

MEMO                    BvtH/M07.047  
Datum                    15 juli 2008  
Auteur(s)                Bas van 't Hof  
Onderwerp                Optimale koppeling in  
                              domein-decompositieberekeningen

## Documentinformatie

Versie	Auteur	Datum	Opmerkingen	Review
0.1	BvtH	12-07-2007	Concept	
0.2	BvtH	22-07-2007	Correcties n.a.v. review	
0.4	EV	29-08-2007	Aanpassingen n.a.v. discussie BvtH	
0.5	BvtH	21-01-2008	Aanpassingen n.a.v. test csm-zuno	
1.0	BvtH	18-03-2008	Aanpassingen n.a.v. correctie csm-zuno	EV,ES
1.1	BvtH, JG	18-03-2008	Aanpassingen n.a.v. review ES	
2.0	BvtH	08-07-2008	Opname in de moederversie	JG
Bestandslocatie:    /v3/E05q_bo_simona/c74231-dd-uitg-info/report				

## 1 Inleiding

In het kader van overeenkomst RKZ-1569, “*performanceverbetering domein decompositie*”, heeft VORtech met assistentie van WL|Delft Hydraulics onderzoek gedaan naar de mogelijkheden van het koppelen van domeinen, waarbij de verschillende domeinen verschillende tijdstappen aanhouden. Hierbij is het idee van “uitgaande informatie” bedacht. Uitgaande informatie betreft informatie van een subdomein die (met zekere nauwkeurigheid) al aan het begin van een tijdstap kan worden berekend, terwijl ze de nog komende (halve) tijdstap betreft.

In hetzelfde project is vastgesteld dat de methode van uitgaande informatie ook kan worden gebruikt om berekeningen te versnellen waarin horizontale verfijning met gelijke tijdstappen wordt gebruikt. De performance van deze DDHOR-berekeningen met de huidige programmatuur blijft vooralsnog achter bij de performance van 1:1 koppelingen of bij het nesten van grids (off-line koppeling). Dat komt doordat bij horizontale verfijning de Jacobi-methode wordt gebruikt voor het koppelen van de waterstanden, welke veel langzamer convergeert dan het BJ-TWGE algoritme dat voor 1:1 koppelingen en parallel rekenen wordt gebruikt. Het

BJ-TWGE algoritme kon tot dusver niet naar de situatie met verfijning worden gegeneraliseerd.

Binnen het contract RKZ-1569 is een prototype gemaakt van WAQUA/TRIWAQ waarin de methode van uitgaande informatie wordt toegepast bij domein decompositie met horizontale verfijning. Hierbij zijn diverse moeilijkheden opgetreden in de ontwikkeling van de methode, vooral met betrekking tot de keuze van de juiste iteratieniveaus van allerlei grootheden op de rand. Deze moeilijkheden zijn overwonnen en er wordt snelle convergentie mee bereikt. De methode leverde in de gehanteerde testgevallen acceptabele resultaten op. Deze resultaten verschillen wel iets van de oorspronkelijke methode voor DDHOR, omdat er in de nieuwe koppeling andere “koppelingscondities” worden gehanteerd.

In melding c66693 van het SIMONA beheer en onderhoud is het gedrag van dit prototype in enkele bijzondere gevallen onderzocht. Hierbij is gebleken dat de nieuwe koppelingscondities soms zeer ongewenste resultaten geven. De oorzaak bleek te zijn dat de waterstanden langs de interface stuksgewijs constant worden geïnterpoleerd, wat leidt tot een vlakke waterspiegel in de meeste roosterpunten, en een zeer groot verhang in de rest van de roosterpunten. Daarvoor is een verfijning van de methode bedacht waarmee het verhang in de richting langs de interface in de koppelingscondities kan worden verdisconteerd.

Op basis van de resultaten van contract RKZ-1569 en van melding c66693 heeft RIKZ gevraagd om een complete implementatie te maken van de methode van uitgaande informatie. Dit werk wordt uitgevoerd onder change nummer c74231 van het SIMONA B&O contract. In deze change worden de volgende drie aspecten verder uitgewerkt:

- Overnemen van de methode uit het prototype in de moederversie van de SIMONA-programmatuur. De moederversie is inmiddels sterk aangepast, met name zijn de rekenroutines van WAQUA en TRIWAQ met elkaar samengevoegd.
- Uitbreiden van de methode met een mechanisme waarmee het verhang langs de interface kan worden verdisconteerd.
- Uitbreiden van de implementatie van de methode zodanig dat er geen aparte stelsels hoeven te worden geveegd voor het bepalen van de coëfficiënten van de optimale koppeling en voor het bepalen van de waterstanden.

Bij het onderzoeken van het derde punt is er een manier gevonden waarmee de methode van uitgaande informatie als een generalisatie van BJ-TWGE kan worden gezien. Voor 1:1 koppeling is de nieuwe methode aan BJ-TWGE equivalent en kan daarom ook de nieuwe methode worden gebruikt. Het is daarom niet meer nodig om twee verschillende methoden te handhaven in de programmatuur.

De wiskundige aspecten van de nieuwe methode en de benodigde algoritmieken worden gepresenteerd in Sectie 2.

Bij de implementatie is er speciale aandacht besteed aan het scheiden van de verschillende onderdelen van de methode in verschillende subroutines. De implementatie wordt gepresenteerd

in Sectie 4.

In Sectie 5 worden ten slotte de resultaten gepresenteerd van enkele simulaties: sequentiële, parallelle berekeningen en berekeningen met horizontale verfijningen. Hierbij wordt zowel gekeken naar de berekende waterstanden en stroomsnelheden, als naar de rekestijden.

## 2 Koppeling met BJ-TWGE en met uitgaande informatie

### 2.1 Stelsels van gekoppelde impuls- en continuïteitsvergelijkingen

Een belangrijk onderdeel van een berekening met WAQUA is het oplossen van de gekoppelde impuls- en continuïteitsvergelijking. Door voldoende termen expliciet te nemen wordt een impulsvergelijking verkregen met de volgende vorm:

$$aak_{m,n,k} s_{m,n} + u_{m,n,k} + cck_{m,n,k} s_{m+1,n} = ddk_{m,n,k}. \quad (1)$$

waarin

$m$	$m$ -coördinaat
$n$	$n$ -coördinaat
$k$	$k$ -coördinaat
$s_{m,n}$	waterstand in het roosterpunt $(m, n)$
$u_{m,n,k}$	$u$ -stroomsnelheid in het roosterpunt $(m, n, k)$
$aak_{m,n,k}, cck_{m,n,k}, ddk_{m,n,k}$	coëfficiënten in de vergelijking.

Hieruit wordt de diepte-geïntegreerde impulsvergelijking berekend:

$$aa_{m,n} s_{m,n} + qx_{m,n}/guu_{m,n} + cc_{m,n} s_{m+1,n} = dd_{m,n}, \quad (2)$$

waarin

$qx_{m,n}$	debiet in het roosterpunt $(m + 1/2, n)$
$guu_{m,n}$	lengte van de cell-face $(m + 1/2, n)$
$aa_{m,n}, cc_{m,n}, dd_{m,n}$	coëfficiënten in de vergelijking.

Door de geïntegreerde impulsvergelijking in te vullen in de continuïteitsvergelijkingen, wordt een tridiagonaal stelsel verkregen. De vergelijkingen worden gelineariseerd, waarna ze er als volgt uitzien:

$$a_{m,n} s_{m-1,n} + b_{m,n} s_{m,n} + c_{m,n} s_{m+1,n} = d_{m,n}, \quad (3)$$

waarin

$a_{m,n}, b_{m,n}, c_{m,n}, d_{m,n}$	coëfficiënten in de vergelijking.
--------------------------------------	-----------------------------------

Het probleem bij domein decompositie is dat er waarden in het stelsel voorkomen die net buiten het eigen domein liggen. De punten  $(mf + 1 : ml, n)$  liggen in het eigen domein, en de punten  $(mf, n)$  en  $(ml + 1, n)$  liggen dus net buiten het eigen domein. De waterstanden in deze punten komen voor in het stelsel (3).

## 2.2 Koppelingscondities in de huidige DDHOR-methode

In de huidige programmatuur worden de vergelijkingen op de volgende manier gekoppeld tussen verschillende domeinen:

- de waarden  $s_{mf,n}$  en  $s_{m1+1,n}$  net buiten het eigen domein worden verkregen via interpolatie van waarden uit het buurdomein, soms aangevuld met waarden uit het eigen domein. De gebruikte interpolatie-formules vormen hierbij extra vergelijkingen waarmee de waterstanden in interpolatiepunten in de interne waterstanden worden uitgedrukt.
- De debieten  $qx_{mf,n}$  en  $qx_{m1,n}$  op de interface van het grove domein worden verkregen via interpolatie (sommatie) van de debieten van het fijne domein, dat de *eigenaar* van de interface is.

Deze aanvullende vergelijkingen vormen de zogenaamde *koppelingscondities* voor het koppelen van de stelsels (2) en (3) voor verschillende domeinen. Aan deze vergelijkingen wordt voldaan zodra het gebruikte iteratieproces is geconvergeerd. In het iteratieproces worden net iets andere vergelijkingen gehanteerd:

- Voor de waterstanden in interpolatiepunten worden waarden van de vorige iteratie  $[q-1]$  gebruikt. Dit is de zogenaamde Jacobi-koppeling.
- Voor de debieten in interfacepunten van het grove domein wordt een combinatie van een impulsvergelijking en de debietvergelijking gebruikt. Er wordt een impulsvergelijking in de interfacepunten bepaald die aangeeft hoe het debiet ongeveer met de waterstanden net binnen en net buiten het domein varieert. Deze impulsvergelijking wordt gebruikt als preconditionering van de vergelijking waarmee het debiet van het grove domein wordt opgelegd. Door deze toevoeging wordt snellere convergentie bereikt dan wanneer alleen het debiet van het fijne domein wordt opgelegd.

