

MEMO EV/M04.101
Datum 1 november 2006
Auteur(s) E.A.H. Vollebregt, C. van Velzen
Onderwerp Automatisering van de calibratie van WAQUA riviermodellen

Documentinformatie

Versie	Auteur	Datum	Opmerkingen	Review
0.9	EV	30-11-2004	Concept versie van rapportage.	CvV
1.0	EV	06-12-2004	Aanpassingen n.a.v. bespreking opdrachtgever.	
1.1	EV	07-12-2004	Kleine verbeteringen	
1.2	JG	31-10-2006	c61169: kleine aanpassingen n.a.v. opname in SIMONA	
Bestandslocatie: /v3/D02e_calibriv/report				

1 Inleiding

De afdeling WSR van Rijkswaterstaat/RIZA ontwikkelt hydraulische modellen van de grote rivieren voor diverse gebruikers binnen Rijkswaterstaat. De modellen worden voor verschillende operationele toepassingen gebruikt, zoals onder andere de voorspelling van waterstanden in de rivieren gedurende een hoogwatergolf. Hiervoor moeten de modellen met enige regelmaat opnieuw worden gecalibreerd (afgeregeld) opdat de in werkelijkheid optredende waterstanden zo goed mogelijk kunnen worden voorspeld.

In de afgelopen jaren heeft RIZA een pragmatische procedure ontwikkeld voor het inregelen van de WAQUA modellen. Hierbij worden in een aantal stappen modelwaarden vergeleken met metingen en worden op basis daarvan aanpassingen aan de ruwheidsparameter gemaakt. In eerste instantie was hierbij veel handwerk nodig: inlezen van modelwaarden en metingen in Excel, berekenen van de gewenste aanpassingen, invoeren hiervan in WAQUA invoerfiles, en opstarten van nieuwe berekeningen.

In oktober 2004 heeft RIZA aan VORtech opdracht gegeven voor het schrijven van hulp-programma's waarmee dit handwerk voor een belangrijk deel wordt geautomatiseerd. Deze opdracht is in november 2004 uitgevoerd. Dit memo bevat de rapportage van dit werk. Eerst worden de functionele aspecten van de nieuwe programmatuur beschreven (paragraaf 2, relevant voor gebruikers), dan de technische aspecten van de programma's (paragraaf 3,

relevant voor het programmeren van uitbreidingen), en tenslotte wordt in paragraaf 4 over de uitgevoerde testen gerapporteerd.

2 Uitwerking van de hulpprogramma's `calibriv.pl` en `calibriv.exe`

De gewenste automatisering van de calibratieprocedure wordt bereikt via een run-procedure “`calibriv.pl`” en een Fortran90-programma “`calibriv.exe`”. In deze paragraaf geven we een overzicht van de keuzes die zijn gemaakt bij het schrijven van deze programma's die relevant zijn voor de gebruikers van de programmatuur.

2.1 Run-procedure `calibriv.pl`

De run-procedure `calibriv.pl` is gemodelleerd naar de run-procedures `waqpre.pl` en `waqpro.pl`; runs kunnen zowel in de voorgrond als in batch-mode worden opgestart, en allerlei parameters kunnen op vergelijkbare manier worden ingesteld.

De parameters die kunnen/moeten worden opgegeven aan `calibriv.pl` zijn als volgt:

<code>-runid</code>	Identificatie van de calibratie-run (<code>calibriv-m.<runid></code>) en van de simulaties die worden uitgevoerd (<code>waqpr[eo]-m.<runid></code> , <code>SDS-<runid></code>)
<code>-params</code>	Naam van het invoerbestand met de definitie van het te calibreren probleem: tabel met te calibreren stations en parameters en met resultaten van de iteraties van de calibratie-loop.
<code>-observ</code>	Naam van het bestand met gemeten waarden voor de te calibreren stations
<code>-input</code>	Naam van de simulatie-invoerfile die wordt gebruikt (<code>siminp.???</code>)
<code>-rough</code>	Naam van het invoerbestand dat in de <code>siminpfile</code> wordt gebruikt voor de definitie van ruwheidswaarden (<code>rough.karak</code>).
<code>-maxit</code>	Het maximaal aantal iteraties (runs van <code>WAQPRE</code> en <code>WAQPRO</code>) dat mag worden uitgevoerd.
<code>-npart</code>	Het aantal subdomeinen dat in de simulatie moet worden gebruikt.
<code>-partit</code>	De roosteropdeling die in de simulatie moet worden gebruikt.
<code>-buf_pre</code>	De buffersize voor pre-processor <code>WAQPRE</code> , default 10.
<code>-buf_prt</code>	De buffersize voor partitioner <code>COPPRE</code> en collector <code>COPPOS</code> (default: 10).
<code>-buf_exc</code>	De buffersize voor executive (masterproces) <code>COEXEC</code> (default: 10).
<code>-buf_pro</code>	De buffersize voor processor <code>WAQPRO</code> (default: 10).
<code>-bufsize</code>	De buffersize voor <code>CALIBRIV</code> (default: 10).
<code>-back</code>	Vlag waarmee de hele calibratie-loop in de achtergrond kan worden opgestart (<code>-back y</code>).

