

Toekomstige Ontwikkeling van Domein Decompositie in WAQUA en TRIWAQ

Technisch Rapport TR03-02 versie 0.9 (29 augustus 2003)

Datum

29 augustus 2003

Auteur(s)

dr.ir. E.A.H. Vollebregt

dr.ir. M.R.T. Roest

In opdracht van

Rijkswaterstaat/RIKZ (overeenkomst RKZ-1260)

Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enigerlei wijze hetzij elektronisch, mechanisch, door fotokopieën, opnamen, of op enige andere manier, zonder voorafgaande schriftelijke toestemming van de opdrachtgever danwel VORtech Computing, Postbus 260, 2600 AG DELFT.

© VORtech Computing 2003.

Log-sheet

Versie	Auteur	Datum	Opmerkingen
0.9	EV/MR	01-09-2003	DDHRVR01: concept-versie van project RKZ-1260

Samenvatting

In de afgelopen jaren is WAQUA/TRIWAQ sterk uitgebreid met mogelijkheden voor het gebruiken van domein decompositie met horizontale en verticale roosterverfijning. Deze functionaliteit voorziet in belangrijke behoeftes omtrent het flexibeler kunnen definiëren van omvangrijke gebiedsschematisaties en het reduceren van de daarvoor benodigde rekentijd. Ze wordt inmiddels operationeel gebruikt in diverse projecten van de Rijkswaterstaat.

Dit rapport geeft beknopte werkplannen voor verdere ontwikkeling van de domein decompositie functionaliteit in een tweetal richtingen: de pre- en postprocessing, en nieuwe soorten koppelingen.

Met betrekking tot de pre- en postprocessing is het volgens ons gewenst dat de gebruiker domein decompositie berekeningen veel meer als een geheel kunnen behandelen, in plaats van per (deel-)domein apart invoer aan te leveren en resultaten te visualiseren. Centraal in de aanpak die we hiervoor voorstellen staat een configuratiefile voor de gehele berekening. Hierin wordt aangegeven welke domeinen er zijn (DDHOR), welke siminp-files de domeinen beschrijven, wat de roosteropdeling per domein is voor verticale verfijning, en welke laagafhankelijke informatie daarbij moet worden gebruikt. Er wordt een stappenplan gepresenteerd waarmee de configuratiefile kan worden ingevoerd; eerst in het rekensysteem zelf, en dan stap-voor-stap in IPW, de interactieve preprocessor voor WAQUA.

Met betrekking tot uitbreiding van de mogelijke koppelingen in domein decompositie berekeningen wordt er een drietal categoriën onderscheiden:

1. uitbreidingen van de huidige functionaliteit, o.a. koppeling van bolcoördinaten met kromlijnige coördinaten, en gebruik van dynamische barriëresturing,
2. verschillende procesmodellen per domein, zoals aan/uitschakelen van k -Nikuradse ruwheidsberekening, zouttransport, $k - \epsilon$ turbulentiemodel, etc.
3. verschillende tijdstappen per domein, koppeling op tijdstapbasis naar andere modellen, en koppeling van WAQUA aan TRIWAQ of van roosters met verschillende orientaties.

Met betrekking tot deze uitbreidingen is onze mening dat de ontwikkeling van de tijdstap-koppeling snel moet worden opgestart via een prototyping-activiteit om inzicht te krijgen in de problematiek en mogelijke oplosrichtingen. Daarnaast denken we dat er met het kunnen aan/uitschakelen van transport en turbulentie per deeldomein met vrij weinig werk een aanzienlijke performancewinst kan worden behaald.

