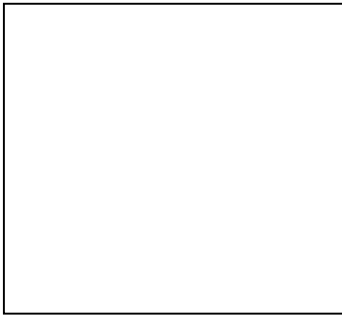
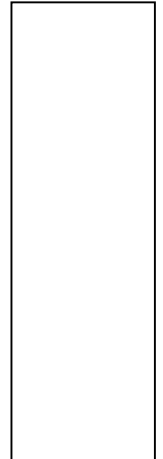




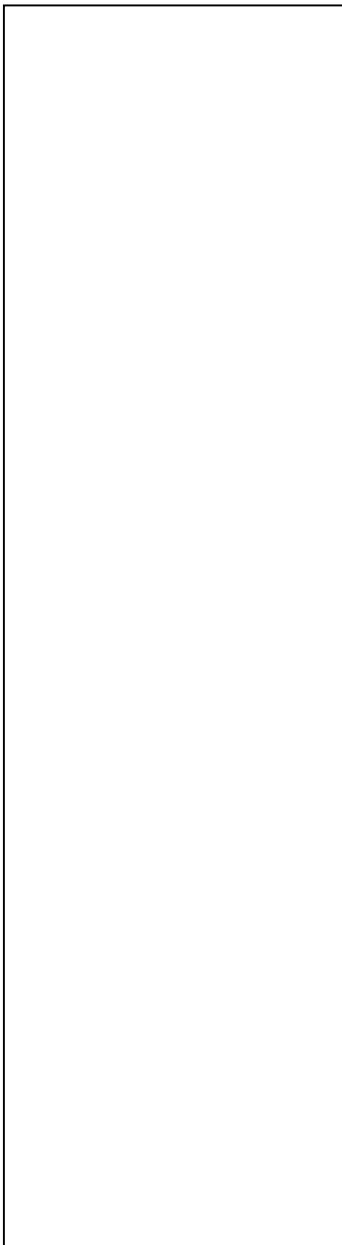
**HANDS-ON TRAINING DOMEIN  
DECOMPOSITIE**  
**Cursusboek Module 2**



# **HANDS-ON TRAINING DOMEIN DECOMPOSITIE**



**Cursusboek Module 2**



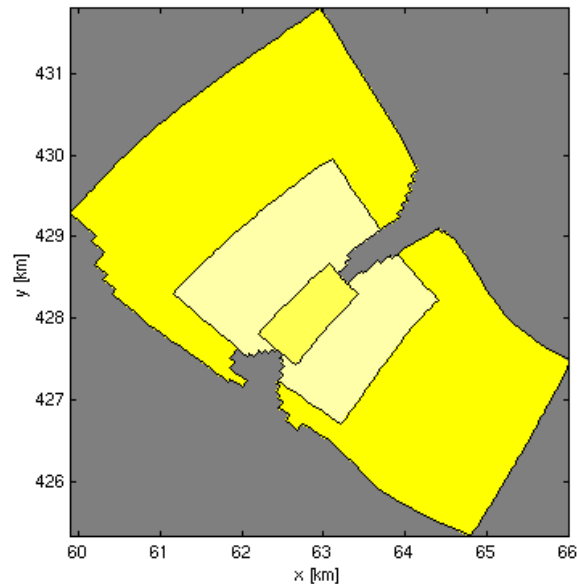
Version : 1.0, oktober 2002  
Maintenance : see [www.helpdeskwater.nl/waqa](http://www.helpdeskwater.nl/waqa)  
Copyright : Rijkswaterstaat

## Inhoudsopgave

Inhoudsopgave .....	2
1 Domein decompositie met verticale verfijning.....	4
1.1 Globaal overzicht van het DDVERT systeem .....	5
1.2 Een rekengebied opdelen met VISIPART .....	7
1.3 Speciale constructies in de <i>siminp</i> -file .....	8
1.4 Aansturing van de runprocedure waqpre.run.....	9
1.5 Toegestane laagverdelingen.....	9
1.6 Runprocedure waqpro.run.....	11
1.7 Resultaten van een berekening met verticale verfijning .....	12
2 Domein decompositie met horizontale verfijning.....	15
2.1 Globaal overzicht van het DDHOR systeem .....	16
2.2 Aan elkaar passende domeinen .....	17
2.3 Een configuratiebestand maken .....	20
2.4 Een simulatie opstarten.....	20
2.5 De uitvoer van de simulatie .....	21
Bibliografie.....	22



# 1 Domein decompositie met verticale verfijning



**Figuur 1: Mogelijke opdeling van het model van de Monding van het Haringvliet (MOHA). Er zijn 4 lagen gebruikt nabij de sluis, 2 lagen in een wijdere omtrek, en 1 laag op grotere afstand van de sluis.**

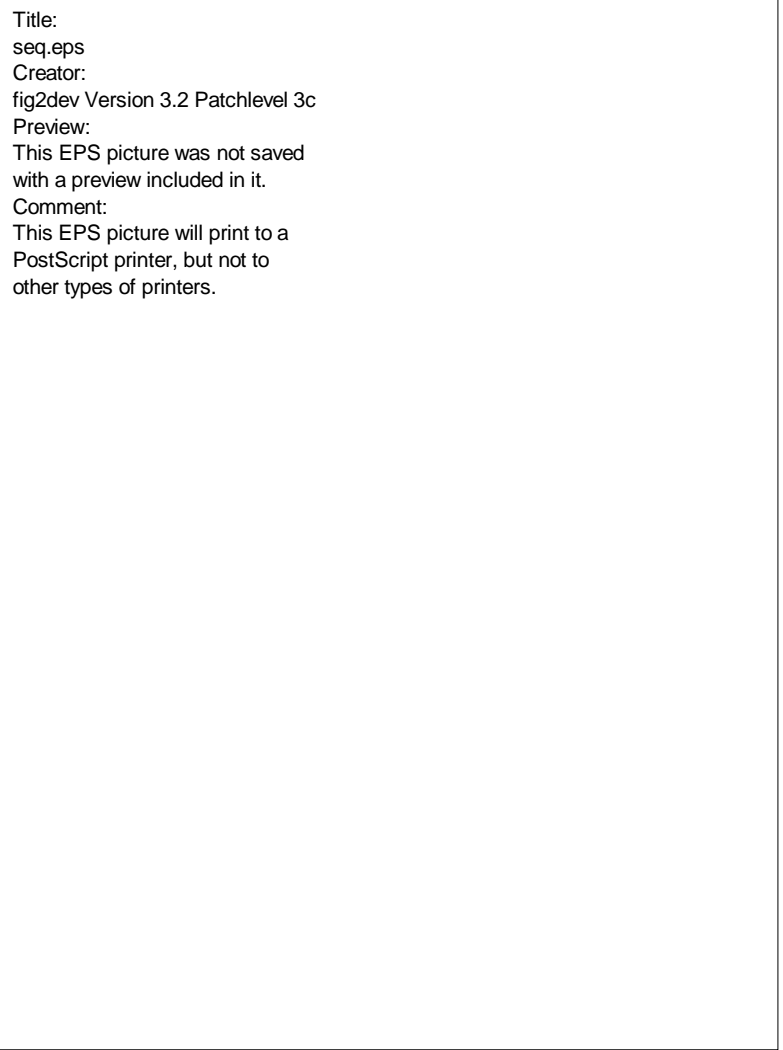
Verticale verfijning kan worden gebruikt voor berekeningen van situaties waar in een klein gedeelte van het rekengebied sterke gelaagdheid optreedt in de stroomsnelheden of saliniteiten.