Tabel 1: *Definitie van opties die kunnen worden opgegeven aan run-procedure `calibriv.pl`.*

Wanneer de run-procedure in een interactieve sessie wordt opgestart dan worden de eerste zes argumenten (t/m `-maxit`) gecontroleerd. De opgegeven bestanden moeten bestaan, en `maxit` moet een positieve waarde zijn. Als er nog geen geldige waarde voor de argumenten

is opgegeven dan wordt er door de run-procedure om gevraagd. In batch-runs wordt er in dit geval een foutmelding naar het messages-bestand geschreven en wordt de berekening afgebroken.

De overige parameters `npart` e.d. worden niet door `calibriv.pl` gecontroleerd. De waardes die in `calibriv.pl` bekend zijn worden direct doorgegeven aan `waqpre.pl` en `waqpro.pl`. Die twee procedures worden in background-mode opgestart. Dit betekent dat er nooit om ontbrekende parameters zal worden gevraagd. In plaats daarvan zullen er default-waardes worden gebruikt (`npart=1`, `partit=orb`, etc.). En als bijvoorbeeld de opgegeven partitiefiler niet bestaat dan wordt er een foutmelding gegeven en gestopt.

Domein decompositie wordt door de procedure nog niet ondersteund. Het probleem hiermee is dat er met een enkele simulatie meerdere SDS-files worden gecreëerd. Vooralsnog gaat het programma `calibriv.exe` er echter van uit dat alle te calibreren stations in dezelfde SDS-file staan. Er is extra programmeerwerk nodig om `calibriv.exe` slimmer te maken zodat hij kan uitzoeken welke stations in welke SDS-file staan. Verder zou ook de run-procedure `calibriv.pl` iets moeten worden uitgebreid. In domein decompositie runs moet de preprocessor WAQPRE namelijk voor ieder domein worden opgestart. Domein decompositie met alleen verticale verfijning zou wel heel snel kunnen worden toegevoegd. Dit is nog niet gedaan omdat de calibratiemethode primair op riviermodellen is gericht, waarin praktisch nooit verticale verfijning wordt gebruikt.

2.2 Calibratieprogramma `calibriv.exe`

De belangrijkste keuzes voor gebruikers die zijn gemaakt bij het uitwerken van het programma `calibriv.exe` betreffen de formaten van de inputfiles.

De belangrijkste invoerfile is de parameterfile waarin de te calibreren stations en parameters en de geschiedenis van de calibratie worden gedefiniëerd. Voor dit bestand is gekozen voor een “komma-gescheiden” bestandsformaat (CSV-file). Dit formaat is gemakkelijk te interpreteren binnen `calibriv.exe`, en kan ook goed in Excel worden geïmporteerd en geëxporteerd.

Het bestand bevat 19 kolommen die worden gedefiniëerd in Tabel 2. Lege regels en commentaarregels worden genegeerd in de optimalisatieprocedure, maar worden wel teruggezet in de uitvoerfile. Alle regels die als eerste karakter anders dan spaties een “#” bevatten worden als commentaar beschouwd.

Restricties aan de waardes die in de parameterfile worden opgegeven zijn als volgt:

- De opgegeven stationsnamen moeten als controlestation voor de opgegeven grootheid in de simulatie-invoerfile zijn gedefiniëerd.
- De enige grootheid die vooralsnog kan worden gebruikt in de calibratie is “waterlevel”.
- De te calibreren ruwheidsparameter (`ruw_code`, parameter) moet voorkomen in het ruwheidsbestand, en moet (alleen) worden gebruikt in het traject benedenstrooms van het station dat bij deze parameter wordt gebruikt.

1	station	Naam van het te calibreren station
2	calibratie grootheid	Fysische grootheid in het station die moet worden geoptimaliseerd, m.n. “waterlevel”.
3	decimalen ruwheden	Het aantal cijfers achter de komma dat moet worden gebruikt voor het schrijven van ruwheden (“rough.karak”, kolom 13).
4	ruw_code	De ruwhedscode die wordt gebruikt voor het traject benedenstrooms van het te calibreren station (bijv. 411).
5	parameter	De parameter van de ruwhedscode die moet worden gecali- breerd (A, B, C of D)
6	decimalen tijdsries	Het aantal cijfers achter de komma voor gemeten en berekende waardes en daarvan afgeleide gegevens (kolommen 14 t/m 19).
7	starttijd	Starttijd van het tijdvenster per station waarvoor metingen en voorspellingen worden vergeleken, in uren ten opzichte van de referentiedatum van de simulatie.
8	eindtijd	Eindtijd van het tijdvenster, zie kolom 7.
9	max. afwijking	Maximaal toegestane afwijking tussen metingen en voorspelling- gen per station.
10	initiële aanpassing	Initiële procentuele aanpassing van de te calibreren parameter als er nog geen gevoeligheidsinformatie beschikbaar is, bijvoor- beeld 10 (%).
11	afhankelijkheid	Naam van het station waarvan het huidige afhankelijk is.
12	proc. afhankelijk	Percentage van de gemiddelde afwijking van het gerelateerde station (kolom 11) dat wordt verondersteld ook terug te komen in de gemiddelde afwijking voor het huidige station (kolom 1).
13	parameterwaarde	De waarde voor de parameter (kolommen 4 en 5) die in een iteratie is gebruikt.
14	gemeten waarde	Het gemiddelde van de gemeten waardes voor de calibratie grootheid (kolom 2) in het opgegeven tijdvenster (kolommen 7 en 8).
15	berekende waarde	Het gemiddelde van de berekende waardes voor de calibratie grootheid in het opgegeven tijdvenster, bij de parameterwaarde van kolom 13.
16	gemidd. afwijking	Gemiddelde van gemeten min berekende waarde over het tijd- venster.
17	standaardafwijking	Standaardafwijking van gemeten min berekende waarde in het tijdvenster.
18	richtingscoëfficiënt	De afhankelijkheid van de gemiddelde afwijking afw van de pa- rameterwaarde p wordt benaderd met de formule $afw = a \cdot p + b$. Hierin is a de richtingscoëfficiënt. Deze wordt bepaald uit ko- lommen 16 en 13 van de huidige en vorige iteraties.
19	intercept	Waarde van de coëfficiënt b uit de hierboven beschreven formule.