## 2.3 De BJ-TWGE methode voor situaties zonder verfijning

In de huidige programmatuur wordt bij 1:1 koppeling in het horizontale vlak de BJ-TWGE methode gebruikt. Deze methode is zeer efficiënt omdat nieuwe informatie van een domein binnen een iteratie over veel grotere afstanden wordt getransporteerd dan wanneer een Jacobi-koppeling wordt gebruikt. De nieuwe methode op basis van uitgaande informatie is ontworpen als een aanpassing van BJ-TWGE, met als doel om hetzelfde principe ook bij horizontale verfijning mogelijk te maken. Als inleiding op de uitleg van de nieuwe methode wordt daarom eerst de BJ-TWGE methode uitgelegd.

Wanneer een domein slechts aan één kant is gekoppeld aan een ander domein, kan het stelsel (3) worden geveegd tot een bovendriehoeks-stelsel van de vorm

$$b_{2,m,n} s_{m,n} + c_{2,m,n} s_{m+1,n} = d_{2,m,n}, \quad (4)$$

of tot een onderdriehoeks-stelsel van de vorm

$$a_{3_{m,n}} s_{m-1,n} + b_{3_{m,n}} s_{m,n} = d_{3_{m,n}}. \quad (5)$$

In het geval dat het domein aan beide uiteinden gekoppeld is kan een dergelijk stelsel worden verkregen door aan een van beide einden de waterstand voor te schrijven met de beste waarde die daarvoor beschikbaar is (Jacobi-koppeling). Vaak heeft het invullen van deze waterstand alleen gevolgen voor de resultaten daar dicht bij in de buurt, en niet voor het andere einde van het domein.

Bij een 1:1 koppeling kan de verkregen vergelijking (4) in het punt  $(m1, n)$  worden gebruikt door het aangrenzende buurdomein. Dit is op overeenkomst van de volgende punten gebaseerd:

$$\tilde{s}_{m\tilde{f},\tilde{n}} = s_{m1,n} \quad , \quad \tilde{s}_{m\tilde{f}+1,\tilde{n}} = s_{m1+1,n}, \quad (6)$$

waarbij de waarden van het aangrenzende domein met een tilde zijn aangegeven.

Net zo kan vergelijking (5) van het buurdomein worden verkregen en in het punt  $(m1 + 1, n)$  worden gebruikt. Zo krijgt het domein de beschikking over de twee vergelijkingen

$$\begin{aligned} b_{2_{m1,n}} s_{m1,n} + c_{2_{m1,n}} s_{m1+1,n} &= d_{2_{m1,n}}, \\ \tilde{a}_{3_{m\tilde{f}+1,\tilde{n}}} s_{m1,n} + \tilde{b}_{3_{m\tilde{f}+1,\tilde{n}}} s_{m1+1,n} &= \tilde{d}_{3_{m\tilde{f}+1,\tilde{n}}}. \end{aligned} \quad (7)$$

Hiermee kunnen de waterstanden aan het einde van dit domein worden berekend, en daarna de rest van de waterstanden. In een simulatie waarin domeinen slechts aan een kant zijn gekoppeld wordt zo het globale lineaire stelsel in een keer opgelost. In het geval dat de domeinen aan beide kanten zijn gekoppeld, wordt een ingewikkeld schema gebruikt waaruit voor iedere iteratie de juiste veegrichting wordt bepaald zodanig dat een snelle convergentie wordt bereikt (zie Sectie 4).

### 3 De koppelingsmethode op basis van uitgaande informatie

De methode van uitgaande informatie is gebaseerd op de gedachte dat bepaalde lineaire combinaties van grootheden, bijvoorbeeld waterstand en snelheid of waterstand en debiet, aan het einde van een rij ongevoelig zijn voor de informatie die van het buurdomein afkomt. In [1] is aangetoond dat er een coëfficiënt  $\alpha$  voor zulke lineaire combinaties kan worden bepaald en dat daarmee snelle algoritmes kunnen worden gemaakt. In dit huidige werk gaan we een stap verder, en combineren we de gedachte van uitgaande informatie met BJ-TWGE. Het idee is nu dat de optimale  $\alpha$  impliciet voorkomt in de geveegde stelsels (4) en (5). We bepalen de uitgaande informatie van ieder domein nu uit deze stelsels.

Van de andere kant kan ook worden gezegd dat BJ-TWGE hiermee wordt veralgemeniseerd. Zoals gezegd in paragraaf 1 kunnen de vergelijkingen (4) en (5) bij horizontale verfijning niet betekenisvol worden gecommuniceerd. Het is bijvoorbeeld niet duidelijk hoe de coëfficiënten  $a_2$  enzovoort moeten worden geïnterpoleerd. Dit wordt nu mogelijk gemaakt doordat deze vergelijkingen in termen van debieten en waterstanden worden geformuleerd.

### 3.1 Uitgaande informatie

De nieuwe koppelmingsmethode op basis van uitgaande informatie is met het nodige gepuzzel tot stand gebracht, zie [1]. Daarbij is gevariëerd met de koppelmingscondities die worden gebruikt en met de combinaties van grootheden die voor de uitgaande informatie wordt gebruikt.

In de uiteindelijke vorm is er gekozen voor uitgaande informatie op basis van debieten en waterstanden:

$$q\mathbf{x}_{m,n} + q2sep_{m,n} \mathbf{su}_{m,n} = \mathbf{outgo}_{m,n}, \quad (8)$$

$$\mathbf{su}_{m,n} = \mathbf{tetau}_{m,n} \mathbf{s}_{m,n} + (1 - \mathbf{tetau}_{m,n}) \mathbf{s}_{m+1,n} \quad (9)$$

waarin

- $q2sep_{m,n}$  coëfficiënt ( $\alpha$ ) in de berekening van  $\mathbf{outgo}$
- $\mathbf{su}_{m,n}$  van waterstandspunten naar snelheidspunten geïnterpoleerde waterstand in het punt  $(m + 1/2, n)$
- $\mathbf{outgo}_{m,n}$  uitgaande informatie ( $r$ ) in het snelheidspunt op de interface  $(m + 1/2, n)$ .

De vraag is nu voor welke waarde van  $q2sep_{m,n}$  vergelijking (8) inderdaad uitgaande informatie betreft, wat wil zeggen dat de waarde van  $\mathbf{outgo}_{m,n}$  niet van grootheden van het buurdomein afhankelijk is. Als we deze waarde van  $q2sep$  hebben, dan kunnen we de uitgaande informatie  $\mathbf{outgo}$  berekenen en toesturen aan het buurdomein. In [1] werd de waarde voor  $q2sep$  bepaald door een extra stelsel te vegen met een speciaal rechterlid. Hiermee werd een zogenaamde “mode” bepaald, en uit deze mode volgde de optimale  $q2sep$ .

In de huidige change is een verbetering ten opzichte van deze methode bedacht waardoor er geen extra stelsel meer hoeft te worden geveegd. De crux van de nieuwe aanpak zit in de observatie dat we al informatie hebben over het einde van een domein. Namelijk de geveegde vergelijking (4) of (5). We kennen niet de waarde van  $\mathbf{s}_{m+1,n}$ , maar wel een vergelijking waarmee deze in  $\mathbf{s}_{m,n}$  wordt uitgedrukt. Alle coëfficiënten hierin ((4):  $b2$ ,  $c2$ ,  $d2$ ) zijn bekend, en zijn bepaald zonder gebruik te maken van informatie van het buurdomein. Als we nu (8) van dezelfde vorm maken als (4) dan hebben we daarmee de goede verhouding van  $q\mathbf{x}$  en  $\mathbf{su}$  ten behoeve van uitgaande informatie bepaald.

We gaan uit van de gewenste vorm voor de koppelmingsvergelijking (8). In het linker lid schrijven we  $q\mathbf{x}$  uit met behulp van de diepte-geïntegreerde impulsvergelijking (2), en  $\mathbf{su}$  met zijn definitie (9):

$$\begin{aligned} & \mathbf{guu}_{m,n} (\mathbf{dd}_{m,n} - \mathbf{aa}_{m,n} \mathbf{s}_{m,n} - \mathbf{cc}_{m,n} \mathbf{s}_{m+1,n}) + \\ & q2sep_{m,n} (\mathbf{tetau}_{m,n} \mathbf{s}_{m,n} + (1 - \mathbf{tetau}_{m,n}) \mathbf{s}_{m+1,n}) = \mathbf{outgo}_{m,n} \end{aligned} \quad (10)$$

We groeperen de termen met waterstanden  $\mathbf{s}_{m,n}$  en  $\mathbf{s}_{m+1,n}$ , en eisen dat de coëfficiënten hiervan dezelfde verhouding hebben als  $b2$  en  $c2$  in (4):

$$\frac{q2sep_{m,n} \mathbf{tetau}_{m,n} - \mathbf{guu}_{m,n} \mathbf{aa}_{m,n}}{q2sep_{m,n} (1 - \mathbf{tetau}_{m,n}) - \mathbf{guu}_{m,n} \mathbf{cc}_{m,n}} = \frac{b2}{c2} \quad (11)$$

Deze waarde is

$$q2sep_{m,n} = \frac{c2_{m,n}aa_{m,n} - b2_{m,n}cc_{m,n}}{c2_{m,n}tetau_{m,n} - b2_{m,n}(1 - tetau_{m,n})} guu_{m,n} \quad (12)$$

of een vergelijkbare waarde, als niet de vergelijking (4) maar de vergelijking (5) beschikbaar is. Met deze waarde is vergelijking (8) equivalent gemaakt met (4) of met (5). Het verschil met die geveegde vergelijkingen is echter dat (8) wél betekenisvol kan worden gecommuniceerd van het fijne naar het grove buurdomein.