Inhoudsopgave

Log-sheet	ii
Samenvatting	iii
1 Inleiding	1-1
1.1 Inleiding domein decompositie	1-1
1.2 Doel van het huidige rapport	1-1
1.3 Indeling van dit rapport	1-2
1.4 Gerelateerde documenten	1-2
2 Werkplan met betrekking tot pre- en postprocessing	2-1
2.1 Inleiding	2-1
2.2 Indeling van dit hoofdstuk	2-2
2.3 Huidige werkwijze en ondersteuning	2-2
2.4 Visie over de toekomstige werkwijze en ondersteuning	2-3
2.5 Uitwerking van de aanpak met configuratiefile	2-5
2.6 Korte bespreking van mogelijke alternatieven	2-6
2.7 Werkplan	2-8
3 Werkplan met betrekking tot nieuwe koppelingen	3-1
3.1 Inleiding	3-1
3.2 Indeling van dit hoofdstuk	3-2
3.3 Uitbreidingen van de huidige domein decompositie functionaliteit	3-3
3.4 Per domein aan- of uitzetten van verschillende procesmodellen	3-6
3.5 Koppelen van ongelijksoortige rekenprocessen	3-9
3.6 Werkplan	3-14
Referenties	3-15

Hoofdstuk 1

Inleiding

1.1 Inleiding domein decompositie

Domein decompositie is een algemene techniek voor het vergroten van de vrijheid van gebruikers van WAQUA en TRIWAQ in het definiëren van gebiedsschematisaties. Bijvoorbeeld biedt het de mogelijkheid om het rekenrooster dat wordt gebruikt te verfijnen in gebieden waarin men sterk is geïnteresseerd, of in gebieden waar zich zeer lokale processen afspelen (rond sluizen en dergelijken). Tegelijkertijd kan met domein decompositie de doorlooptijd voor een simulatie sterk worden teruggebracht doordat fijne vermazing in het ene deel van het domein kan worden gebruikt samen met grovere vermazing in een ander deel.

De afgelopen jaren is domein decompositie gerealiseerd in de standaardversies van WAQUA en TRIWAQ langs de lijnen die zijn uitgezet met de introductie van parallel rekenen. Waar parallel rekenen inmiddels gemeengoed is geworden bij de gebruikers van WAQUA/TRIWAQ, wordt domein decompositie nog wat beperkter gebruikt. Niettemin zijn al een aantal projecten uitgevoerd die ondenkbaar waren geweest zonder deze functionaliteit, waaronder bijvoorbeeld een studie naar een nevengeul in de Maas. De belangstelling voor deze functionaliteit is ondermeer gebleken in een hoge opkomst voor een cursus parallel rekenen en domein decompositie die in 2002 is gegeven. Daarbij is duidelijk naar voren gekomen dat de aangeboden functionaliteit in een behoefte voorziet.

1.2 Doel van het huidige rapport

Het voorliggende rapport geeft beknopte werkplannen voor een tweetal gewenste ontwikkelingen. In de eerste plaats is dit een werkplan voor aanpassingen aan de pre- en postprocessing ten behoeve van domein decompositie. Dit wordt ingegeven door het feit dat de interactieve preprocessor voor WAQUA (IPW) niet overweg kan met de huidige domein decompositie files.

Ten tweede zal een werkplan gepresenteerd worden voor het introduceren van nieuwe soorten koppelingen, waaronder

1. uitbreidingen van de huidige functionaliteit, o.a. koppeling van bolcoördinaten met kromlijnjige coördinaten, en gebruik van dynamische barrieresturing,
2. verschillende procesmodellen per domein, zoals aan/uitschakelen van zouttransport, k-Nikuradse berekening, etc.

3. verschillende tijdstappen per domein, en tijdstapkoppeling naar andere modellen.

1.3 Indeling van dit rapport

Hoofdstuk 2 geeft het werkplan voor de pre- en postprocessing. Hoofdstuk 3 geeft het werkplan voor de nieuwe soorten koppelingen. Beide werkplannen kunnen afzonderlijk gelezen worden en bevatten hun eigen aanbevelingen.