Tabel 2: Definitie van de kolommen van de calibratie parameterfile (invoer- en resultatenbe-stand).

- De enige parameter die momenteel kan worden gecalibreerd is parameter **a**.
- Het aantal decimalen dat in de programmatuur wordt gebruikt is tenminste 2 en maximaal 7. De breedte van kolommen 13 t/m 19 is steeds 8. Wanneer de te printen waarden te groot zijn voor het gevraagde format dan wordt het aantal decimalen achter de komma verminderd.
- Het tijdvenster dat wordt opgegeven moet overlappen met de tijden waarvoor time-histories worden berekend in de simulatie. Het is voordelig om dezelfde tijdstap in observaties en time-histories te gebruiken (zo veel mogelijk vergelijkingen) maar dit is niet verplicht.
- De initiële aanpassing moet tenminste 0.01% en maximaal 80% zijn.
- Het station waarvan een station afhankelijk is moet worden gedefiniëerd in de tabel. De naam mag niet gelijk zijn aan die van het huidige, afhankelijke station.
- Stationsnamen mogen meerdere keren voorkomen met verschillende te calibreren parameters. Het is hierbij echter niet zinvol om deze verschillende entries tegelijk “actief” te laten zijn. Voor alle entries moet een grote maximale afwijking worden gehanteerd (999.9) behalve voor de entry die in de huidige run moet worden gecalibreerd.

De volgorde van de stations (“entries”) in de tabel is vrij, ook wat betreft afhankelijke stations, en wordt zo veel mogelijk behouden. Ook commentaarregels worden behouden op de plek waar ze voorkwamen in het invoerbestand.

De gemeten waarden worden aangeleverd in een enkel bestand waarvan de naam wordt aangegeven via de optie `-observ`. In dit bestand worden meetreeksen gegeven per station. Voor ieder station wordt er eerst een regel gegeven met de stationsnaam, daarna volgen er meerdere regels met een tijdstip en een waarde. De tijdstippen worden opgegeven in aantal uren ten opzichte van de referentiedatum. Dus bij referentiedatum 1 januari komt het tijdstip 25.3333 overeen met 2 januari 1:20. Het observatiesbestand mag commentaar bevatten (“#-regels”).

Er wordt vooralsnog geeist dat de meetreeks regelmatig is, al mag ze wel gaten bevatten. Dit wordt als volgt gecontroleerd. Als tijdstap wordt praktisch gezien het minimum genomen van de tijd tussen iedere twee opeenvolgende tijdstippen in de meetreeks. (In het programma wordt er wat met deze tijdstap en de tijd tussen de eerste en laatste meting gemanipuleerd om minder last te hebben van afrondfouten.) De tijdstippen van de regelmatige reeks worden vervolgens op $t(1) + (i - 1) \cdot \delta t$ gesteld, met $t(1)$ het eerste tijdstip van de reeks, δt de tijdstap, en i het volgnummer binnen de regelmatige reeks. Alle waarden van de regelmatige reeks worden op een herkenbare dummywaarde geïntialiseerd. Tenslotte worden de opgegeven metingen gekopieërd naar de overeenkomstige tijdstippen. Hierbij wordt geeïst dat de opgegeven tijdstippen ten hoogste $0.01 \cdot \delta t$ afwijken van de tijdstippen van de regelmatige reeks.

De statistieken die in kolommen 14 t/m 17 in de parameterfile worden gepresenteerd worden berekend op basis van de “overeenkomstige tijdstippen binnen het gevraagde tijdvenster”. Dat

wil zeggen dat er per beschikbare meting in het tijdvenster wordt gekeken of er een voorspelling beschikbaar is, zo niet dan wordt de meting genegeerd. Bij het vergelijken van de tijdstippen van metingen en voorspellingen wordt hier een tolerantie van 0.1% van de kleinste tijdstap gehanteerd. Er wordt een waarschuwing gegeven als de tijdseries het gevraagde tijdvenster niet volledig overdekken (het is toegestaan om altijd tijdvenster [0, 999999] te gebruiken). Ook wordt er gewaarschuwd wanneer niet iedere meting aan een voorspelling kon worden gematched.

De stationsnamen die in het parameterbestand en het observatiesbestand worden gebruikt moeten gelijk zijn aan controlestations die in de simulatie-invoerfile en de SDS-file zijn gedefinieerd. Het vergelijken van namen is “case-insensitive”. Ook worden alle spaties in de namen bij het vergelijken genegeerd. De volgorde van de stations in het observatiebestand is niet van belang.