Voor een koppeling aan het begin van een rij gaan we uit van de vergelijkingen:

$$a3_{mfu,n} s_{mf,n} + b3_{mfu,n} s_{mfu,n} = d3_{mfu,n}, \quad (13)$$

$$su_{mf,n} = tetau_{mf,n}s_{mf,n} + (1 - tetau_{mf,n})s_{mfu,n} \quad (14)$$

$$qx_{mf,n} + q2sep_{mf,n}su_{mf,n} = outgo_{mf,n} \quad (15)$$

We nemen vergelijking (15) en vullen hierin de impulsvergelijking (2) en de interpolatie van de waterstand (14) in. Dit geeft

$$outgo_{mf,n} = guu_{mf,n} (dd_{mf,n} - aa_{mf,n} s_{mf,n} - cc_{mf,n} s_{mfu,n}) + q2sep_{mf,n} (tetau_{mf,n}s_{mf,n} + (1 - tetau_{mf,n})s_{mfu,n}) \quad (16)$$

We groeperen de termen met waterstanden  $s_{mf,n}$  en  $s_{mfu,n}$ , en eisen dat de coëfficiënten hiervan dezelfde verhouding hebben als  $a3$  en  $b3$  in (13):

$$\frac{q2sep_{mf,n} tetau_{mf,n} - guu_{mf,n} aa_{mf,n}}{q2sep_{mf,n} (1 - tetau_{mf,n}) - guu_{mf,n} cc_{mf,n}} = \frac{a3}{b3} \quad (17)$$

Dit geeft als oplossing voor  $q2sep$ :

$$q2sep_{m,n} = \frac{b3_{m,n}aa_{m,n} - a3_{m,n}cc_{m,n}}{b3_{m,n}tetau_{m,n} - a3_{m,n}(1 - tetau_{m,n})} guu_{m,n} \quad (18)$$

### 3.2 Koppelingscondities

De *koppelingscondities* zijn de eisen waaraan de oplossing moet voldoen, wanneer de vergelijkingen in de verschillende domeinen allemaal opgelost zijn. In [1] is er met deze koppelingscondities gevarieerd om een goede methode te bereiken. Niet alle koppelingscondities kunnen namelijk met de verschillende vormen van uitgaande informatie worden gecombineerd. Met name is het niet gelukt om de koppelingscondities van de huidige DDHOR-methode (waterstanden/debieten via interpolatie gedefinieerd, zie paragraaf 2.2) te gebruiken in een methode die op uitgaande informatie is gebaseerd.

De koppelingscondities die uiteindelijk in het prototype van [1] zijn verwerkt zijn:

1. Gelijkheid van het debiet op de interface, ten behoeve van massabehoud:

$$q_{\mathbf{x}_{m1,n}} = \sum_{\tilde{\mathbf{n}}} q_{\tilde{\mathbf{x}}_{\tilde{m}\tilde{f},\tilde{\mathbf{n}}}} \quad (19)$$

Hierin verwijzen de tildes naar grootheden van het fijne domein.

2. Gelijkheid van de waterstand op de interface (snelheidspunten) tussen rijen van het grove domein en alle daarmee overeenkomende rijen van het fijne domein.

$$\tilde{s}_{\tilde{\mathbf{u}}_{\tilde{m}\tilde{f},\tilde{\mathbf{n}}}} = \mathbf{s}_{\mathbf{u}_{m1,n}}. \quad (20)$$

Met name met de laatste formule (20) is er later in change c66693 een probleem geconstateerd. Deze formule stelt namelijk dat de waterstand op de interface bij 1:3 verfijning in drie opeenvolgende rijen van het fijne domein hetzelfde is. Dit leidt tot ongewenste effecten voor golven die parallel aan de interface lopen.

In de huidige change is de tweede koppelingsconditie daarom aangepast door de koppelingsconditie voor de waterstanden als volgt uit te breiden:

- 2'. De waterstand in interfacepunten van het fijne domein is gelijk aan de waterstand in het overeenkomstige punt van het grove domein plus een variatie  $d\tilde{\mathbf{s}}\mathbf{u}$  langs de interface:

$$\tilde{s}_{\tilde{\mathbf{u}}_{\tilde{m}\tilde{f},\tilde{\mathbf{n}}}} = \mathbf{s}_{\mathbf{u}_{m1,n}} + d\tilde{\mathbf{s}}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{\mathbf{n}}} \quad (21)$$

Een moeilijkheid hierbij is om het profiel  $d\tilde{\mathbf{s}}\mathbf{u}$  langs de interface te kiezen. Op deze moeilijkheid wordt dieper ingegaan in Sectie 3.4.

In het geval van 1:1 koppeling komt conditie (21) overeen met (20) met de keuze  $d\tilde{\mathbf{s}}\mathbf{u} = 0$ . Wanneer beide domeinen dezelfde impulsvergelijking en tetau-interpolatie hanteren, dan voldoet alleen de volgende oplossing hieraan:

$$q_{\mathbf{x}_{m1,n}} = q_{\tilde{\mathbf{x}}_{\tilde{m}\tilde{f},\tilde{\mathbf{n}}}}, \quad \tilde{s}_{\tilde{\mathbf{u}}_{\tilde{m}\tilde{f},\tilde{\mathbf{n}}}} = \mathbf{s}_{m1,n}, \quad \tilde{s}_{\tilde{\mathbf{u}}_{\tilde{m}\tilde{f}+1,\tilde{\mathbf{n}}}} = \mathbf{s}_{m1+1,n} \quad (22)$$

In dat geval is de koppeling equivalent met die van de huidige DDHOR programmatuur. In het geval van horizontale verfijning leveren deze koppelingen niet exact dezelfde oplossingen als de huidige programmatuur, maar wel oplossingen die even nauwkeurig zouden moeten zijn.

### 3.3 Oplossen van de koppelingsvergelijkingen

In het prototype kon er vanuit de koppelingscondities (19) en (20) en de koppelingsinformatie (8) met `q2sep` berekend volgens (12) eenvoudig een praktisch algoritme worden geconstrueerd voor gevallen met verfijning.

- Alle domeinen berekenen hun eigen coëfficiënten voor de optimale koppeling `q2sep` en de bijbehorende uitgaande informatie `outgo`.



- Het grove domein krijgt de uitgaande informatie `outgo` en de coëfficiënt `q2sep` toegestuurd van het buurdomein. De contributies van de verschillende buurrijen van de grof-roosterrij worden bij elkaar opgeteld:

$$\text{outngh}_{m1,n} = \sum_{\tilde{n}} \text{outgo}_{m\tilde{f},\tilde{n}} \quad , \quad \text{q2sngh}_{m1,n} = \sum_{\tilde{n}} \text{q2ssep}_{m\tilde{f},\tilde{n}} \quad (23)$$

- Het grove domein heeft nu de beschikking over de volgende vergelijkingen:

$$\begin{aligned} \text{eigen vgl:} \quad \text{outgo}_{m1,n} &= \text{qx}_{m1,n} + \text{q2sep}_{m1,n} \text{su}_{m1,n}, \\ \text{ontvangen vgl:} \quad \text{outngh}_{m1,n} &= \sum_{\tilde{n}} \left( \text{q}\tilde{\text{x}}_{m\tilde{f},\tilde{n}} + \text{q2ssep}_{m\tilde{f},\tilde{n}} \text{s}\tilde{\text{u}}_{m\tilde{f},\tilde{n}} \right) \end{aligned} \quad (24)$$

- In deze vergelijkingen mogen de koppelingscondities (19) en (20) worden ingevuld. De vergelijkingen worden hiermee:

$$\begin{aligned} \text{eigen vgl:} \quad \text{outgo}_{m1,n} &= \text{qx}_{m1,n} + \text{q2sep}_{m1,n} \text{su}_{m1,n}, \\ \text{ontvangen vgl:} \quad \text{outngh}_{m1,n} &= \text{qx}_{m1,n} + \text{q2sngh}_{m1,n} \text{su}_{m1,n} \end{aligned} \quad (25)$$

- De twee vergelijkingen (25) kunnen eenvoudig worden opgelost en leveren voor het grove domein de waterstand `sum1,n` en het debiet `qxm1,n` op de interface op.
- De waterstand op de interface wordt naar het fijne domein gecommuniceerd. Hierbij wordt conform de koppelingsconditie (20) constante interpolatie gebruikt.
- Het fijne domein gebruikt de geveegde vergelijking (4) en de definitie (9) van de waterstand in het interfacepunt voor het oplossen van zijn waterstanden, en de oplossing is bekend.

In het geval van 1:1 koppeling wordt de uitgaande informatie naar beide domeinen gecommuniceerd, waarna beide domeinen het stelsel (25) oplossen, en verdere communicatie niet nodig is.

Voor de nieuwe koppelingsconditie (21) verloopt het algoritme grotendeels hetzelfde. De introductie van het profiel langs de interface `d̃su` werkt echter door in vergelijking (25), zodat het grove domein nu het volgende stelsel op te lossen krijgt:

$$\begin{aligned} \text{eigen vgl:} \quad \text{outgo}_{m1,n} &= \text{qx}_{m1,n} + \text{q2sep}_{m1,n} \text{su}_{m1,n}, \\ \text{ontvangen vgl:} \quad \text{outngh}_{m1,n} &= \text{qx}_{m1,n} + \text{q2sngh}_{m1,n} \text{su}_{m1,n} + \text{dcross}_{m1,n}, \end{aligned} \quad (26)$$

waarbij de term `dcross` gegeven wordt door

$$\text{dcross}_{m1,n} = \sum_{\tilde{n}} \text{q2ssep}_{m\tilde{f},\tilde{n}} \text{d}\tilde{\text{su}}_{m\tilde{f},\tilde{n}}. \quad (27)$$

Voor het opstellen van de vergelijking (26) moet naast de termen `q2sngh` en `outngh` ook de term `dcross` verkregen worden door middel van communicatie. Vervolgens kunnen de twee onbekenden `qxm1,n` en `sum1,n` op het grove rooster eenvoudig worden bepaald. De waterstand wordt weer naar het fijne domein gecommuniceerd, en daar kan vervolgens de totale oplossing worden bepaald.