1.4 Gerelateerde documenten

Een eerste verkenning van de introductie van domein decompositie bij RIKZ is vastgelegd in het TNO-rapport [1]. Dat rapport is later in meer detail uitgewerkt in het rapport "Ontwikkeling Domein Decompositie op Middellange Termijn" [3]. Het voorliggende rapport, en dan met name het werkplan voor de nieuwe soorten koppelingen bouwt voort op deze twee rapporten.

De achtergronden van parallel rekenen en domein decompositie met verticale en horizontale verfijning worden verder toegelicht in het cursusmateriaal van de cursus "Domein decompositie en Parallel rekenen in SIMONA" [2, 4, 5], met name het cursusboek voor Module 3.

Een overzicht van WAQUA/TRIWAQ met domein decompositie kan ook worden verkregen via de verschillende vormen van gebruikersdocumentatie:

- de User's Guide for parallel WAQUA/TRIWAQ and for Domain Decomposition;
- de Quick Reference Guide WAQUA;

Gedetailleerdere informatie over verticale verfijning is te vinden in het detailontwerp [6] van de DDVERT functionaliteit zoals die is geoperationaliseerd in SIMONA. Met betrekking tot de DDHOR functionaliteit is gedetailleerde informatie te vinden in het betreffende detailontwerp [7].

Voor meer algemene informatie over parallel rekenen, domein decompositie en over de in dit document gehanteerde terminologie is het reeds genoemde cursusmateriaal een goed startpunt, met name Module 1 van de cursus.

Hoofdstuk 2

Werkplan met betrekking tot pre- en postprocessing

2.1 Inleiding

In de afgelopen jaren zijn de belangrijkste mogelijkheden met betrekking tot domein decompositie voor de gebruiker gerealiseerd. Ook is er ervaring opgedaan met de nieuwe functionaliteit in operationele toepassingen. Hiermee is inzicht verkregen in de wensen die dit gebruik oproept ten aanzien van pre- en postprocessing. Voor de pre- en postprocessing voor domein decompositie berekeningen wordt momenteel gebruik gemaakt van VISIPART, voor het partitioneren van rekenroosters, en sinds kort van IPW, voor het matchen van roosters in een DDHOR berekening. Maar veel van de voorbereidingen voor domein decompositie gebeuren nog met de hand. Om het gebruik van domein decompositie te faciliteren, is het wenselijk dat er betere ondersteuning komt voor de pre- en postprocessing.

Een concrete aanleiding om hier op dit moment aandacht aan te besteden is een probleem met het gebruik van IPW voor het maken van invoer voor DDVERT berekeningen. DDVERT maakt gebruik van constructies %KMAX%, %LAYER% en %DOMAIN% in de siminp-file. Deze constructies zullen in het vervolg aangeduid worden met de term %-variabelen. Omdat de preprocessor IPW onder water de generieke preprocessor van SIMONA `siprep` gebruikt om een invoerfile te lezen, en `siprep` niet overweg kan met de %-variabelen, loopt IPW vast.

Daarnaast bestaat er behoefte om de laagafhankelijke informatie op een goede manier op te slaan, om later te kunnen zien welke informatie er in een run is gebruikt, en eventueel een run te herhalen. Op dit moment wordt een gedeelte van de parameters van een run (gebruikte decompositie-file, aantal lagen per deeldomein) alleen beschreven in start-scriptjes waarmee runs worden opgestart. Deze start-scriptjes vormen dus eigenlijk een onderdeel van de specificatie van een uitgevoerde run. Dit is ongewenst, omdat de scriptjes er heel anders uitzien dan de modelinvoer en daarom niet als specificatie van de run worden onderkend.

In dit hoofdstuk wordt een visie gepresenteerd voor het vormgeven van de pre- en postprocessing voor domein decompositie en parallel rekenen. Daarbij wordt ook een aanpak geformuleerd voor bovenstaande twee concrete vragen.