Tenslotte moet de initiële versie van het ruwheidsdefinitiebestand worden aangeleverd (“`rough.karak`”). Hierbij zijn we uitgegaan van bestaande versies van dit bestand. Een eerste eis die wordt gesteld is dat voor de te calibreren parameters de ruwheidscode op dezelfde regel staat als de ruwheidswaarde. Een tweede eis is dat deze regels correct zijn geformatteerd, zodat er na iedere parameternaam (`a`, `b`, `c` of `d`) een numerieke waarde volgt. Voor het overige worden er weinig eisen aan de inhoud en vormgeving van het bestand gesteld. Bijvoorbeeld is het niet verplicht om bij iedere ruwheidscode alle parameters `a` t/m `d` te specificeren. Bij het wegschrijven van het bestand wordt de initiële vormgeving zo goed mogelijk bewaard.

3 Technische keuzes in `calibriv.pl` en `calibriv.exe`

De belangrijkste keuze in de technische uitwerking van `calibriv.exe` is dat dit programma twee keer per iteratie wordt opgestart. Hiervoor zijn er twee aparte modes in het programma geprogrammeerd: “`pre_run`” en “`post_run`”. De run-procedure `calibriv.pl` voert nu het algoritme uit dat in Algoritme 1 wordt gepresenteerd.

De tweede belangrijke keuze bij de uitwerking van de programmatuur betreft de centrale data-structuren die worden gebruikt. Dit zijn de derived types “`t_caltable_entry`”, “`t_caltable`”, “`t_caltimeser`”, “`t_rough`”, en “`t_roughfile`”. De eerste twee of drie hiervan zijn het belangrijkste omdat ze op meerdere plekken in het programma worden gebruikt. De anderen worden meer lokaal gebruikt.

De types `t_caltable_entry` en `t_caltable` worden gebruikt voor het representeren van de parameterfile. Hierbij wordt de oorspronkelijke file in meerdere “entries” verdeeld. Een entry wordt gedefinieerd door de combinatie van een te calibreren station en te calibreren parameter. Merk op dat stations meerdere keren mogen voorkomen met verschillende parameters. De parameters moeten wel uniek zijn, maar zijn minder herkenbaar en zijn daarom niet geschikt om in print-output te worden gebruikt.

Een data-item van het type `t_caltable_entry` bevat alle data van een entry, voor alle iteraties die tot nu toe zijn uitgevoerd. Een entry beslaat dus meerdere regels uit de parameterfile.

Algorithm 1 - Uitwerking van de overall calibratie-loop in `calibriv.pl`

```
controleer argumenten van de procedure
zet iter=0
while ( not converged AND iter≤maxit ) do
  iter = iter + 1
  start calibriv.exe in pre_mode:
    determine whether all stations are converged,
    if not, create appropriate version of roughness file
  if ( not converged AND iter≤maxit ) then
    start run-procedure waqpre.pl for pre-processor WAQPRE
  end if
  if ( not converged AND iter≤maxit ) then
    start run-procedure waqpro.pl for simulation (WAQPRO)
  end if
  if ( not converged AND iter≤maxit ) then
    start calibriv.exe in post_mode:
      compute differences of time-series for all active stations,
      write additional information to parameterfile (parameter, statistics, etc.)
  end if
end while
```

De eerste regel van het entry bevat de statische data (kolommen 1 t/m 12, zie Tabel 2). De eerste en alle volgende regels geven de beschikbare informatie van iteraties 1, 2 en verder. Dit wordt in de data-structuur opgelost door per entry het aantal beschikbare waarden op te slaan (`nvals`, “aantal uitgevoerde iteraties”), en door arrays met lengte `nvals` te gebruiken voor de variabele eigenschappen: `param_val`, `obs_val`, `pred_val`, etc. Merk op dat `obs_val` (kolom 14) eigenlijk niet variabel is. Het is praktisch om deze waarde toch te printen op iedere regel. Hierdoor kan de tabel gemakkelijker worden bewerkt in Excel.

De data-structuur `tabel` van het type `t_caltable` bevat de totale inhoud van de tabel. Het belangrijkste hierin is het array `entries` waarmee de aparte entries worden gerepresenteerd. Verder bevat `tabel` vooral nog de oorspronkelijke inhoud van de parameterfile, inclusief het commentaar. Deze inhoud wordt gebruikt bij het wegschrijven van de aangepaste tabel.

Het type `t_caltimeser` wordt gebruikt voor het representeren van tijdseries. Een instantiatie van dit type wordt gebruikt voor de tijdserie van een enkel station. Deze structuur wordt zowel voor de observaties (array `obstimeser`) als voor de voorspellingen (array `sdstimeser`). Vooralsnog wordt er alleen met regelmatige tijdseries gewerkt. Hiervoor bevat een tijdserie de volgende velden: `statname`, `tfirst`, `tstep`, `tlast`, `rdumval` en `values(:)`. De stationsnaam wordt opgeslagen ter identificatie. Met name hoeven de tijdseries voor observaties en voor voorspellingen niet in dezelfde volgorde te worden gegeven. De voorspellingen worden

bij het inlezen wel gelijk in de volgorde van de entries gezet, mede omdat hetzelfde station meerdere keren mag voorkomen in de tabel, maar ook omdat een SDS-file veel meer stations kan bevatten dan er worden gecalibreerd. Het timeframe van een tijdserie wordt in minuten geadministreerd. De dummy-waarde wordt nog niet gebruikt. Ze zou kunnen worden toegepast voor het aanvullen van tijdseries die “bijna” regelmatig zijn. Het array `values` bevat tenslotte de waardes voor de tijdstippen `tfirst:tstep:tlast`.