Een voorbeeld van een dergelijke situatie is het model van de Monding van het Haringvliet (MOHA): nabij de Haringvlietssluisen is sprake van veel sterkere gelaagdheid dan elders. Een mogelijke opdeling van het MOHA-model in deelgebieden wordt getoond in Figuur 1. Hierbij is een groter aantal lagen in het rooster gekozen op de plaatsen waar dat nodig geacht wordt.

Dit document beschrijft hoe simulaties kunnen worden uitgevoerd op roosters zoals in Figuur 1 wordt getoond. Hierbij worden verwijzingen gegeven naar andere documenten voor aanvullende informatie. Verder worden er in een apart document opdrachten gegeven die kunnen worden gebruikt voor het leren werken met de hier beschreven technieken.

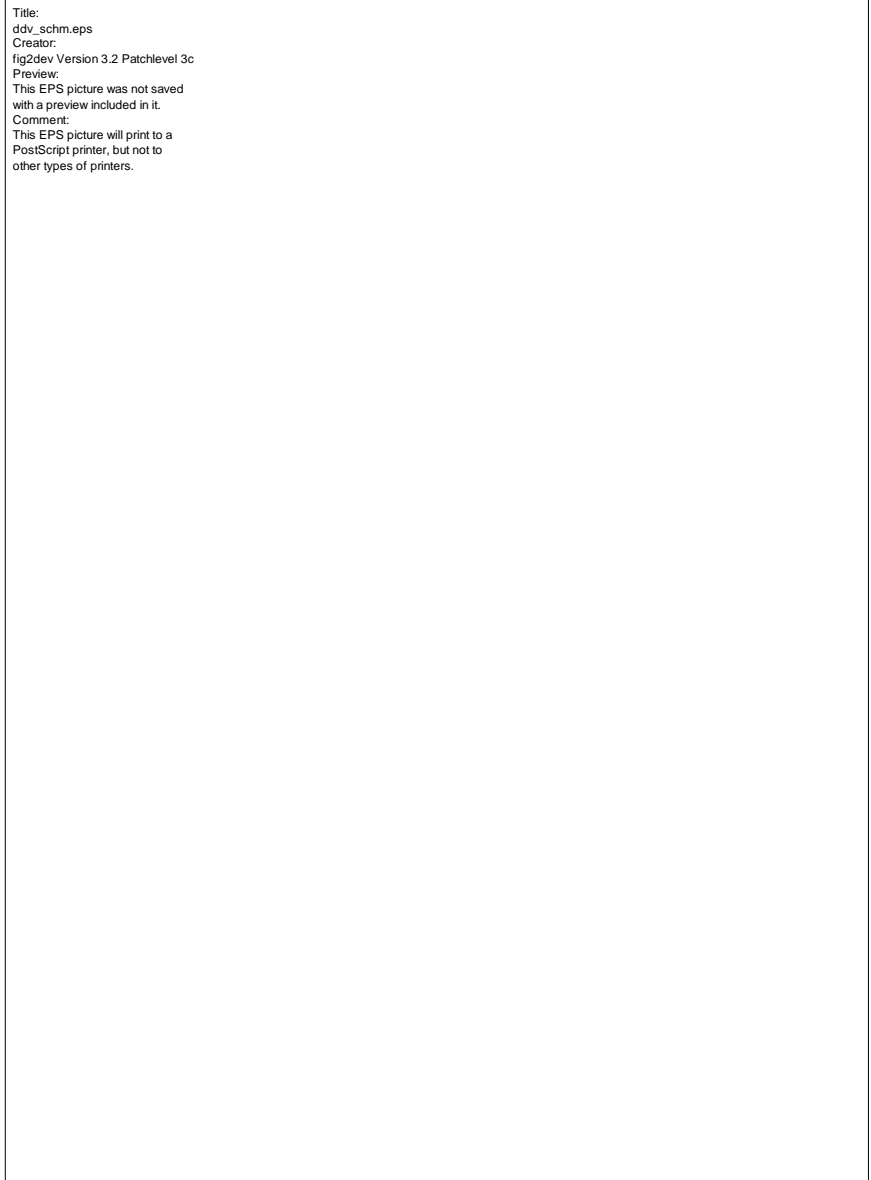
## 1.1

### Globaal overzicht van het DDVERT systeem



Title:  
seq.eps  
Creator:  
fig2dev Version 3.2 Patchlevel 3c  
Preview:  
This EPS picture was not saved  
with a preview included in it.  
Comment:  
This EPS picture will print to a  
PostScript printer, but not to  
other types of printers.

**Figuur 2: Globaal overzicht van een sequentiële berekening**



Title:  
ddv\_schm.eps  
Creator:  
fig2dev Version 3.2 Patchlevel 3c  
Preview:  
This EPS picture was not saved  
with a preview included in it.  
Comment:  
This EPS picture will print to a  
PostScript printer, but not to  
other types of printers.

### Figuur 3: Globaal overzicht van een DDVERT-simulatie

Een berekening met verticale verfijning (Figuur 3) verloopt in grote lijnen net zo als een sequentiële berekening (Figuur 2). Net zoals bij een sequentiële berekening wordt in de zogenaamde *siminp*-file een beschrijving gegeven van de gebiedsschematisatie en de forcings. De *siminp*-file dient als invoer voor het WAQUA-systeem, dat als uitvoer een SDS-file produceert waarin de simulatieresultaten staan.

