M M	PREPNETT	7 July 1	High	fixed	melger	FLS settings in PREPNETT, ARS 2059.
000000000003147	Heeringen K J v.	PREPNETT	3 August	Medium	fixed	melger	FLS functionalities are default switched on.
000000000003571	Laguzzi M M	PREPNETT	5 Octobe	High	fixed	Melger	FLS-Boundary definition is incorrect.
000000000003497	Dhondia J F	SBKCHECK	23 Septem	High	fixed	melger	Functionality for FLS (1d2d) part in SBKCHECK.
000000000002952	Heeringen K J v	SETTINGS	6 July 1	High	fixed	Melger	SETTINGS for SOBEK-FLS, R3240, ARS 2059.
000000000003471	Laguzzi M M	SETTINGS	20 Septem	High	fixed	Melger	Output timestep for SOBEK-FLS, ARS 2059.
000000000003494	Laguzzi M M	SETTINGS	22 Septem	High	fixed	Melger	FLS lay-out changes required.
000000000003803	Dhondia	SETTINGS	3 Novemb	High	assigned to	Dhondia	Incremental file Ouput Option, FLS, ARS 2059.
000000000003472	Laguzzi M M	SIMULATE	20 Septem	High	fixed	Melger	Output timestep for SOBEK-FLS, ARS 2059.

WL | delft hydraulics

Rotterdamseweg 185
postbus 177
2600 MH Delft
telefoon 015 285 85 85
telefax 015 285 85 82
e-mail info@wldelft.nl
internet www.wldelft.nl

Rotterdamseweg 185
p.o. box 177
2600 MH Delft
The Netherlands
telephone +31 15 285 85 85
telefax +31 15 285 85 82
e-mail info@wldelft.nl
internet www.wldelft.nl