Voor de verdeling van de verschillende procedures en functies over de verschillende broncode files wordt verwezen naar de `html`-pagina's die hiervoor zijn gegenereerd. Deze pagina's worden gemaakt met het tooltje `f90doc`. Hiermee worden bepaalde commentaar-regels uit de code gebruikt voor het genereren van een overzicht van de code.

4 Testen met de nieuwe procedure

Voor het testen van de nieuwe procedure is een schematisatie van een eenvoudig kanaaltje gebruikt. Het belangrijkste voordeel hiervan is dat simulaties snel kunnen worden uitgevoerd zodat er gemakkelijk veel verschillende configuraties (parameterfiles) kunnen worden uitgeprobeerd. Verder is het praktisch om weinig te calibreren stations te gebruiken omdat de uitvoer dan gemakkelijk kan worden nagerekend. Tenslotte kunnen met een klein modelletje ook de echte gevoeligheden en afhankelijkheden worden bepaald.

4.1 Beschrijving van het testmodel

Het testmodel dat is gebruikt is afgeleid van het testmodel “nikuradse_ddh”. Dit is een test om te onderzoeken of de k -Nikuradse berekening goed functioneert in DDHOR-berekeningen. Daarbij wordt er een enkele area gebruikt met een sterk afwijkende ruwheid die precies op de DDHOR-interface wordt gelegd. Als dit area niet goed wordt verwerkt in de berekening dan is het duidelijk in de simulatieresultaten te zien.

In het huidige project zijn we uitgegaan van de fijne versie van het rooster en worden alleen sequentiële runs uitgevoerd. Het kanaal is 10 km lang en 500 m breed en wordt met een roosterafstand van 166.7 m gediscetiseerd (60×3 interne cellen). Het totale verval over het hele kanaal is 1 m , ofwel de helling is 0.01% . Op de instroomrand op $M = 1$ wordt het debiet opgedrukt: $Q = 2500\text{ m}^3/\text{s}$. Op de uitstroomrand $M = 62$ wordt waterstand $H = 2.835\text{ m}$ gevraagd. Deze waarde komt uit een testje met Chezy-stroming met $C = 65\text{ m}/\text{s}^{1/2}$. In de huidige test wordt echter de Nikuradse-formulering met k -waardes van 0.01 en 0.1 gebruikt. Na vijf uur simulatie is een stationaire toestand bereikt.

Het kanaal is in drie trajecten verdeeld. Het eerste traject loopt van $M = 1$ tot 29 en gebruikt `ruw_code 211`. De k -waarde is hier constant 0.01 . Het tweede traject betreft $M = 30 - 32$ en gebruikt `ruw_code 403`. Het derde traject loopt van $M = 33$ tot 62 en gebruikt `ruw_code 213`. Deze laatste twee trajecten worden in verschillende testen afzonderlijk of tegelijk gecalibreerd. Hierbij wordt uitgegaan van “echte” ruwheidswaardes van respectievelijk 0.1 en 0.01 . Met deze waardes is een eerste simulatie uitgevoerd, en de tijdseries hiervan worden als “metingen”


```

links v. interface, waterlevel, 4, 403, a, 5, 0.00, 5.00, 0.00001, 5.00, ...
      0.2000, 3.65425, 3.69081, -0.03655, 0.03771, ,
      0.1900, 3.65425, 3.68760, -0.03335, 0.03440, 0.32105, -0.09435
      0.0950, 3.65425, 3.65203, 0.00222, 0.00229, 0.37441, -0.03335
      0.1009, 3.65425, 3.65463, -0.00038, 0.00039, 0.44127, -0.04490
      0.1000, 3.65425, 3.65424, 0.00001, 0.00003, 0.43185, -0.04317
      param , obsval , predval, meandif, stddev , rico , intercept

```

Tabel 3: *Uitvoerfile voor testrun “stat1” waarbij parameter 403 wordt gecalibreerd met metingen voor station links v. interface. Voor de leesbaarheid zijn lege kolommen en spaties verwijderd en is als laatste de betekenis van de kolommen toegevoegd.*

voor de calibratie-runs gebruikt. De stations die hiervoor worden gebruikt heten “links v. interface” en “rechts v. interface” en liggen op respectievelijk $M = 30$ en $M = 33$.

4.2 Testen met één calibratiestation

In de eerste paar testen wordt er slechts een parameter in de ruwheidsfile verstoord en wordt de echte waarde via calibratie gereconstrueerd. Bijvoorbeeld geeft Tabel 3 de resulterende tabel voor het geval dat de initiële ruwheid voor code 403 op 0.2 wordt gezet en dat er op de waterstanden van station “links v. interface” wordt gecalibreerd.