Voor de berekening met verticale verfijning is enige extra invoer nodig. Ook zijn de bewerkingen binnen het systeem uitgebreid ten opzichte van de sequentiële berekening. Deze uitgebreide structuur van het WAQUA-systeem wordt grotendeels aan het oog van de gebruiker onttrokken. Daarom is deze voor een beginnende gebruiker van minder belang dan de uitbreidingen aan de invoer. Met de extra invoer krijgt de gebruiker echter direct te maken. In deze Sectie wordt daarom

beschreven waar de extra invoer toe dient en hoe zij door de gebruiker gemaakt kan worden.

De invoer voor de berekening staat bovenaan Figuur 2 en Figuur 3. In Figuur 2 staat hier alleen de *siminp*-file, terwijl hier in Figuur 3 naast de *siminp*-file ook een *areas*-file en enkele *include*-files staan.

In de *areas*-file staat de opdeling van het rekengebied in deelgebieden. Eenvoudige opdelingen van een rekendomein kunnen worden gemaakt door het WAQUA/TRIWAQ-systeem zelf (door het programma COPPRE), maar voor berekeningen met verticale verfijning ligt het niet voor de hand om deze mogelijkheid te gebruiken. De *areas*-file kan in dat geval worden gemaakt met behulp van het programma VISIPART, maar kan ook met de hand gemaakt worden. In Sectie 1.2 wordt beschreven hoe het rekengebied met VISIPART kan worden opgedeeld in deelgebieden. Hoe een *areas*-file handmatig gemaakt wordt, staat in de gebruikershandleiding voor parallel WAQUA/TRIWAQ en domein decompositie [ 1], pagina 13.

De *include*-files behoren eigenlijk tot de uitbreidingen van de *siminp*-file voor verticale verfijning. Deze uitbreidingen worden besproken in Sectie 1.3.

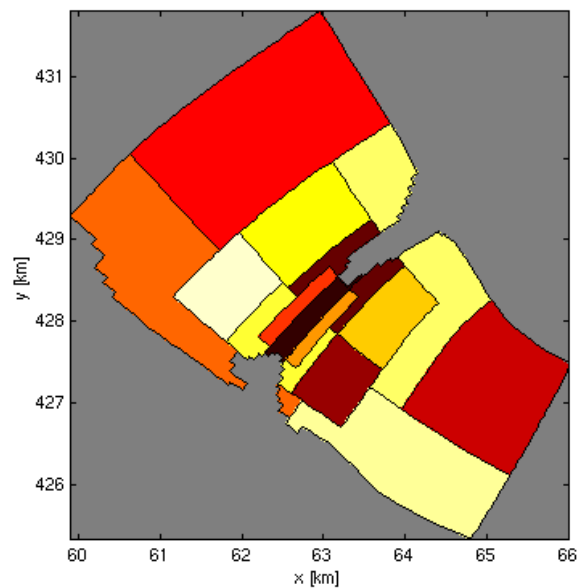
De gebruiker heeft relatief weinig te maken met de uitgebreide structuur van het inwendige van WAQUA-systeem. Toch kan, door middel van de invoerparameters die aan de runprocedure worden meegegeven, de werking van de afzonderlijke delen van het systeem apart worden aangestuurd. Het gebruik van de runprocedure wordt daarom besproken in Sectie 1.4.

## 1.2 Een rekengebied opdelen met VISIPART

Voor een berekening met verticale verfijning moet het rekendomein in stukken opgedeeld worden. Hiervoor kan het programma VISIPART gebruikt worden. Nadat de geometrie van het rekendomein is ingeladen, worden de gebieden met twee en vier lagen gemaakt door ze te selecteren met de muis. Vervolgens wordt de opdeling opgeslagen in de *areas*-file.

VISIPART biedt ook faciliteiten voor het verder opdelen van het rekengebied om de beschikbare processoren zo efficiënt mogelijk te gebruiken, zoals te zien is in Figuur 4. Het gebruik van VISIPART wordt uitvoerig uitgelegd in [ 2].





**Figuur 4:** een opdeling in 14 deeldomeinen die met VISIPART heel eenvoudig gemaakt kan worden.

### 1.3

#### Speciale constructies in de *siminp*-file

De gebiedsschematisatie en de forcings voor de berekening staan beschreven in de *siminp*-file. Deze *siminp*-file is bij verticale verfijning niet wezenlijk anders dan bij een sequentiële berekening. Bij gebruik van verticale verfijning is namelijk niet alle informatie die in de *siminp*-file staat gelijk voor elk van de delen van het rekengebied.

Het is duidelijk dat de waarde van de parameter KMAX (het aantal lagen in het rekenrooster) per deel verschilt. Ook de inhoud van het hoofdstukje VERTICAL (waarin de dikten van de verschillende lagen opgegeven wordt) kan per deeldomein verschillen. In sommige gevallen wordt er echter nog veel meer gevarieerd.

Denk hierbij bijvoorbeeld aan de diffusieparameter voor de getransporteerde stoffen. Deze wordt in schematisaties met weinig lagen vaak hoger gekozen dan in schematisaties met veel lagen, omdat de gelaagdheid van de stroomsnelheid, die alleen kan worden gereproduceerd in schematisaties met veel lagen, tot gevolg heeft dat de opgeloste stoffen sneller worden getransporteerd.

Deze verschillen in de invoer voor de verschillende deeldomeinen kan worden weergegeven in de *siminp*-file middels de codes %KMAX% en %DOM%. Deze codes worden door de runprocedure **waqpre.run**, voordat de executable **waqpre.exe** wordt aangeroepen, vervangen door het aantal lagen en het volgnummer van het deeldomein. Door deze codes te gebruiken in INCLUDE-regels, kunnen de

gebiedsschematisatie en forcings heel precies per deeldomein worden gespecificeerd.

## 1.4 Aansturing van de runprocedure **waqpre.run**

Het gebruik van de runprocedures **waqpre.run** en **waqpro.run** wordt uitgebreid beschreven in [ 3]. De invoerparameters maken het mogelijk om de verwerking van de invoer te sturen en zo te bepalen wat er in Figuur 3 gebeurt in het 'systeem'-gedeelte.

Het gebruik van de runprocedure **waqpre.run** bij verticale verfijning lijkt sterk op dat gebruik bij een sequentiële berekening. Alleen zijn de volgende opties van belang bij verticale verfijning, en niet bij sequentiële berekening:

- ndom** Het aantal deeldomeinen
- decomp** De naam van de *areas*-file
- kmax** Een lijstje, met voor elk van de processoren het aantal lagen.
- buf\_prt** De *buffer-size* voor het programma **coppre.exe**

De runprocedure activeert de programma's **waqpre.exe** en **coppre.exe** eenmaal voor elk van de deeldomeinen. Het programma **waqpre.exe** creëert de SDS-file SDS-*<runid>*, waarin de invoer voor het hele domein is verwerkt. Daarna haalt **coppre** de informatie voor het betreffende deeldomein uit deze SDS-file haalt en bewaart deze in de SDS-file met de naam SDS-*<runid>*-*<xxx>*. Met *<xxx>* wordt het volgnummer van het deeldomein aangegeven. In de message-file, waqpre-m.*<runid>*, staat na afloop alleen de uitvoer over het laatste deeldomein.