2.2 Indeling van dit hoofdstuk

In dit hoofdstuk zal allereerst geschetst worden hoe er op dit moment door de gebruikers omgegaan wordt met parallel rekenen en domein decompositie en welke ondersteuning daarbij geboden wordt. Vervolgens zal een visie geformuleerd worden over de faciliteiten voor pre- en postprocessing die wenselijk zijn.

Daarna zullen een aantal voorstellen gedaan worden om tot verbeteringen te komen in de manier waarop laagafhankelijke informatie wordt aangeboden, om daarmee een antwoord te formuleren op de problemen met IPW en op de wens om configuratie informatie op een georganiseerde manier op te slaan.

Vervolgens zal het meest veelbelovende voorstel verder worden uitgewerkt en wordt een concreet werkplan gepresenteerd.

2.3 Huidige werkwijze en ondersteuning

Een uitgangspunt bij de ontwikkeling van de parallelle en domein decompositie functionaliteit is steeds geweest dat gebruikers die functionaliteit zo gemakkelijk mogelijk moeten kunnen gebruiken. Voor parallel rekenen is dit zondermeer gelukt: de gebruiker hoeft alleen het aantal processoren op te geven.

Bij de ontwikkeling van DDVERT was het nodig om de gebruiker iets meer zelf te laten doen: hij moet immers zelf bepalen waar hij de subdomeingrenzen legt en hoe hij de laagverdeling kiest. Door bij DDVERT uit te gaan van één enkele siminp-file waarbinnen de gebruiker variaties kan aanbrengen, is het gebruik nog redelijk overzichtelijk gehouden.

Het programma VISIPART geeft goede ondersteuning bij het werken met verticale verfijning en parallel rekenen. Men kan hiermee een partitionering van het rooster maken en wijzigen en men kan een advies krijgen over het te gebruiken aantal processoren per DDVERT deeldomein. De uitvoer van VISIPART is een partitioneringsfile die rechtstreeks gebruikt wordt door programma's in de run-procedures. Niettemin moet de gebruiker zelf include files maken (met geschikte namen) voor het aanbieden van informatie die samenhangt met het aantal lagen. Verder is een beperking van VISIPART dat er Matlab voor nodig is.

Bij DDHOR was het nodig om dieper in te grijpen in de manier waarop gekoppelde simulaties worden beschreven. Dit komt doordat de siminp-file sterk op (m, n) -coördinaten, dus op het gebruikte kromlijnige rooster is gebaseerd. Hierdoor is het niet mogelijk om meerdere domeinen met verschillende horizontale roosterfijnheden in een enkele siminp-file op te nemen. Daarom was het noodzakelijk om de gebruiker aparte siminp-files te laten aanleveren voor de verschillende DDHOR-domeinen. Per siminp-file mag door middel van een AREAS-file een gedeelte worden weggesneden (inactief gemaakt). En door middel van een configuratiefile wordt opgegeven uit welke domeinen de complete berekening bestaat.

Het wegsnijden van overlappende gedeeltes en goed op elkaar laten aansluiten van verschillende domeinen levert in de praktijk soms veel gedoe op. Hierbij kan VISIPART enige ondersteuning geven, maar sinds kort kan ook IPW hierbij worden gebruikt. Op dit moment kan IPW slechts één siminp-file tegelijkertijd behandelen. Daarbij kan men voor het beschouwde model een AREAS-file laten genereren door er een tweede rooster bij te laten tekenen en aan te geven waar de roosters op elkaar aan moeten sluiten. Het is nog niet mogelijk om meer dan twee roosters tegelijk te laten tekenen, hoewel dat wel voorzien wordt.

Bij het gelijktijdig gebruik van DDHOR en DDVERT (mogelijk ook nog gecombineerd met parallel rekenen) krijgt de gebruiker met alle zaken tegelijk te maken: met VISIPART AREAS files maken voor het wegsnijden van stukken domein en voor het aangeven van verticale verfijningsdomeinen, met IPW en met de hand verschillende siminp-files aansluitend maken, verticale verfijningsinformatie in de DDVERT domeinen maken (met de hand), partitionering van deeldomeinen (via VISIPART). Het beeld wordt dan tamelijk onoverzichtelijk met alle problemen van dien om een goed model op te tuigen.