De tabel laat zien dat er op waterstanden wordt gecalibreerd, dat ruwheidswaardes met 4 decimalen worden geprint, dat het tijdvenster gelijk is aan de hele simulatieperiode (0-5 uur), dat er een hoge nauwkeurigheid wordt gevraagd (verschil obs-pred kleiner dan 10^{-5}), en dat de initiële verstoring 5% bedraagt. Er blijken zelfs voor deze hoge nauwkeurigheid slechts 5 iteraties nodig te zijn. De richtingscoëfficiënt varieert geleidelijk als functie van k en is zoals verwacht positief. De echte ruwheidswaarde wordt perfect gereconstrueerd.

Vergelijkbare problemen worden op dezelfde manier opgelost. Bijvoorbeeld kan ook het derde traject met parameter 213 met station “links v. interface” worden gecalibreerd. Voor het calibreren van parameter 213 met startwaarde 0.012 (echte waarde is 0.010) zijn zelfs slechts drie iteraties vereist. Hierbij blijkt een rol te spelen dat 0.010 de kleinste waarde is die mag worden gebruikt. De calibratieprocedure kiest in de derde iteratie 0.00990, WAQPRE neemt dan de minimale waarde 0.010 en de metingen worden exact gereproduceerd.

In `calibriv.exe` wordt bij het berekenen van nieuwe parameterwaardes wel enige begrenzing toegepast. In eerdere experimenten bleek een zeer kleine geschatte richtingscoëfficiënt namelijk tot niet-fysische parameterwaardes te kunnen leiden. Een probleem is echter dat de grenzen die in WAQPRE voor de parameters worden gebruikt afhankelijk zijn van de ruwheidsformulering en het ecotooptype die worden gebruikt. Het lijkt ons gewenst dat de grenzen die in WAQPRE worden gebruikt ook in `calibriv.exe` worden ingebouwd, of dat de grenzen per parameter worden toegevoegd als extra kolommen in de parameterfile.

Een probleem waarbij wordt geprobeerd parameter 403 (het tweede traject) te calibreren met de metingen van station “rechts v. interface” kan niet met `calibriv` worden opgelost. Het probleem is hier dat het controlestation benedenstrooms van het traject ligt terwijl bovenstroomse ligging van het station wordt gevraagd. Voor een station dat benedenstrooms van een traject ligt is de werkelijke richtingscoëfficiënt klein en negatief. Dit is ook zichtbaar in de resultaten van `calibriv`. Maar `calibriv` gaat ervan uit dat de werkelijke rico altijd positief is. In geval van een negatieve rico past hij daarom de initiële stapgrootte toe. En hij denkt daarbij dat een positief verschil “obs-pred” verhoging van de ruwheid vergt. Dat is in dit geval niet correct, en daardoor divergeert het proces.

Een oplossing hiervoor zou kunnen zijn om ook negatieve richtingscoëfficiënten als “waar” te beschouwen. In dat geval convergeert het hierboven beschreven probleem wel (in 5 iteraties). Maar dan worden situaties waarin de richtingscoëfficiënt niet goed wordt geschat (bijvoorbeeld door een verkeerd opgegeven afhankelijkheid) mogelijk niet meer goed afgehandeld. De vraag is dus tegen welke gebruikersfouten de meeste bescherming is gewenst.

4.3 Testen met twee calibratiestations

Vervolgens zijn testen uitgevoerd waarbij alle twee de stations tegelijk worden gecalibreerd. Na het debuggen van de precieze formules die moeten worden gebruikt en na kleine bijstellingen in de methode werkt dit ook probleemloos. Dit wordt geïllustreerd in Tabel 4 voor het geval dat de initiële waardes van de parameters 403 en 213 worden ingesteld op respectievelijk 0.20 en 0.015.

De tabel laat zien dat allerlei attributen per station/parameter kunnen worden gevarieerd. Bijvoorbeeld het aantal decimalen waarmee resultaten worden geprint, de nauwkeurigheid die wordt gevraagd per station en de initiële stapgrootte. Verder wordt geïllustreerd hoe een afhankelijkheid wordt gespecificeerd. Het station “rechts v. interface” ligt het meest stroomafwaarts en is daarom het onafhankelijke station. Het station “links v. interface” ligt net bovenstrooms hiervan en wordt daarom beïnvloed door aanpassing van ruwheidswaarde 213. In dit geval is de afhankelijk op 70% gezet.

Door middel van twee extra testen met de echte invoerfile is de werkelijke afhankelijkheid bepaald. Hiervoor worden de resultaten vergeleken van runs met ruwheden $(213,403) = (0.01, 0.10)$, $(0.0105, 0.10)$ en $(0.01, 0.105)$. De twee echte ruwheden worden dus onafhankelijk van elkaar 5% verstoord. Met drie metingen voor twee stations kunnen alle zes de parameters van een lineair model worden gefit:

$$\underline{y} = A\underline{p} + \underline{b}, \quad y = \begin{pmatrix} y_{rechts} \\ y_{links} \end{pmatrix}, \quad p = \begin{pmatrix} p_{213} \\ p_{403} \end{pmatrix}, \quad A = \begin{pmatrix} 9.665 & 0.00172 \\ 8.720 & 0.432 \end{pmatrix} \quad (1)$$