In toekomstige versies zullen **waqpre** en **coppre** wellicht maar eenmaal gedraaid hoeven worden voor alle deeldomeinen waarvoor de *siminp*-file identiek is.

**NB:** het deeldomein met het hoogste volgnummer moet altijd de fijnste laagverdeling hebben. De reden hiervoor is dat later alle simulatieresultaten in de (laatste) file SDS-*<runid>* vergaard moeten kunnen worden door het programma **coppre**.

## 1.5 Toegestane laagverdelingen

Bij gebruik van verticale verfijning kan in de verschillende deeldomeinen voor verschillende laagverdelingen gekozen worden. Hierbij kan precies aangesloten worden bij de behoefte aan detail in de verticale richting.

Niet alle denkbare laagverdelingen zijn echter toegestaan. In Sectie 1.4 is al uitgelegd dat de fijnste laagverdeling in het laatste deeldomein gebruikt moet worden om de resultaten in één SDS-file te kunnen verzamelen. In deze sectie wordt dat

idee iets preciezer uitgewerkt. In Figuur 5 worden twee mogelijke laagverdelingen getoond. De bovenste is toegestaan als het linkerdomein het laatste is; de onderste niet. Hoewel ook in de onderste laagverdeling de meeste lagen worden gebruikt in het linkerdomein, worden in het linkerdomein immers de bovenste twee lagen van de ‘fijnste verdeling’ samen genomen.

Title:  
layers.eps  
Creator:  
fig2dev Version 3.2 Patchlevel 3c  
Preview:  
This EPS picture was not saved  
with a preview included in it.  
Comment:  
This EPS picture will print to a  
PostScript printer, but not to  
other types of printers.

**Figuur 5: Alle laagverdelingen moeten een deelverzameling zijn van een fijnste verdeling, die gebruikt wordt in het laatste deeldomein. De onderste verdeling is dus niet toegestaan.**

Ook de verdeling in Figuur 6 is niet toegestaan. Voor het uitvoeren van de berekeningen is het voor **waqpro.exe** namelijk nodig om van elke koppeling tussen domeinen te bepalen welke van de twee de fijnste laagverdeling heeft (behalve als ze de zelfde laagverdeling hebben). Het is onmogelijk om te bepalen of de laagverdeling in het middelste domein in de figuur fijner of grover is dan in het rechterdomein.

Title:  
layers2.eps  
Creator:  
fig2dev Version 3.2 Patchlevel 3c  
Preview:  
This EPS picture was not saved  
with a preview included in it.  
Comment:  
This EPS picture will print to a  
PostScript printer, but not to  
other types of printers.

**Figuur 6: Ook hier is de onderste verdeling niet toegestaan: het is niet**

bepaald of de middelste laagverdeling grover of fijner is dan het rechter.

## 1.6 Runprocedure waqpro.run

Ook het gebruik van de runprocedure **waqpro.run** bij verticale verfijning lijkt sterk op het gebruik bij een sequentiële berekening. De interne werking van het systeem is echter veel uitgebreider, zoals blijkt uit de vergelijking van Figuur 2 en Figuur 3.

De runprocedure start eerst het programma **PVM** op. Via de *hostfile* (waarin de namen van de gebruikte computers kunnen worden ingevuld) en de waqpro-optie **-force\_pvm** (waarmee oude files van PVM kunnen worden verwijderd, zie [ 1 ], Secties 3.2 en 3.4) heeft de gebruiker enige invloed op de werking van **PVM**. De runprocedure **waqpro.run** start **coexec.exe** op. Het programma **coexec.exe** start vervolgens meerdere **waqpro.exe** processen op: één voor elk deeldomein. Nadat de tijdstappen zijn berekend door de **waqpro.exe**'s, verzamelt het programma **coppos.exe** de resultaten in een enkele SDS-file.

De volgende parameters zijn van belang bij een berekening met verticale verfijning:

- npart**            Het aantal deeldomeinen in de berekening (gelijk aan de parameter **-ndom** voor **waqpre.run**).
- fases**  
<fases>            De 'fases' (onderdelen) van de berekening die moeten worden uitgevoerd. Voor <fases> moet een van de volgende opties worden ingevuld:
  - all**            Voer de gehele berekening uit (default).
  - pro**            Voer alleen de simulatie uit: de resultaten hoeven niet te worden verzameld in de file SDS-<runid>.
  - col**            Verzamel de resultaten in de file SDS-<runid>. De simulatie is al uitgevoerd.
- buf\_prt**            de *buf\_size* voor het programma **coppos.exe**.

Vooraf uit het gebruik van de parameter **-fases** blijkt dat de structuur van het WAQUA-systeem uitgebreid is ten opzichte van de structuur bij een sequentiële berekening.

De uitvoer van de programma's wordt opgeslagen in de message-file **waqpro-m.<runid>**. Een aantal verschillen tussen

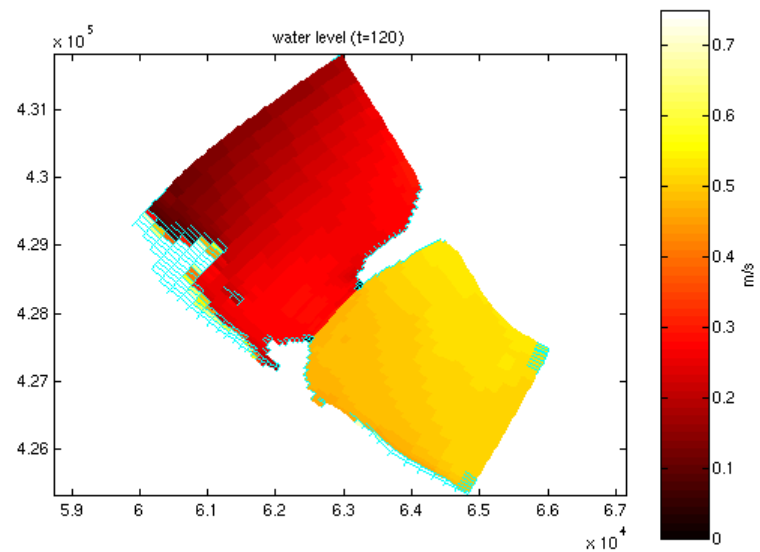
de message-file van een sequentiële berekening en de message-file bij gebruik van verticale verfijning valt op:

1. De message-file begint met uitvoer van het programma **coexec.exe**. Dit programma wordt alleen gebruikt bij parallele en domeindecompositie berekeningen. Het programma **coexec.exe** start de verschillende processen op, controleert of zij naar behoren functioneren, en verzamelt hun uitvoer in de message-file. In sommige gevallen geeft **coexec.exe** meldingen, omdat er een kans bestaat dat een van de processen ermee is opgehouden. In dat geval wordt een controle uitgevoerd, waarvan **coexec.exe** melding maakt.
2. De uitvoer van de verschillende processen wordt voorafgegaan door de markering ' 1 #', '2 #',... Deze markering geeft aan van welk rekenproces de uitvoer afkomstig is.
3. De uitvoerregels van de verschillende rekenprocessen is soms afgebroken: een regel uitvoer is dan over twee regels verdeeld. De reden hiervoor is dat het programma **coexec.exe** de uitvoer ontvangt in pakketten met een vaste omvang (bijvoorbeeld 1 kilobyte). Wanneer een pakket 'vol' is, wordt het afgedrukt, ook als de regel nog niet ten einde is. Als het volgende pakket dat **coexec.exe** ontvangt van hetzelfde rekenproces komt dan wordt dit erachter geplakt; is het van een ander rekenproces, dan wordt er op een nieuwe regel begonnen.

## 1.7

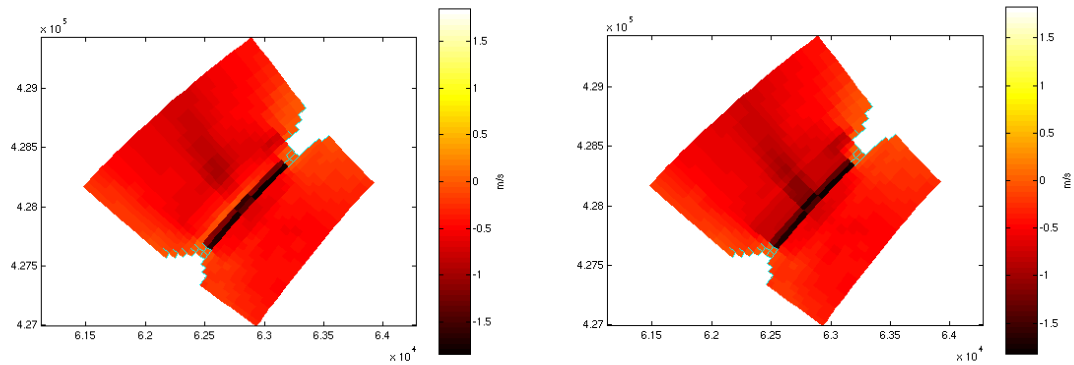
### Resultaten van een berekening met verticale verfijning

De resultaten van de berekeningen in alle deeldomeinen worden verzameld in de SDS-file SDS-*<runid>*. In sommige van de deeldomeinen is echter gerekend met minder lagen dan het aantal dat in deze file gebruikt wordt. De informatie uit deze deeldomeinen wordt daarom (met behulp van heel eenvoudige interpolatieformules) uitgebreid tot het maximum aantal lagen. Dit heeft tot gevolg dat de rekenresultaten kunnen worden gevisualiseerd op precies dezelfde manier als de resultaten van een sequentiële berekening, zoals wordt geïllustreerd in Figuur 7.

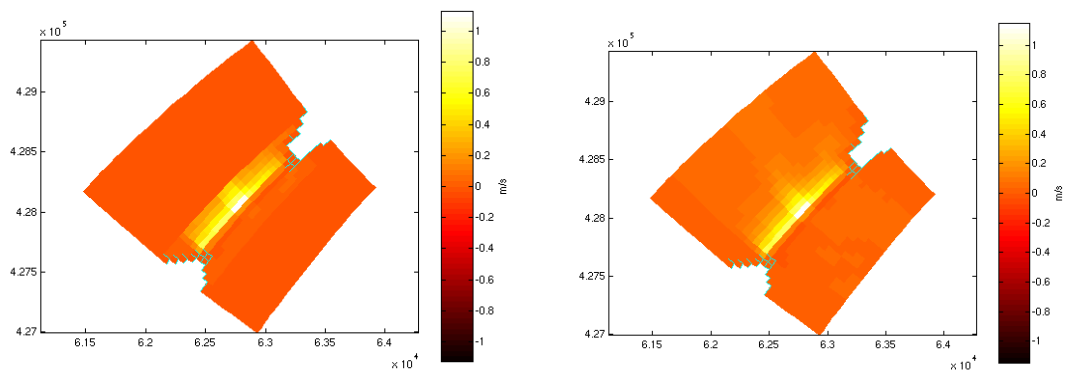


**Figuur 7: waterstanden, berekend met verticale verfijning**

Het aanvullen van de rekenresultaten tot het maximum aantal lagen heeft echter wel gevolgen voor de interpretatie van de rekenresultaten. In de SDS-file staat immers een mate van detail die in de berekening helemaal niet beschikbaar was. Dit wordt geïllustreerd in Figuur 8 en Figuur 9. Aangezien de fijnere laagverdeling nodig is in die delen van het rekengebied waarin gelaagdheid optreedt, is het interessant om na afloop van de berekening te kijken of de gelaagdheid ook inderdaad in het verfijnde gedeelte voorkomt. Hierbij kan het echter voorkomen dat de rekenresultaten een te positief beeld schetsen. In het linker-gedeelte van Figuur 9 is helemaal geen verschil te zien tussen de snelheden vlak boven de bodem (in laag 4) en de snelheden in de laag daar direct boven (laag 3). De reden is echter niet, dat de stroomsnelheid niet varieert in de diepte, maar wel dat de snelheid in de onderste twee lagen feitelijk kopieën zijn van de snelheid in de onderste (dikke) laag van de deelgebieden waar maar met 2 of 1 laag is gerekend. Het rechterplaatje in Figuur 9 is verkregen door de berekening te herhalen met 4 lagen in het hele gebied. Nu is te zien dat de stroming inderdaad weinig verschilt in de verticaal, behalve in het eerder verfijnde gebied.



**Figuur 8:** De snelheden in u-richting, na 120 simulatieminuten. Links: in de onderste laag. Rechts: in de laag daarboven (3e laag).