### 3.4 Waterstandsprofiel langs de interface in het fijne domein

In paragraaf 3.2 is de term  $\tilde{d}\mathbf{s}\mathbf{u}$  geïntroduceerd, waarmee het profiel langs de interface voor het fijne domein wordt verwerkt. In deze paragraaf presenteren en bespreken we vier verschillende keuzen voor deze term:

- In het prototype [1] is geëxperimenteerd met de eenvoudigste keuze  $\tilde{d}\mathbf{s}\mathbf{u} = 0$ . Deze keuze bleek niet te voldoen, omdat deze keuze leidt tot een vlakke waterspiegel in de meeste roosterpunten, en een zeer groot verhang in de rest van de roosterpunten.
- Een beter profiel wordt verkregen door middel van (lineaire) interpolatie van de waterstanden op de interface:

$$\tilde{s}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{n}} = \tilde{z}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{n}} \implies \tilde{d}\mathbf{s}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{n}} = \tilde{z}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{n}} - \mathbf{s}\mathbf{u}_{m1,n}. \quad (28)$$

waarbij  $\tilde{z}\mathbf{u}$  wordt verkregen uit de interpolatie van de grofroosterwaterstanden  $\mathbf{s}\mathbf{u}$ . Het probleem van deze keuze is echter dat de geïnterpoleerde waarde  $\tilde{z}\mathbf{u}$  afhangt van de oplossing uit verschillende rijen en daarom niet vooraf bekend is. Een eenvoudig iteratief proces kan worden gebruikt om het stelsel op te lossen. Het is echter duidelijk dat de introductie van (nog) een extra iteratief proces het algoritme en de programmatuur (verder) compliceert.

- Een keuze voor  $\tilde{d}\mathbf{s}\mathbf{u}$  die vooraf kan worden berekend, voorkomt de introductie van een extra iteratief proces. Een eenvoudige manier om een expliciete formulering te verkrijgen, is om de keuze (28) eenmalig te berekenen, op basis van de invoer:

$$\tilde{d}\mathbf{s}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{n}} = \tilde{z}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{n}}^{old} - \mathbf{s}\mathbf{u}_{m1,n}^{old}. \quad (29)$$

- Een laatste keuze die we willen bespreken baseert zich op de waarden van de coëfficiënt  $\mathbf{e}$ . Deze coëfficiënt bevat een benadering voor de waterstand aan het einde van de komende tijdstap, waarbij alleen de debieten in de richting haaks op de rekenrichting zijn verwerkt. De coëfficiënt  $\mathbf{e}$  wordt gegeven door

$$\mathbf{e}_{m,n} = \mathbf{s}_{m,n}^{old} - \text{hdt} \frac{\mathbf{q}\mathbf{y}_{m,n} - \mathbf{q}\mathbf{y}_{m,n-1}}{\mathbf{g}\mathbf{s}\mathbf{q}\mathbf{s}_{m,n}}. \quad (30)$$

Het profiel  $\tilde{d}\mathbf{s}\mathbf{u}$  wordt, gebaseerd op  $\tilde{\mathbf{e}}$ , zodanig gekozen dat de grofroosterwaterstand gelijk is aan het gemiddelde van de fijnroosterwaterstanden in de overeenkomstige rijen. Er moet dus gelden

$$\sum_{\tilde{n}} \tilde{d}\mathbf{s}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{n}} = 0. \quad (31)$$

Hieraan kan worden voldaan door een coëfficiënt  $\mathbf{e}2$  te introduceren, die per grofroosterrij een enkele waarde heeft:

$$\tilde{d}\mathbf{s}\mathbf{u}_{\tilde{m}\tilde{f},\tilde{n}} = \tilde{\mathbf{e}}_{\tilde{m}\tilde{f}+1,\tilde{n}} - \mathbf{e}2_{m1,n}. \quad (32)$$

De gemiddelden van de fijnroosterwaterstanden zijn gelijk aan de grofroosterwaterstand wanneer de coëfficiënt  $e_2$  gelijk is aan de gemiddelde waarde van  $\tilde{e}$ :

$$e_{2_{m1,n}} = \frac{\sum_{\tilde{n}} \tilde{e}_{m\tilde{n}+1,\tilde{n}}}{\sum_{\tilde{n}} 1}. \quad (33)$$

### 3.5 Geneste iteratieprocessen

De benadering van de uitgaande informatie zou goed moeten werken wanneer verstoringen ruimtelijk gezien slechts een beperkt invloedsgebied hebben, dus wanneer de verschillende domeingrenzen relatief ver uit elkaar liggen en er een relatief kleine tijdstap wordt gebruikt. We nemen aan dat deze benadering goed werkt en dat ze de over-all convergentie niet beïnvloedt.

De benadering (28) voor de waterstandsconditie blijkt in veel gevallen zeer snel te convergeren, maar blijkt in bepaalde gevallen met horizontale verfijning wel de overall convergentie te kunnen vertragen. Om deze invloed te kunnen onderzoeken c.q. uitschakelen is er vooralsnog een extra binnen-iteratie geïntroduceerd. Hiermee kan de ideale performance van de gebruikte koppelingscondities en koppelingsalgoritme worden onderzocht.

### 3.6 Praktische uitwerking van lastige gevallen

De beschrijving van de methode is nagenoeg compleet. Er zijn nog wel enkele voorzieningen nodig geweest om kleine complicaties te verwerken:

- In geval van droogval wordt de uitgaande informatie gelijk aan het debiet plus nul maal de waterstand. Als beide buurdomeinen deze zelfde informatie doorsturen wordt het onmogelijk om hieruit de waterstand net buiten het eigen domein te berekenen. In geval van droogval wordt daarom in het geval zonder horizontale verfijning net iets andere informatie doorgestuurd.

In het geval van horizontale verfijning wordt een homogene Neumann-randvoorwaarde gehanteerd voor de waterstand.

- Door het gebruik van het `tetau`-schema voor de interpolatie van de waterstand kan er een stelsel ontstaan waar het diagonaalelement  $b$  net buiten het domein gelijk is aan nul. In dat geval moet het vegen van het stelsel een klein beetje anders worden aangepakt. Deze problematiek speelt alleen bij domeinen die aan beide kanten gekoppeld zijn.

## 4 Realisatie van de methode in de programmatuur

Bij de implementatie van de hierboven beschreven methode in de WAQUA/TRIWAQ programmatuur spelen de volgende extra aspecten een rol:

- Bij de implementatie van het BJ-TWGE algoritme wordt gebruik gemaakt van een schema om kleuren toe te kennen aan rijen. Aan de hand hiervan wordt de richting van de sweeps per iteratie bepaald zodanig dat de Jacobi-koppelingen steeds zo gunstig mogelijk worden gekozen. Het is de vraag of een dergelijk schema ook in de nieuwe methode nuttig en/of mogelijk is.
- Voor het vegen van tridiagonale stelsels wordt een generieke routine `trstrd` gebruikt. Dit is gewenst, en in ieder geval zou het vegen (lineaire algebra) zo veel mogelijk gescheiden moeten worden gehouden van het opstellen van de randvergelijkingen (fysica).
- Er zijn nieuwe communicaties nodig voor waarden op de interface.

Deze aspecten worden hieronder verder uitgewerkt.

### 4.1 Richting van de sweeps

Bij de implementatie van het BJ-TWGE algoritme wordt gebruik gemaakt van een schema om kleuren toe te kennen aan rijen. Aan de hand hiervan wordt de richting van de sweeps per iteratie bepaald zodanig dat de Jacobi-koppelingen steeds zo gunstig mogelijk worden gekozen. Het is de vraag of een dergelijk schema ook in de nieuwe methode nuttig en/of mogelijk is.

Het nut van een dergelijk schema kan eenvoudig worden geïllustreerd aan de hand van een domein dat aan twee kanten gekoppeld is. Als je aan een kant van het domein een Jacobi randvoorwaarde gebruikt dan kun je aan de andere kant de nieuwe methode toepassen. Maar je moet dit in de volgende iteratie wel verwisselen. Als je steeds op dezelfde rand een Jacobi-vergelijking gebruikt dan zakt de overall convergentie ineen.

Het is echter niet noodzakelijk om Jacobi-vergelijkingen te gebruiken in de uiteindelijke sweep van een iteratie. Je kunt ook aan beide kanten de uitgaande informatie gebruiken. Hiervoor moet je eerst twee kanten uit vegen (startend vanuit een Jacobi-vergelijking), aan beide kanten de uitgaande informatie bepalen, aan beide kanten de waterstand op de interface berekenen, de goede randvoorwaarden opstellen voor het domein, en tenslotte het stelsel met de goede randvoorwaarden oplossen middels een double sweep. In dit schema komen geen kleuren voor, maar wordt er wel dubbel zo veel rekenwerk gedaan (twee volledige double sweeps). Dit is met name voor parallel rekenen sterk ongewenst.

Dit schema met uitgaande informatie aan beide kanten kan met kleuren efficiënter worden gemaakt. Als de domeinen namelijk om en om beginnen, de een met een forward sweep, de ander met een backward sweep, dan kan er na de eerste sweep al een goede randvoorwaarde

worden bepaald. De tweede sweep die nodig is om uitgaande informatie te berekenen kan hierop worden gebaseerd, in plaats van dat een Jacobi-vergelijking wordt gebruikt. Hiermee wordt een van de vier sweeps uitgespaard. In plaats daarvan moet er wel twee keer uitgaande informatie worden gecommuniceerd; voor de ene groep van interfacepunten na de eerste sweep, voor de andere groep na de tweede sweep.

## 4.2 Overzicht van de implementatie in `trscnt`

Het hart van de solver voor de continuïteitsvergelijking in Waqua/Triwaq wordt gevormd door subroutine `trscnt`. Hierin is het iteratieproces geïmplementeerd waarin steeds de impulsvergelijking verticaal wordt geïntegreerd, in de continuïteitsvergelijking wordt gesubstitueerd, tri-diagonale stelsels worden opgelost, laagposities worden berekend, en droogvalcontroles worden uitgevoerd.

Ten behoeve van de nieuwe methode voor parallel rekenen en voor domein decompositie op basis van uitgaande informatie is de manier waarop de tri-diagonale stelsels per rij worden opgelost aangepast. Subroutine `trstrd` waarin het BJ-TWGE schema is geïmplementeerd wordt niet meer gebruikt in `trscnt` en is vervangen door drie nieuwe routines: `trsswp`, `trsout` en `trssub`.