Deze resultaten laten zien dat de waterstand rechts (y_{rechts}) inderdaad vrijwel niet afhankelijk is van de parameter links (p_{403}). Omgekeerd is de waterstand links wel gevoelig voor de parameter rechts. Het effect van een aanpassing van parameter rechts (p_{213}) op het linker station (y_{links}) is in dit geval 90.2% ($8.720/9.665$) van het effect dat de aanpassing in het

```

rechts v. interface, waterlevel, 4, 213, a, 5, 0.00, 5.00, 0.00001, 5.00, ...
      0.0150, 3.51561, 3.55848, -0.04286, 0.04518,
      0.0142, 3.51561, 3.55307, -0.03745, 0.03921, 6.76637, -0.13353
      0.0087, 3.51561, 3.51609, -0.00047, 0.00204, 6.72383, -0.05897
      0.0086, 3.51561, 3.51566, -0.00005, 0.00022, 4.26607, -0.03674
      0.0086, 3.51561, 3.51562, 0.00000, 0.00004, 4.74095, -0.04077
      0.0086, 3.51561, 3.51561, 0.00000, 0.00004, 4.77019, -0.04102

links v. interface, waterlevel, 5, 403, a, 6, 0.00, 5.00, 0.00003, 30.00, ...
      rechts v. interface, 70.000, ...
      0.20000, 3.650836, 3.725479, -.074643, 0.077391,
      0.14000, 3.650836, 3.700226, -.049390, 0.051361, 0.357729, -0.09947
      0.07522, 3.650836, 3.639242, 0.011595, 0.011961, 0.541799, -0.02916
      0.09723, 3.650836, 3.649606, 0.001230, 0.001269, 0.484453, -0.04587
      0.09984, 3.650836, 3.650754, 0.000082, 0.000088, 0.451624, -0.04501
      0.10002, 3.650836, 3.650833, 0.000004, 0.000023, 0.463735, -0.04638
param , obsval , predval, meandif, stddev , rico , intercept

```

Tabel 4: *Uitvoerfile voor testrun “2stats” waarbij parameters 403 en 213 worden gecali-
breerd met metingen voor stations links v. interface en rechts v. interface. Voor de
leesbaarheid zijn lege kolommen en spaties verwijderd en is als laatste de betekenis van de
kolommen toegevoegd.*

rechter station heeft (y_{rechts}). Tenslotte is opvallend dat de waterstand links sterker afhangt van parameter p_{213} dan van zijn eigen parameter p_{403} .

Vanwege het laatste aspect hebben we geëxperimenteerd met verschillende formules voor het schatten van de richtingscoëfficiënt voor het afhankelijke station. Het corrigeren voor de afhankelijkheid geeft hier een gunstig effect op de richtingscoëfficiënt. Op het aantal uitgevoerde iteraties is het effect echter niet duidelijk te zien.

De correctie die wordt gebruikt is als volgt. Er worden twee iteraties met elkaar vergeleken voor het afhankelijke station. Enerzijds hebben we hier de verschillen “obs-pred”, anderzijds de parameterwaardes. De recht-toe-recht-aan berekening van de richtingscoëfficiënt is om het verschil in de fout obs-pred te delen door het verschil in de parameter. Maar de fout obs-pred wordt mede door het niet-afhankelijke station bepaald. We nemen aan dat de verandering van de fout obs-pred in het niet-afhankelijke station tussen de twee beschouwde iteraties volgens het opgegeven percentage doorwerkt in het afhankelijke station:

$$dy = perc \cdot (y_{rechts}^{(2)} - y_{rechts}^{(1)}), \quad rico = (y_{links}^{(2)} - y_{links}^{(1)} - dy) / (p_{links}^{(2)} - p_{links}^{(1)})$$

De snelle convergentie is vrij typisch voor het onderzochte probleem. Alleen als de twee parameters extreem worden verstoord (bijv. $(213, 403) = (0.10, 0.01)$) dan loopt het aantal iteraties op tot 10 of meer. Hierbij blijkt de convergentie vrij ongevoelig te zijn voor het percentage dat voor de afhankelijkheid wordt opgegeven. Wanneer dit percentage op 91%

wordt gezet dan worden er vijf iteraties uitgevoerd, maar ook bij 1% zijn er slechts vijf iteraties vereist. In dit laatste geval speelt een rol dat we hebben ingebouwd dat een parameter per iteratie niet sterker mag worden aangepast dan halvering of verdubbeling. Dat het gekoppelde systeem zo snel convergeert lijkt verder samen te hangen met het eerder opgemerkte effect dat de echte ruwheid voor parameter 213 (0.01) gelijk is aan de minimale waarde die wordt toegestaan door WAQPRE.

Dat een parameter per iteratie niet meer mag worden aangepast dan halvering of verdubbeling is voor het verhogen van de robuustheid bedoeld. Door deze begrenzing van de relatieve aanpassing worden de gevolgen van een slechte schatting van de richtingscoëfficiënt sterk ingeperkt. Er kunnen niet langer volledig arbitraire parameterwaarden worden geprobeerd. Deze begrenzing heeft geen invloed op het uiteindelijke resultaat van de calibratie, maar beïnvloedt alleen de weg die wordt gekozen om dit resultaat te bereiken. In de opgeleverde versie van de programmatuur is deze begrenzing vooralsnog uitgeschakeld.