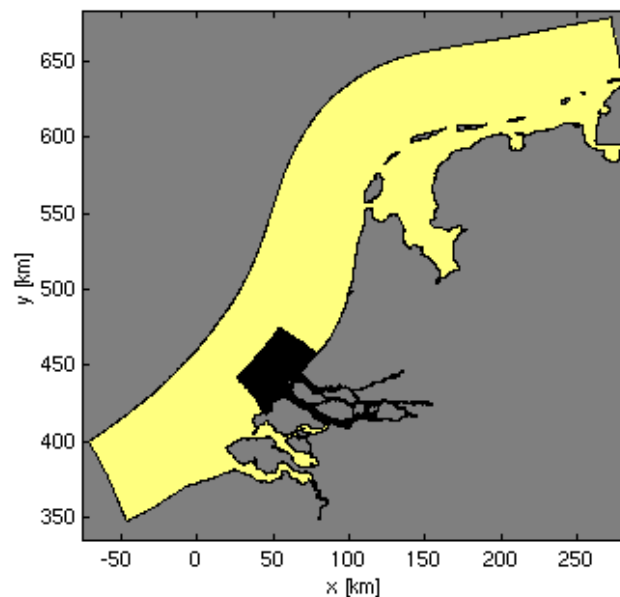
**Figuur 9:** het verschil tussen de u-snelheid in de onderste twee lagen. Links: met verticale verfijning. Rechts: berekend met 4 lagen overall.

Er is nog een mogelijk nadeel van het verzamelen van alle rekenresultaten in een SDS-file, waarbij extra laag-informatie wordt verkregen door interpolatie. Deze SDS-file wordt namelijk erg groot. Er kan daarom in sommige gevallen voor worden gekozen om het verzamelen van de gegevens over te slaan door middel van de optie **-fases pro** voor de runprocedure **waqpro.run**.

## 2 Domein decompositie met horizontale verfijning

Horizontale verfijning kan worden gebruikt voor berekeningen van situaties waar een gedeelte van het rekengebied met een fijner rooster uitgevoerd dient te worden of als meerdere aan elkaar grenzende roosters samen doorgerekend dienen te worden.

Een voorbeeld hiervan is de koppeling van het Kuststrookmodel aan het Zeedeltamodel, zoals te zien is in Figuur 10. Het Zeedeltamodel biedt veel meer detail dan het Kuststrookmodel, maar moet gevoed worden met de randvoorwaarden uit het Kuststrookmodel. Beide modellen maken deel uit van de zogenaamde ‘modellentrein’, die bestaat uit een groot aantal aan elkaar passende modellen.



**Figuur 10: Weergave van het kuststrook model waarbij is aangegeven welk deel door het Zeedelta model berekend moet worden.**

Een gekoppelde simulatie met de DDHOR-programmatuur vergt minder handelingen van de gebruiker dan de koppeling via randvoorwaarden die uit het Kuststrookmodel worden gelezen en bij het Zeedeltamodel worden opgelegd. Een ander voordeel van de koppeling met de DDHOR-programmatuur is dat niet alleen de resultaten van het grove model een invloed hebben op het fijne, maar dat ook de resultaten van het fijne model een invloed hebben op het grove model. Dit kan bijdragen aan de nauwkeurigheid van de oplossingen.

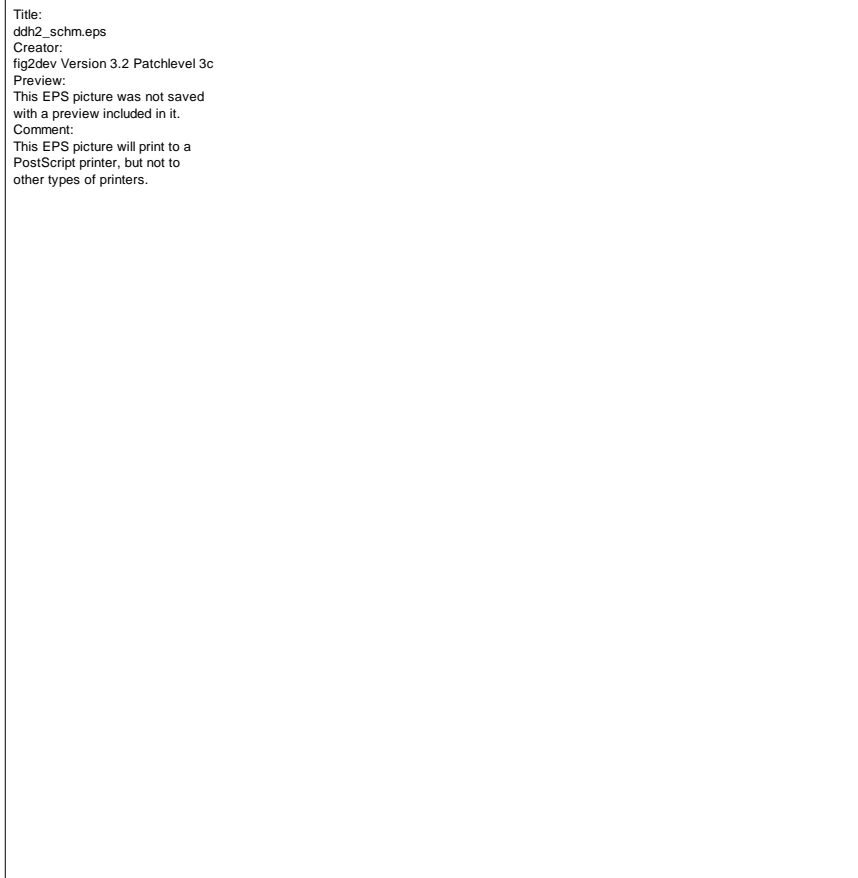
Dit document beschrijft hoe simulaties kunnen worden uitgevoerd op roosters zoals in Figuur 10 wordt getoond. Hierbij worden verwijzingen gegeven naar andere documenten



voor aanvullende informatie. Opdrachten worden gebruikt ter illustratie.

## 2.1

### Globaal overzicht van het DDHOR systeem



Title:  
ddh2\_schm.eps  
Creator:  
fig2dev Version 3.2 Patchlevel 3c  
Preview:  
This EPS picture was not saved  
with a preview included in it.  
Comment:  
This EPS picture will print to a  
PostScript printer, but not to  
other types of printers.

**Figuur 11: Globaal overzicht van een DDHOR-simulatie.**

Het verloop van de berekening met gebruik van horizontale verfijningen wordt geïllustreerd door Figuur 11. De figuur is veel uitgebreider dan het overzicht van een sequentiële simulatie (Figuur 2) of een DDVERT-simulatie (Figuur 3). Wel is te zien dat de meeste systeemonderdelen tussen de strepen staan die het ‘systeem’ scheiden van de ‘gebruikersinvoer’ en de ‘gebruikersuitvoer’. Deze onderdelen worden grotendeels aan het gezicht van de gebruiker onttrokken, en daarom hoeft in dit document niet veel aandacht aan deze onderdelen besteed te worden.