Eerst worden alle tri-diagonale stelsels door `trsswp` voor de helft geveegd. Voor rijen die aan een kant gekoppeld zijn aan een ander subdomein gebeurt dit van de echte rand naar de interface toe. Voor rijen die niet gekoppeld zijn of die aan twee kanten gekoppeld zijn wordt de richting van de sweep op basis van een rij-kleur bepaald. Rode rijen worden voorwaarts geveegd, zwarte rijen achterwaarts. Hierin wordt niet langer per iteratie gevarieerd.

Voor rijen die aan twee kanten gekoppeld zijn wordt er tegelijkertijd met het oorspronkelijke stelsel een tweede rechterlid mee geveegd. Dit rechterlid bestaat uit een 1 in het randpunt van waaruit wordt gestart en verder uit nullen. Met dit rechterlid kan later de fout in de gebruikte Jacobi randvoorwaarde worden gecorrigeerd.

Een belangrijk produkt van `trsswp` is de geveegde vergelijking voor het laatste (voorwaartse sweep) of eerste (achterwaarts) interne punt. Dit zijn de vergelijkingen (4) en (5) paragraaf 2.3. Deze vergelijkingen worden naar aparte arrays `arow`, `brow`, `crow` en `drow` gekopieerd zodat ze niet door andere bewerkingen verloren kunnen gaan.

Na deze algebraïsche bewerkingen wordt er in subroutine `trsout` een stuk fysica gedaan. In deze routine wordt voor alle koppelranden de coëfficiënt van optimale koppeling `q2sep` en de uitgaande informatie `outgo` bepaald. Deze informatie wordt gecommuniceerd en hieruit worden de waterstanden op de interfaces opgelost.

In dit stadium wordt er bij sommige interfacepunten een benadering van de uitgaande informatie gebruikt. Met name geldt dit voor de zijde van dubbel-gekoppelde rijen waar een Jacobi-voorwaarde is gehanteerd. Hier wordt ofwel de uitgaande informatie van de vorige iteratie of een allereerste benadering gebruikt.

De volgende stap betreft terugsubstitutie van de half geveegde stelsels. Dit gebeurt in subroutine `trssub`. Daarbij wordt ter plekke de randvergelijking voor interfacepunten opgesteld en opgelost. Deze vergelijking is dat de waterstand op de interface gegeven is.

Voor rijen met aan weerszijden een koppeling naar een ander subdomein worden twee extra acties uitgevoerd:

1. Naast de gewone oplossing voor rechterlid `d` wordt ook de oplossing voor het rechterlid `xbj` bepaald. Hiermee kan later de correctie voor de Jacobi-koppeling worden gemaakt.
2. De geveegde vergelijking voor het begin van de rij (rode rijen, forward sweep in `trsswp`) of einde van de rij (zwarte rijen) wordt bepaald en apart gezet in `a-drow`.

Nu wordt `trsout` opnieuw aangeroepen, met name voor het bepalen van de uitgaande informatie aan de tweede zijde van dubbel gekoppelde rijen. In dit stadium is er voor alle interfacepunten uitgaande informatie beschikbaar. Na de uitwisseling hiervan is in alle interfacepunten de resulterende waterstand bekend.

Tenslotte wordt `trssub` voor de tweede keer in de iteratie opgestart. Hierin worden nu twee verschillende acties uitgevoerd:

- Voor enkel-gekoppelde rijen wordt de terugsubstitutie gedaan. Deze was overgeslagen in de eerste call van `trssub` omdat niet zeker was of van de buur-rij al goede uitgaande informatie ontvangen was.
- Voor dubbel gekoppelde rijen wordt een correctie voor de Jacobi-vergelijking toegepast. Deze is een factor maal de oplossing `xbj`. De factor die wordt gebruikt is de correcte waterstand op de interface die uit de uitgaande informatie is bepaald minus de waterstand die bij oplossing van het oorspronkelijke stelsel is gevonden.

In dit schema wordt vaker dan in het oorspronkelijke BJ-TWGE schema gecommuniceerd. Ook wordt er voor dubbel gekoppelde rijen meer rekenwerk gedaan. Daar staat tegenover dat er minder iteraties hoeven te worden uitgevoerd.

## 5 Resultaten met de nieuwe versie

In de `test` directory van deze melding zijn 15 testen klaargezet. Deze 15 testen kunnen alle achter elkaar worden aangezet met behulp van een scriptje. In elk van de testen wordt een bestand `uitvoer` gegenereerd, waarin eventueel falen van de berekening, de verschillen in de resultaten, de reketijden en de aantallen iteraties worden gerapporteerd.

Omdat de simulaties allemaal een beetje kort zijn, moet rekening worden gehouden met relatief veel ruis in de metingen van de reketijden. De aantallen iteraties kunnen echter zonder enige ruis worden gemeten. De betrouwbaarheid van de meting van reketijden neemt ook toe doordat er veel modellen zijn vergeleken: reketijden per model kunnen onnauwkeurig gemeten zijn, maar het algemene beeld zal naar alle waarschijnlijkheid betrouwbaar zijn.

Tabel 1: *Iteraties in trscnt in verschillende modellen*

model	x-richting			y-richting		
	nieuw	oud	verschil	nieuw	oud	verschil
test_riemann_1a	3.00	11.33	-73.52%	1.65	1.65	+00.00%
test_riemann_4	3.60	10.89	-66.94%	3.50	3.50	+00.00%
ddh_test_1a	7.80	18.30	-57.38%	2.60	2.60	+00.00%
maas_gr	2.46	4.46	-44.84%	2.43	5.08	-52.17%
schot_1:2	1.85	1.85	+00.00%	3.03	6.22	-51.29%
var_dp_1:2	1.91	1.92	-00.52%	2.95	5.74	-48.61%
test_along	1.96	2.65	-26.04%	2.01	3.40	-40.88%
test_lauw	2.55	2.88	-11.46%	2.50	3.71	-32.61%
kort_rij_1:2	2.74	3.52	-22.16%	2.00	2.00	+00.00%
kort_rij_1:4	2.43	3.10	-21.61%	1.98	1.97	+00.51%
ddh_test_3a	1.39	1.39	+00.00%	4.49	5.44	-17.46%
ddh_test_10a	4.76	5.31	-10.36%	3.22	3.45	-06.67%
csm8_trw_par	3.05	3.12	-02.24%	3.00	3.02	-00.66%
bak	2.61	2.61	+00.00%	2.05	2.05	+00.00%
ddh_1op1	3.00	3.00	+00.00%	2.00	2.00	+00.00%

De rekentijd is in alle gevallen vergelijkbaar of behoorlijk minder dan in de originele versie. De 99%-verschillen zijn allemaal minder dan 1.9cm, behalve in het model `ddh_test_10a`. De verschillen in de modellen `ddh_test_10a` en `maas_gr` zijn gevisualiseerd en lijken acceptabel. De rekentijden en iteraties worden getoond in de tabellen: percentages zijn ten opzichte van de oorspronkelijke resultaten.

Na het doorlopen van al deze testen is met de testbank nagegaan wat de invloed van de nieuwe methode is op de resultaten. Hierbij zijn alleen grote verschillen geconstateerd in het model `ddh_test_10` en in de modellen waarin proceskoppeling wordt gebruikt. Deze verschillen zijn gedetailleerd bestudeerd. Het blijkt dat er in deze modellen zeer sterke variaties langs de interfaces voorkomen. Enkele varianten van de gebruikte methode zijn geprobeerd, om te onderzoeken bij welke methode deze verschillen het kleinste waren.

Tot slot is er een test uitgevoerd met het nieuwe `csm-zuno` model. In dit model ontstond er in eerste instantie na verloop van tijd bij Denemarken een draaikolk, die met de standaardversie van de programmatuur niet wordt verkregen. Deze draaikolk wordt getoond in Figuur 1. Deze draaikolk is zodanig sterk, dat de nieuwe methode niet kan worden gebruikt zolang hij optreedt.

Om dit probleem te onderzoeken is met het programma `knippr` een veel kleiner model gemaakt. Hiermee konden de testen veel sneller worden herhaald en kon de oorzaak worden getraceerd. Deze blijkt te zitten in het feit dat de twee roosters niet goed aan elkaar passen. Niet alle dieptepunten op de interface van het grove domein vallen samen met dieptepunten van

Tabel 2: Totale rekentijden in verschillende modellen

model	CPU-tijd			Wall clock tijd		
	nieuw	oud	verschil	nieuw	oud	verschil
ddh_test_1a	2.18	3.33	-34.53%	8.73	11.26	-22.47%
ddh_test_3a	19.04	26.96	-29.38%	47.01	69.98	-32.82%
maas_gr	984.32	1336.94	-26.38%	2168.38	2927.64	-25.93%
ddh_test_10a	10.99	14.46	-24.00%	38.95	48.85	-20.27%
schot_12	0.93	1.16	-19.83%	4.04	4.45	-09.21%
test_along	3.24	3.98	-18.59%	13.14	12.66	+03.79%
var_dp_12	0.92	1.11	-17.11%	4.09	4.35	-05.98%
test_riemann_1a	0.15	0.18	-16.67%	0.62	0.66	-06.06%
kort_rij_12	0.67	0.74	-09.46%	2.81	2.87	-02.09%
test_lauw	4.09	4.45	-08.09%	16.32	13.93	+17.16%
kort_rij_14	0.70	0.74	-05.41%	2.90	2.98	-02.68%
ddh_1op1	0.82	0.85	-03.53%	3.19	3.26	-02.15%
bak	1.71	1.71	+00.00%	8.88	8.70	+02.07%
csm8_trw_par	75.97	74.85	+01.50%	183.91	179.53	+02.44%
test_riemann_4	0.19	0.18	+05.56%	1.66	1.73	-04.07%

het fijne domein, en er worden dus geen integer verfijningsfactoren langs de interface gebruikt. Dit is bereikt door een grote match-accuracy in te stellen in de DDHOR-configuratiefile. De oude DDHOR-methode kan hier blijkbaar tegen, al kunnen er wel onnauwkeurigheden en kleine artefacten worden verwacht. De nieuwe methode blijkt veel gevoeliger voor de correctheid van het rooster te zijn.