Een probleem dat is opgemerkt bij deze testen ontstaat wanneer er voor afhankelijke stations verschillende maximaal toegestane afwijkingen worden gebruikt. Met name wanneer er voor het afhankelijke station een strenger stopcriterium wordt gebruikt dan voor het niet-afhankelijke station. De fout in het afhankelijke station moet namelijk ook via aanpassing van de parameter van het niet-afhankelijke station worden gecompenseerd. Maar omdat het niet-afhankelijke station als geconvergeerd wordt beschouwd worden er daar geen aanpassingen gemaakt. In zo'n geval kan het maximale aantal iteraties worden uitgevoerd zonder dat het afhankelijke station convergeert. Dit wordt voorkomen wanneer voor alle actieve stations dezelfde maximale afwijking wordt gehanteerd.

Als laatste stellen we verfijningen van de optimalisatieprocedure voor voor de afhankelijke stations. Deze verfijningen zijn gerelateerd aan het fitten van een lineair model zoals gepresenteerd in vergelijking (1). De vergelijking laat zien dat de exacte afhankelijkheid kan worden geschat op basis van drie iteraties. Dit zou in de procedure kunnen worden verwerkt.

Een andere mogelijkheid is om na de eerste run met de initiële parameters alleen de niet-afhankelijke parameters te verstoren. Hiermee kan na de tweede iteratie de procentuele afhankelijkheid worden bepaald.

Tenslotte is een mogelijkheid om het systeem zelf te laten bepalen welke stations van elkaar afhankelijk zijn. Hiervoor worden de stations in de richting van de rivier gesorteerd (1=benedenstrooms). Na de initiële run worden eerst de parameters van stations met oneven nummers verstoord, en in de run daarna worden de andere parameters verstoord. Op basis van deze drie runs en de aanname dat lager genummerde stations niet van hoger genummerde stations afhankelijk zijn kan dan een goede schatting van de hele Jacobiaan van het systeem worden gemaakt.

5 Conclusies en aanbevelingen

In dit rapport wordt verslag gedaan van de ontwikkeling van hulpprogramma's voor het calibreren van WAQUA riviermodellen: run-procedure `calibriv.pl` en Fortran90 programma

`calibriv.exe`. Met deze programma's wordt een hoop handwerk bij het afregelen van bodemruwheden in riviertrajecten geautomatiseerd.

Dit memo geeft een overzicht van de aansturing van de procedure en de verwachtingen die ten aanzien van de invoerfiles worden gesteld. Verder worden de belangrijkste data-structuren van het programma toegelicht. Tenslotte wordt over de uitgevoerde testen met een schematisch kanaal gerapporteerd. Met deze testen is een groot aantal details in de werking van de programma's onderzocht (fouten in stationsnamen, tijdvensters, berekening van statistieken e.d.). Ook is de correcte werking van de optimalisatieprocedure getest voor verschillende problemen, met name ook voor "afhankelijke stations".

Op basis van de testen zijn er een paar suggesties voor verdere verbeteringen gedaan:

1. Het programma `calibriv.exe` weet nu niet binnen welke grenzen de parameters mogen worden aangepast. Dit leidt tot vertraging van het convergentieproces wanneer `calibriv.exe` niet weet dat er in de simulatie andere waardes dan voorgeschreven worden gebruikt. Om dit tegen te gaan zouden de grenzen per parameter in extra kolommen in de parameterfile kunnen worden gezet, of kunnen de grenzen die per ruwheidscode in WAQPRE worden gebruikt hard in `calibriv.exe` worden gecodeerd.
2. De voorgeschreven procedure voor het omgaan met negatieve richtingscoëfficiënten zorgt ervoor dat het programma alleen geschikt is voor stations bovenstrooms van een te calibreren traject. Dit kan via aanpassing van de procedure worden verholpen. Een mogelijk nadeel hiervan is wel dat de procedure gevoeliger wordt voor onregelmatigheden in de berekende waterstanden. In de tot nu toe uitgevoerde experimenten was dat echter geen probleem.
3. Tenslotte zijn suggesties gedaan voor het vergemakkelijken van het gebruik van de procedure voor van elkaar afhankelijke stations. Met een kleine aanpassing van de procedure (initiële aanpassing eerst alleen toepassen voor niet-afhankelijke stations) kan de procentuele afhankelijkheid na twee iteraties worden geschat in plaats van dat ze door de gebruiker moet worden gespecificeerd. Met wat ingrijpendere aanpassingen kan zelfs automatisch worden bepaald welke stations van elkaar afhankelijk zijn.

Andere suggesties die naar voren zijn gekomen uit de bespreking met opdrachtgever zijn:

4. Momenteel wordt een aantal controles op de correctheid van de gebruikersinvoer pas gedaan nadat de eerste simulatie is uitgevoerd. Het is gewenst dat het programma `calibriv.exe` wordt uitgebreid met een optie om de gebruikersinvoer te controleren. Hierbij kunnen de foutmeldingen uit het programma mooier en duidelijker worden gemaakt door gebruik van SIMONA's subroutine `sierro` en `error-text-file`.
5. Een kleine wens is om het iteratienummer toe te voegen in een extra kolom van de parameterfile.

6. Voor het volgen/controleren van de berekeningen zou het handig zijn als de metingen en voorspellingen die in `calibriv.exe` worden gebruikt naar een aparte file zouden kunnen worden uitgevoerd.