Uiteraard zijn de gedeelten boven de systeemlaag (de gebruikersinvoer) en eronder (gebruikersuitvoer) wel van groot belang voor de gebruiker. Anders dan bij DDVERT, zijn er meerdere *siminp*-files te zien bij de gebruikersinvoer: één voor elk van de roosters (domeinen). Speciale include-files zijn niet nodig. Wel is een *areas*-file nodig voor elk van de domeinen. Deze *areas*-file dient om aan te geven welk gedeelte van het in de *siminp*-file beschreven gebied gebruikt moet worden. Dat maakt het mogelijk om overlappende gebiedsbeschrijvingen

aan elkaar te koppelen: het Kuststrookmodel bevat bijvoorbeeld een grove versie van het Zeedeltamodel (het zwart gekleurde gebied in Figuur 10). Dit deel van het Kuststrookmodel moet buiten werking worden gesteld wanneer het Kuststrookmodel gekoppeld met het Zeedeltamodel wordt gebruikt. Ook een eventuele verdere opdeling van de domeinen ten behoeve van parallelle verwerking van de simulatie wordt in de *areas*-file opgegeven. Naast de *siminp*- en *areas*-files is ook een *configuratiebestand* nodig, waarin beschreven staat welke domeinen aan elkaar gekoppeld moeten worden.

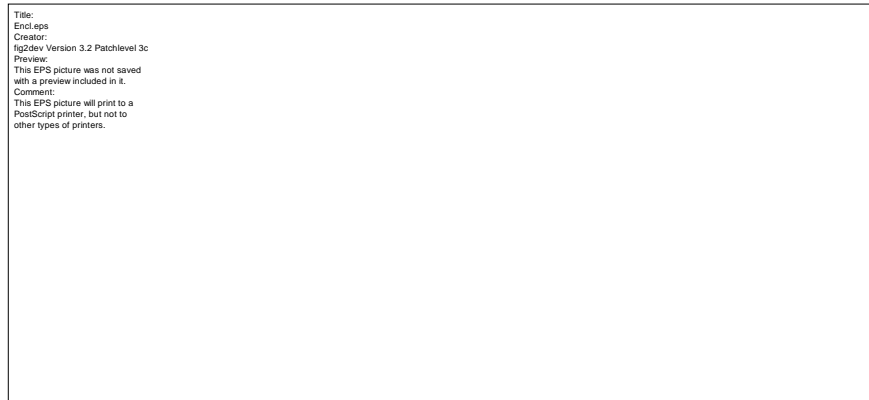
De verschillende invoerbestanden worden in de komende Secties beschreven. In Sectie 2.2 wordt beschreven hoe *areas*-files zo gemaakt kunnen worden, dat de verschillende domeinen in de berekening precies aan elkaar passen. Vervolgens wordt in Sectie 0 beschreven hoe in het configuratiebestand kan worden beschreven welke domeinen gekoppeld moeten worden doorgerekend.

Wanneer alle invoer gereed is, kan de simulatie worden opgestart. Dit wordt beschreven in Sectie 2.4. De simulatieresultaten komen terecht in de uitvoer, die in Sectie 2.5 wordt beschreven.

## 2.2 Aan elkaar passende domeinen

Domeinen kunnen gekoppeld worden doorgerekend, wanneer ze precies aan elkaar passen. In geval van verfijning moeten bovendien de roosterlijnen van het grove domein ononderbroken doorlopen in het fijne. Het is een precies werkje om roosters aan elkaar te passen, en het vergt een nauwkeurige kennis van de ‘gestaggerde’ roosters die in WAQUA worden gebruikt.

De WAQUA-roosters bestaan uit zogenaamde ‘controlevolumes’ of ‘cellen’: aan elkaar passende rechthoeken, die samen het rekengebied overdekken. Op kromlijnige roosters zijn de controlevolumes enigszins verwrongen rechthoeken. Om het rekendomein heen ligt de zogenaamde WAQUA-enclosure, zoals te zien is in Figuur 12.



**Figuur 12: Een rekendomein bestaat uit controlevolumes. De enclosure ligt om het domein heen.**

Elk controlevolume wordt begrensd door zijn zijden, die ‘cell-faces’ genoemd worden. Verschillende grootheden worden op verschillende plaatsen in het WAQUA-rooster berekend:

- De waterstanden worden berekend in het midden van de controlevolumes. De middelpunten van de controlevolumes worden daarom ook ‘waterstandspunten’ genoemd.
- De bodemligging wordt berekend op de hoekpunten van de controlevolumes. De hoekpunten van de controlevolumes worden daarom ‘dieptepunten’ genoemd.
- Op de middens van de cell-faces (de zijden van de controlevolumes) worden de normaalcomponenten van de stroomsnelheden berekend. Deze middens van de cell-faces worden daarom snelheidspunten genoemd.

Elk controlevolume heeft dus een waterstandspunt in het midden, en wordt begrensd door vier snelheidspunten en vier dieptepunten. Omdat elk van de dieptepunten op hun beurt een hoekpunt zijn van vier verschillende controlevolumes, zijn er niet vier maal zoveel, maar ongeveer even veel dieptepunten als controlevolumes. Iets dergelijks gaat op voor de snelheidspunten: er zijn tweemaal zoveel snelheidspunten als controlevolumes: een  $u$ -snelheidspunt en een  $v$ -snelheidspunt per controlevolume. Binnen WAQUA wordt daarom de conventie gebruikt, dat een waterstandspunt, het  $u$ -snelheidspunt rechts van het waterstandspunt, het  $v$ -snelheidspunt boven het waterstandspunt en het dieptepunt rechtsboven het waterstandspunt bij elkaar horen: alle worden met dezelfde  $(m,n)$ -coördinaten aangeduid. Deze adresseringsconventie wordt geïllustreerd in Figuur 13.

```

Title:
Punten.eps
Creator:
fig2dev Version 3.2 Patchlevel 3c
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

```

**Figuur 13: Elk controlevolume bevat een waterstandspunt, en wordt begrensd door twee u-snelheidspunten, twee v-snelheidspunten en vier dieptepunten. De snelheidspunten rechts en boven, en het dieptepunt rechtsboven het waterstandspunt hebben dezelfde (m,n)-coördinaten.**