De grote draaikolk blijkt te kunnen worden verholpen met kleine aanpassingen aan het rooster, zoals wordt getoond in het linker plaatje van Figuur 2. Nadat hiermee de oorzaak was geïsoleerd, is een eenvoudiger remedie bedacht en getest: door middel van een update van de roosterlengten `guu` en `gvv` in de initialisatiefase worden de roosters beter passend gemaakt, waarna ook met een niet-passend rooster correcte resultaten worden verkregen. Hierbij zou de gebruiker gewaarschuwd moeten worden dat zijn rooster wordt aangepast en dat dit mogelijk onnauwkeurigheden introduceert.

## 6 Integratie in de moederversie

Tot slot is het prototype gebruikt om een nieuwe moederversie te maken. Deze is ingecheckt als revisie 2033. Voordat de nieuwe versie werd ingecheckt, is er uitvoerig getest. De testresultaten worden gepresenteerd in de Tabellen 3 en 4.

We verwachten hoogstens zeer kleine verschillen in parallelle en ddvert-modellen, omdat de nieuwe methode nauwelijks verschilt van de oude. Alleen de rij-kleuring en de codering van de



berekeningen zijn anders. In Tabel 3 zijn de grootste verschillen te vinden. In enkele gevallen (modellen `moha_wq_par1` en `t15.0kalman`) komen de verschillen overeen met de verschillen tussen parallelle en sequentiële berekeningen in de moederversie: deze verschillen zijn daarom aanvaardbaar. In de andere modellen, behalve `Lek`, komen de verschillen overeen met de gevoeligheid, die is vastgesteld met de methode `quantify_randomness`. Ook deze verschillen zijn aanvaardbaar. In het model `Lek`, ten slotte, zijn de gevonden verschillen tussen de resultaten van de nieuwe en de moederversie kleiner dan de verschillen tussen de resultaten van de moederversie met dubbele en enkele precisie.

Iets grotere verschillen worden verwacht bij modellen met horizontale verfijning. Deze verschillen worden getoond in Tabel 4. Een aantal verschillen wordt getoond in de figuren in Appendix A. In deze figuren is steeds te zien dat de verschillen plaatsvinden op interfaces waar de resolutie onvoldoende is om de oplossing zeer nauwkeurig weer te geven. Omdat de nieuwe koppeling andere, maar even goede, vergelijkingen oplost, zijn deze verschillen aanvaardbaar.

## 7 Conclusie

De koppeling van DDHOR-modellen met behulp van uitgaande informatie is in de code ingebracht en in de moederversie opgenomen. De nieuwe methode is uitgebreid getest, en in haast alle testen worden resultaten gevonden die sterk overeenkomen met de resultaten die worden verkregen met de moederversie, terwijl er een flinke hoeveelheid rekentijd wordt bespaard.

Een test met het nieuwe `csm-zuno` model verliep in eerste instantie niet correct: een sterke draaikolk ontwikkelde zich ten westen van Denemarken, op de interface tussen beide domeinen. Deze draaikolk is verholpen door een kleine aanpassing in de initialisatiefase van `waqpro`.

De nieuwe methode is daarmee een volwaardig alternatief geworden voor de bestaande methode, en daarom is de voorgaande methode vervangen. De beschrijving van de methode die in dit memo wordt gegeven is aan de documentatie van parallel rekenen en domein decompositie toegevoegd.

## Referenties

- [1] B. van 't Hof, E.A.H. Vollebregt, and M.J.A. Borsboom. Vooronderzoek domein decompositie met tijdstapkoppeling in WAQUA/TRIWAQ. Technical Report TR05-05, VORtech, Postbus 260, 2600 AG Delft, August 2005.

## A Resultaten met geïntegreerde versie

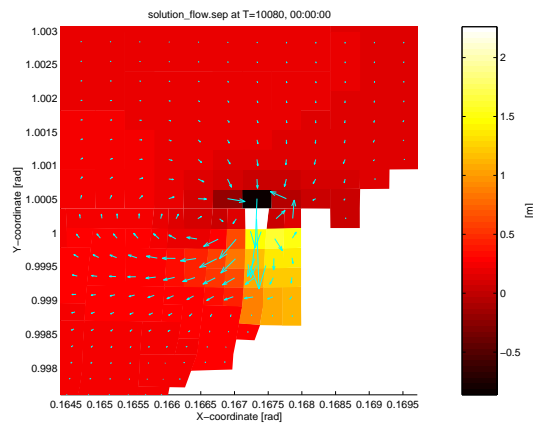
model	variable	single		double	
		trunk-new	seq-parll	trunk-new	seq-parll
moha_wq_par1	sep	0.11632	0.12053	0.12477	0.12477
	up+vp	0.72808	0.75457	0.89310	0.89315
	rp	0.00017	0.00027	0.00001	0.00018
t15.0kalman	sep	0.14378	0.13075	0	0.11626
	up+vp	0.33077	0.25910	0	0.27617
	rp	0.14866	0.14305	0	0.13951

model	variable	single		double	
		trunk-new	random	trunk-new	random
maasdemo	sep	0.04704	0.03840	0	
	up+vp	0.10892	0.09983	0	
maas1	sep	0.02161	0.01978	0	
	up+vp	0.05770	0.06777	0	
waal	sep	0.02015	0.02253	0	
	up+vp	0.04897	0.05469	0	
Lek	sep	0.01010	0.00008	0	
	up+vp	0.06308	0.00022	0	
ijssel	sep	0.00626	0.00653	0	
	up+vp	0.01955	0.01892	0	

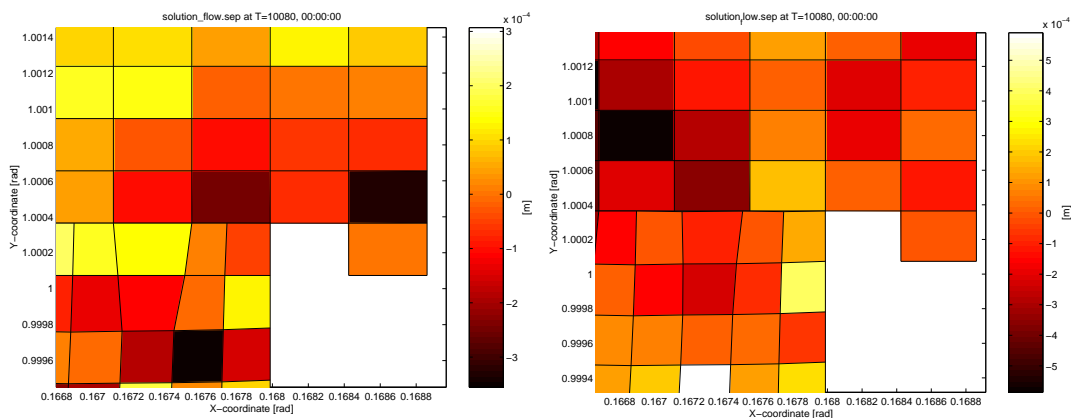
Tabel 3: 99-procentsverschillen tussen de resultaten van de nieuwe versie en de moederversie voor parallelle modellen. De verschillen komen door afrondverschillen of door zeer kleine verschillen in de koppeling. In de modellen `moha_wq_par1` en `t15.0kalman` zijn de verschillen (nagenoeg) gelijk aan de verschillen tussen sequentiële en parallelle berekeningen in de moederversie. In de andere modellen gaat het om afrondverschillen (mogelijk geïnduceerd door andere rijkleuring): in dubbele precisie verdwijnen de verschillen.

model	precision	sep	up+vp	rp
ddh_test_3b	single	0.020175	0.25529	3.591576
	double	0.021080	0.31533 <sup>(Fig 3)</sup>	25.85826 <sup>(Fig 3)</sup>
ddh_test_3c	single	0.024255	0.27261	4.140125
	double	0.024254	0.27018	3.444717
ddh_test_3a	single	0.005664	0.060780	0.913792
	double	0.005664	0.060794	1.210447
proc_kopp_waqua	single	0.034077	0.68116 <sup>(Fig 6)</sup>	6.128359 <sup>(Fig 6)</sup>
	double	0.034169	0.48908	3.803159
proc_kopp_triwaq	single	0.024971	0.034516	0.016803
	double	0.024749	0.034037	0.016302
ddh_test_12	single	0.077505 <sup>(Fig 5)</sup>	0.29217 <sup>(Fig 5)</sup>	
	double	0.077507	0.29217	
ddh_test_10b	single	0.038025 <sup>(Fig 4)</sup>	0.11039 <sup>(Fig 4)</sup>	0.034299
	double	0.038022	0.11036	0.085032
ddh_test_11b	single	0.037201	0.10956	0.030745
		0.037272	0.10983	0.100533
ddh_test_4	single	0.037611	0.16299	
	double	0.037610	0.16299	
ddh_test_11a	single	0.029166	0.097460	0.026594
	double	0.029177	0.097755	0.038847
ddh_test_9	single	0.023785	0.052943	
	double	0.023785	0.052940	
nikuradse_ddh	single	0.020573	0.13975	
	double	0.020576	0.13975	
ddh_gauss_kruger	single	0.019408	0.028500	
	double	0.019397	0.028482	