Aangezien de domeinen bestaan uit controlevolumes, moet bij het aan elkaar passen gekeken worden naar deze controlevolumes. In Figuur 14 is te zien dat de roosters aan elkaar passen wanneer op de rand tussen de domeinen de dieptepunten van het grove rooster ook dieptepunten zijn van het fijne rooster. Merk op dat de roosterlijnen met waterstandspunten in het geval van een 1:even verfijning (zoals de 1:2-verfijning in de figuur) *niet* van het grove domein doorlopen in het fijne domein.

```

Title:
Connection.eps
Creator:
fig2dev Version 3.2 Patchlevel 3c
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

```

**Figuur 14: Een fijn rooster dat aan een grof rooster past.**

Wanneer rechthoekige roosters aan elkaar gepast worden, treedt een onverwachte moeilijkheid op. De reden is, dat in WAQUA de oorsprong van het rooster (aangegeven met XORIGIN en YORIGIN in de *siminp*-file) overeenkomt met het waterstandspunt met de coördinaten  $(\mathbf{m}, \mathbf{n}) = (\mathbf{1}, \mathbf{1})$ . In Figuur 15

is te zien dat de roosters in dat geval alleen aan elkaar gekoppeld kunnen worden, als ze een andere oorsprong hebben. Immers, wanneer de beide roosters dezelfde oorsprong hebben, komen de dieptepunten van de roosters niet met elkaar overeen.

```
Title:
Oorsprong.eps
Creator:
fig2dev Version 3.2 Patchlevel 3c
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.
```

**Figuur 15:** bij rechtlijnige roosters met een verfijningsverhouding 1:2 komen geen samenvallende dieptepunten voor als de oorsprong van het rooster gelijk is.

## 2.3

### Een configuratiebestand maken

Het WAQUA-systeem moet door de gebruiker verteld worden welke roosters aan elkaar gekoppeld moeten worden. De gebruiker geeft dit op via het *configuratiebestand*. De syntaxis van een configuratiebestand wordt uitgelegd op pagina 25 van de gebruikershandleiding voor domeindecompositie [ 1 ]. De volgende informatie staat in deze file;

1. Van elk domein:
  - a. Een door de gebruiker gekozen naam;
  - b. De *runid*, en experimentnaam;
  - c. De naam van de te gebruiken executable. Door een speciale keuze van deze naam kan een niet-standaard buffergrootte ingesteld worden (zie [ 1 ]).
2. Een toegestane afwijking, wanneer samenvallende dieptepunten gezocht worden. Door afrondfouten kan het gebeuren dat de dieptepunten van de beide roosters niet precies, maar ongeveer overeen komen.

## 2.4

### Een simulatie opstarten

Een DDHOR-simulatie bestaat, net als een sequentiële simulatie, uit een preprocessings- en een processingsfase. De simulatie-uitkomsten kunnen vervolgens in een postprocessingsfase worden gevisualiseerd of anderszins gebruikt.

In de preprocessingfase wordt de runprocedure **waqpre.run** aangeroepen voor elk van de domeinen. De verschillende opties van de runprocedure voor DDHOR worden beschreven in de Quick reference guide [ 3 ] op pagina 24. De runprocedure roept het programma **waqpre.exe** aan om de inhoud van de *siminp*-file te verwerken tot een SDS-file. Vervolgens wordt het programma **copppe.exe** aangeroepen, dat SDS-files maakt waarin alleen de actieve delen van het rekengebied staan. Wanneer de domeinen niet verder zijn opgedeeld ten behoeve van parallelle verwerking, maakt **copppe.exe** maar één SDS-file aan.

De processingsfase wordt aangezet met de runprocedure **waqpro.run** (Quick reference guide [ 3 ], pagina 39). Dit script start eerst het programma **PVM** op. Via de *hostfile* (waarin de namen van de gebruikte computers kunnen worden ingevuld) en de waqpro-optie **-force\_pvm** (waarmee oude files van PVM kunnen worden verwijderd, zie [ 1 ], Secties 3.2 en 3.4) heeft de gebruiker enige invloed op de werking van **PVM**. Het programma **PVM** start **coexec.exe** op. Dit programma controleert of de domeinen inderdaad op de opgegeven manier gekoppeld kunnen worden. Als de gebruiker dat met de optie **-check\_only** heeft opgegeven, is dat ook het enige dat er gebeurt. Anders worden nu de verschillende processen opgestart voor de simulatie. Alle uitvoer van de processen wordt afgedrukt in de *message*-file **waqpro-m.<runid>**. De simulatieresultaten worden toegevoegd aan de SDS-files van de verschillende deeldomeinen. Nadat de tijdstappen zijn doorlopen, voegt het programma **coppops.exe** de informatie uit de SDS-files van de deeldomeinen van elk domein bij elkaar in één enkele SDS-file. Na afloop is er dus één SDS-file voor elk van de domeinen.

## 2.5 De uitvoer van de simulatie

De uitvoer van de simulatie bestaat uit de volgende drie onderdelen:

1. De *message*-file **waqpro-m.<runid>**. Er is maar één *message*-file voor de hele simulatie. De *message*-file bestaat uit de volgende onderdelen:
  - a. Uitvoer van de controle die **coexec.exe** uitvoert om te kijken of de roosters inderdaad aan elkaar gekoppeld kunnen worden.
  - b. Uitvoer over het opstarten van processen.
  - c. Uitvoer van elk van de processen, betreffende het maken van de koppeling met andere deeldomeinen. Deze uitvoer wordt voorafgegaan door de codes ' 1 #', ' 2 #', et cetera, om aan te geven van welk proces het afkomstig is.

- d. Uitvoer van het eerste proces, betreffende de voortgang van de simulatie. Tegelijkertijd ook eventuele meldingen van problemen door andere processen. Ook deze uitvoer wordt voorafgegaan door de codes ' 1 #', ' 2 #', et cetera, om aan te geven van welk proces het afkomstig is.
  - e. Uitvoer van het programma **coppere.exe**, voor elk van de domeinen.
2. De SDS-files, genaamd SDS-*<gridid>*. Voor elk domein is er een SDS-file.
  3. De *report*-files **waqpro-r.<gridid>-<xxx>**, waarin de report-informatie staat van het deeldomein van rooster *<gridid>*, met het volgnummer *<xxx>*.

## Bibliografie

- [ 1 ] *User's guide for parallel WAQUA/TRIWAQ and for domain decomposition*, Version 12, Simona Report 2000-01, December 2000.
- [ 2 ] Bas van 't Hof, Nils van Velzen en Edwin Vollebregt (VORtech Computing), *User's guide VISIPART*, September 2002.
- [ 3 ] *Quick reference guide WAQUA*, Version 10.15-DDH-01, Maintained by VORtech Computing, July 2001.