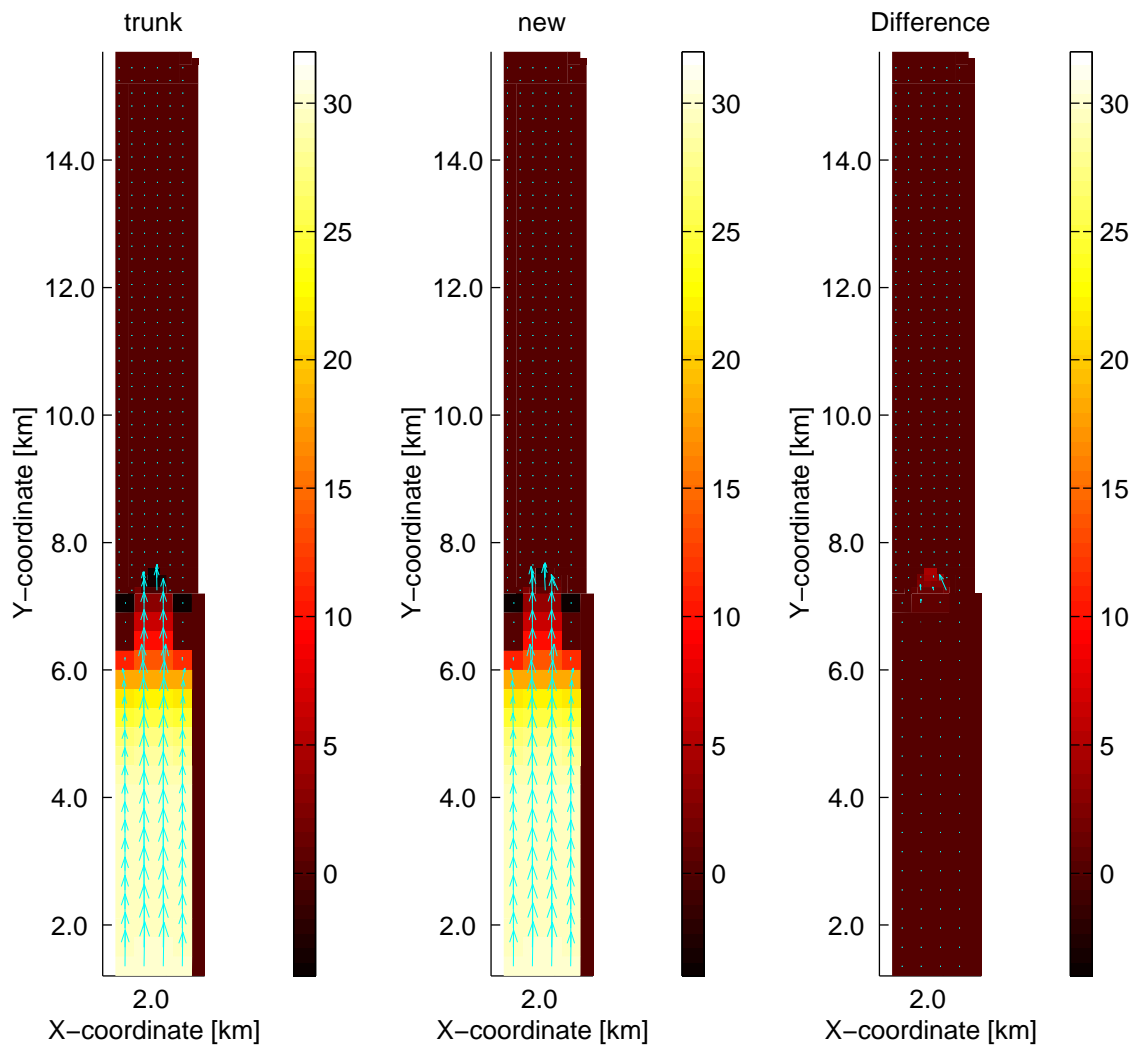
Tabel 4: 99-procentsverschillen tussen de resultaten van de nieuwe versie en de moederversie. Verschillen die zijn gemarkeerd met een verwijzing worden geïllustreerd in Appendix A.



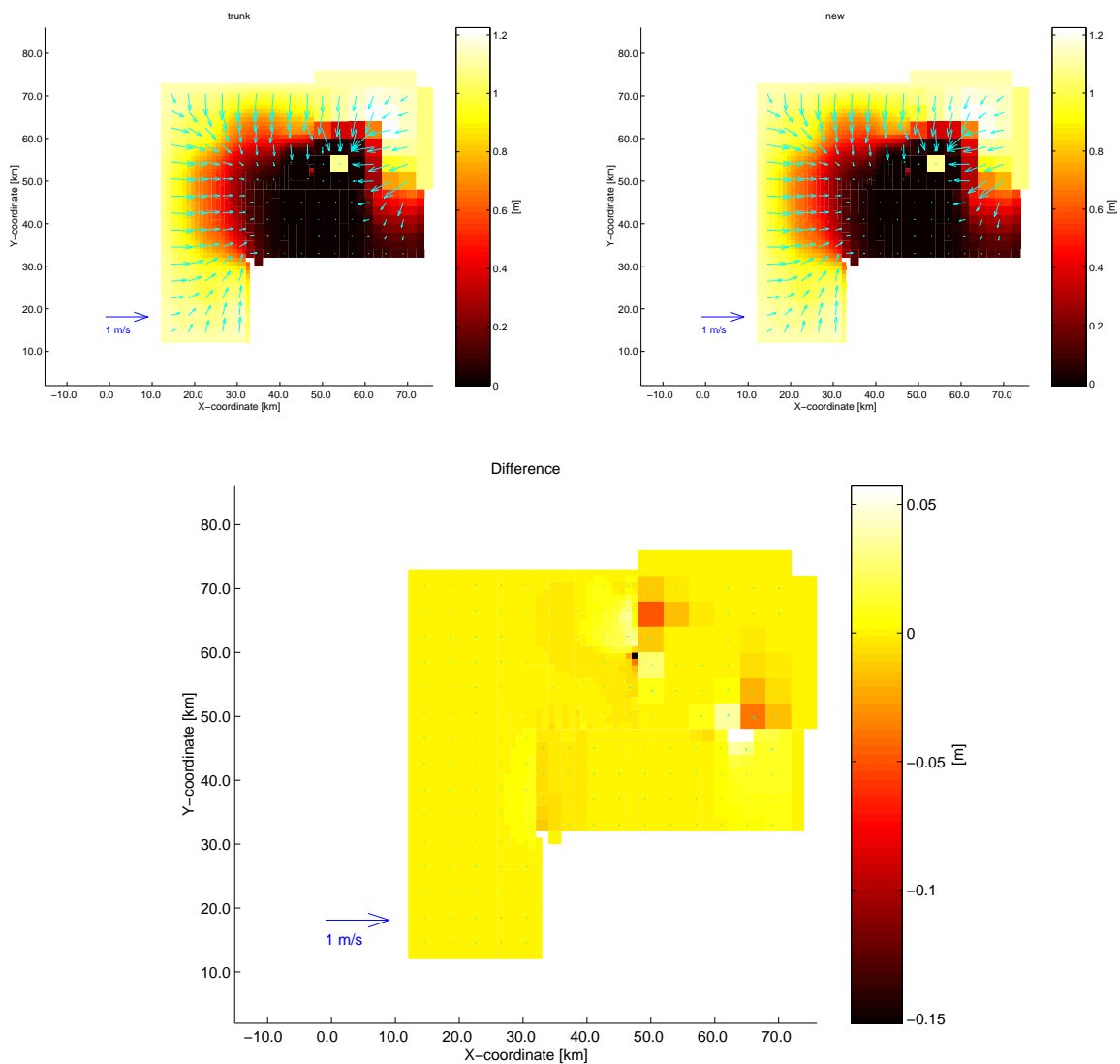
Figuur 1: Een draaikolk ontstaat in de buurt van Denemarken. In de standaard-versie van de programmatuur treedt deze draaikolk niet op.



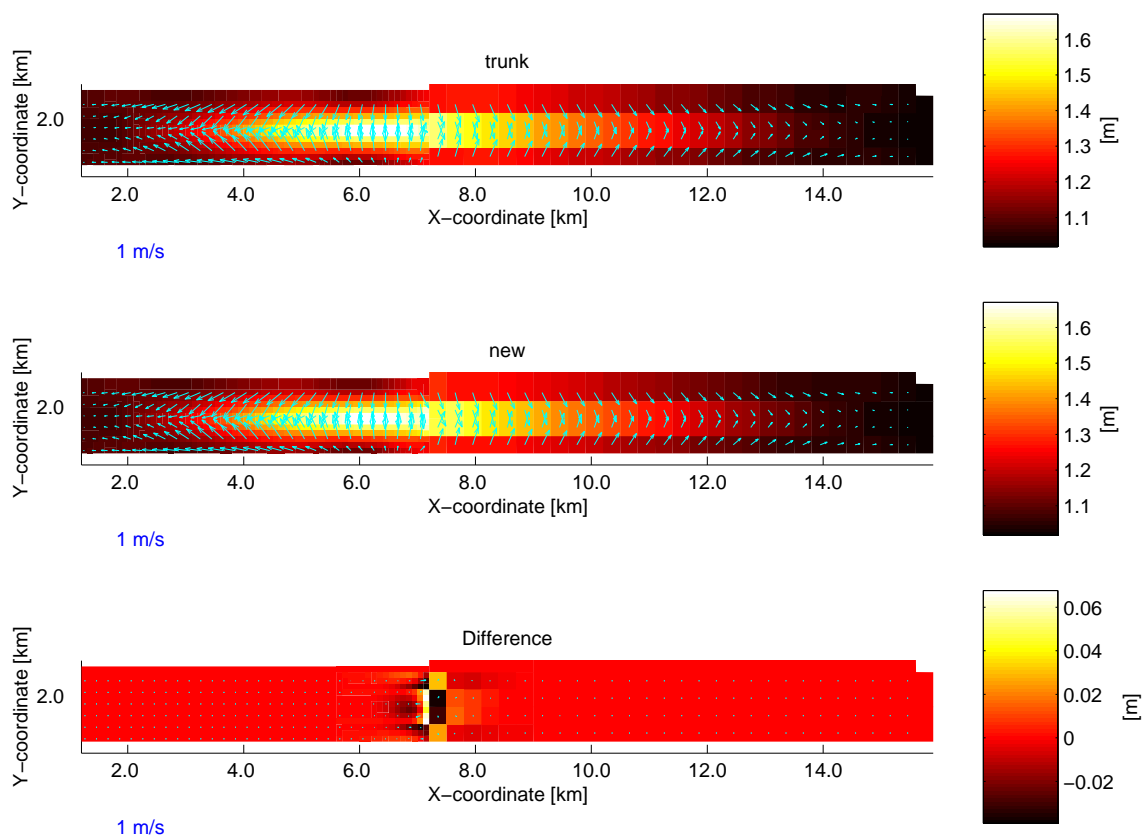
Figuur 2: Verschillen tussen de waterstanden berekend met de oorspronkelijke en de nieuwe koppelingsmethode voor het kleinere knippr-model. Links: de verschillen zijn zeer klein wanneer de roosters goed aan elkaar passend worden gemaakt. Rechts: de verschillen zijn ook zeer klein bij gebruik van het oorspronkelijke rooster, wanneer een extra update van de arrays guu en gvv wordt uitgevoerd in waqpro.



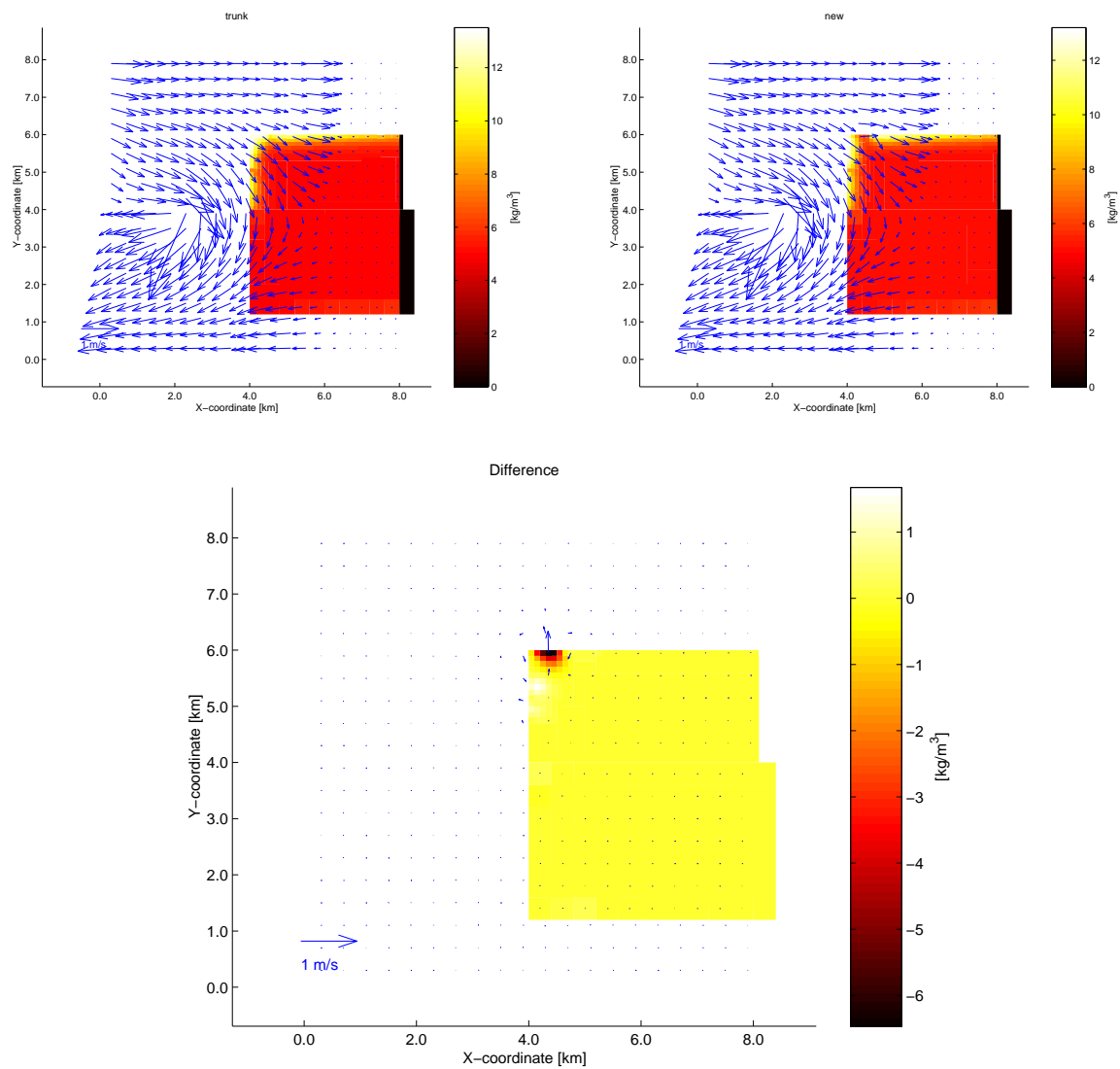
Figuur 3: Concentraties en stroomsnelheden voor het model `ddh_test_3b`. Links: resultaten met uitgangsversie. Midden: resultaten met de nieuwe versie. Rechts: verschil. De verschillen zijn relatief groot, maar de oplossingen zijn kwalitatief vergelijkbaar en de verschillen treden op bij een interface waar de resolutie onvoldoende is.



Figuur 4: Waterstanden en stroomsnelheden voor het model `ddh_test_10b`. Boven: resultaten met uitgangsversie links en nieuwe versie rechts. Onder: verschil. De verschillen zijn relatief groot, maar de oplossingen zijn kwalitatief vergelijkbaar en de verschillen treden op bij een interface waar de resolutie onvoldoende is.

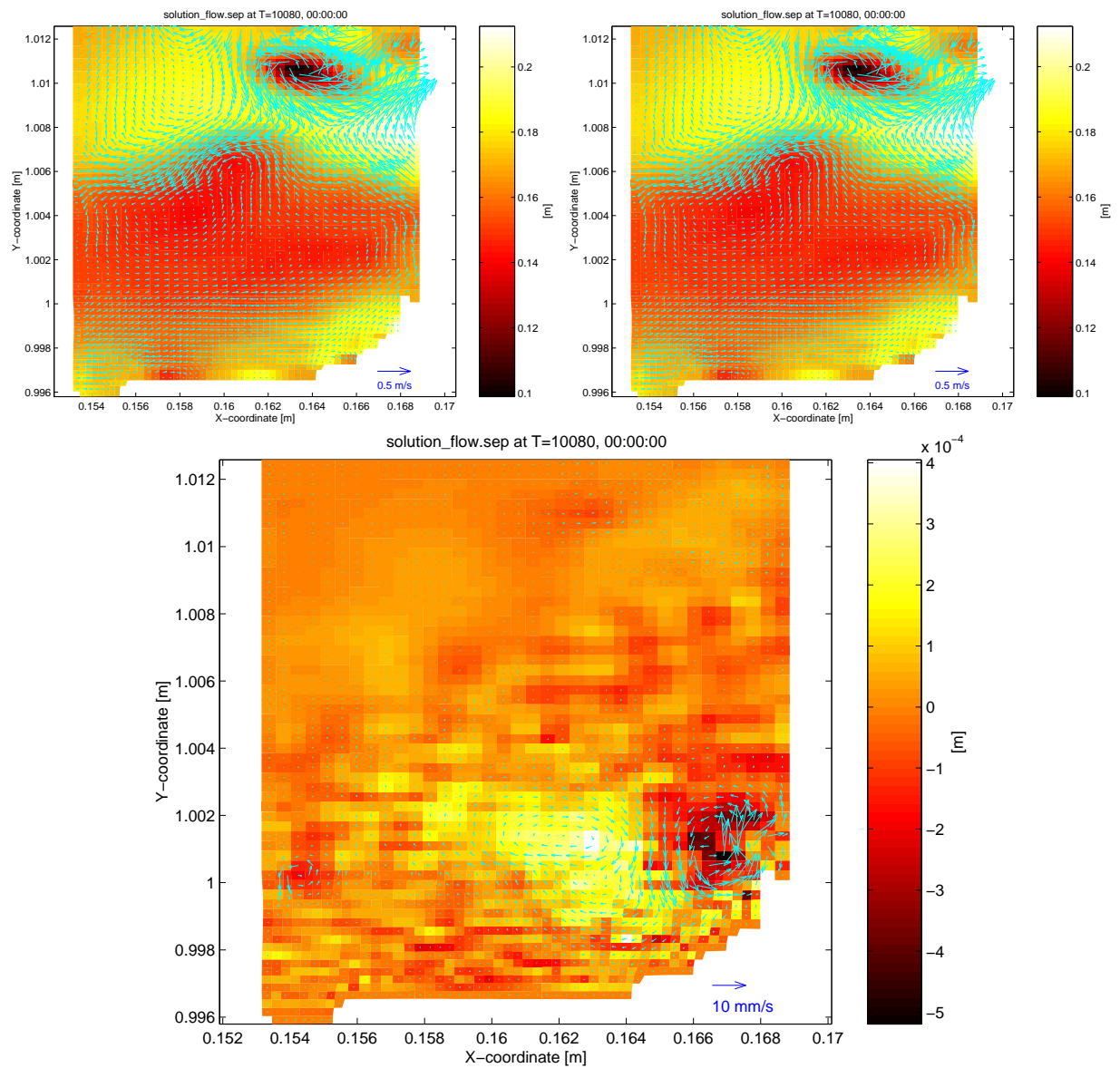


Figuur 5: Waterstanden en stroomsnelheden voor het model `ddh_test_12`. Boven: resultaten met uitgangsversie. Midden: resultaten met de nieuwe versie. Onder: verschil. De verschillen zijn relatief groot, maar de oplossingen zijn kwalitatief vergelijkbaar en de verschillen treden op bij een interface waar de resolutie onvoldoende is.



Figuur 6: Concentraties en stroomsnelheden voor het model `proc_kopp_waqua`. Boven: resultaten met uitgangsversie (links) en met de nieuwe versie (rechts). Onder: verschil. De verschillen zijn relatief groot, maar de oplossingen zijn kwalitatief vergelijkbaar en de verschillen treden op bij een interface waar de resolutie onvoldoende is.





Figuur 7: Waterstanden en stroomsnelheden voor de uitsnede van het nieuwe **czuno**-model. Boven: resultaten met uitgangsversie (links) en met de nieuwe versie (rechts). Onder: verschil. De verschillen zijn klein.