2.4 Visie over de toekomstige werkwijze en ondersteuning

Het is wenselijk dat gebruikers in de toekomst domein decompositie simulaties meer als een geïntegreerd geheel kunnen gaan benaderen, zowel op het gebied van preprocessing als bij postprocessing.

Voor de postprocessing is dit evident; de eindgebruiker wil niet weten waar de subdomeinranden liggen, maar wil bijvoorbeeld de waterstand visualiseren voor het gehele gesimuleerde gebied. En voor de modelontwikkelaar is het nuttig om de aansluiting van de stroomtoestanden van verschillende domeinen op elkaar te kunnen onderzoeken.

Voor de preprocessing gelden vergelijkbare overwegingen. De gebruiker wil de bodemligging van alle domeinen samen kunnen bekijken, wil de aansluiting op DDHOR-randen kunnen onderzoeken, etc. Speciale aandacht gaat uit naar de aansluiting van de grid-enclosures van verschillende DDHOR-domeinen, en naar het netjes op elkaar aansluiten van de overlappen van grove en fijne domeinen aan weerszijden van een interface. Tenslotte wil de gebruiker geholpen worden bij het consistent houden van bepaalde gegevens die voorkomen in meerdere simulatie invoerfiles: waardes die identiek moeten zijn in alle domeinen zoals het tijdsframe en de iteratie-parameters, maar bijvoorbeeld ook waardes voor de windsnelheid en richting die voor de gebruiker niet gekoppeld zijn aan de afzonderlijke domeinen.

2.4.1 Gewenste situatie m.b.t. preprocessing

Met betrekking tot de preprocessing is het volgens ons gewenst dat de huidige opdeling in modules grotendeels gehandhaafd blijft. Hierin is er een ontkoppeling gemaakt tussen het beschrijven van de invoer enerzijds, bijvoorbeeld met IPW en baseline, en het verwerken ervan anderzijds in de preprocessor WAQPRE van WAQUA. Deze onderverdeling is bijvoorbeeld voordelig omdat er met de hand nog kleine aanpassingen aan de invoer kunnen worden gemaakt, onder andere op systemen waar IPW en baseline niet beschikbaar zijn (supercomputers, Linux PC's). Verder is het handig dat WAQPRE per domein afzonderlijk kan worden uitgevoerd, in plaats van dat het verwerken van de invoer voor alle domeinen tegelijk moet worden gedaan.

In het vervolg beschrijven we een mogelijk prettige werkwijze voor het beschrijven van domein decompositie berekeningen met behulp van IPW. Uitgangspunt daarin is dat alle domeinen van de berekening via IPW te benaderen zijn, en waar mogelijk ook gelijktijdig kunnen worden bekeken. Centraal in onze oplossing om dit te bereiken staat een configuratiefile voor de gehele gekoppelde berekening. In eerste instantie bestaat deze uit de lijst

van namen van de gebruikte domeinen, gaandeweg kan hier steeds meer informatie over de domeinen en deeldomeinen aan worden toegevoegd.

In IPW zou de gebruiker primair beginnen met het definiëren van een nieuw domein. De configuratiefile bestaat in dit geval alleen uit de naam van dit domein. Om het domein te vullen, zou de gebruiker een rooster kunnen inlezen en de bijbehorende modelinformatie invullen. Op enig moment kan hij aangeven dat hij binnen het rooster verticale verfijning wil gaan gebruiken. Hij moet dan aangeven hoe het rooster opgedeeld moet worden, door een AREAS-file te selecteren of (idealiter) in IPW zelf te bepalen. Dit laatste betekent dat dat VISIPART geïntegreerd is met IPW. Wanneer de gebruiker het rooster in n stukken verdeelt, dan ontstaan hiermee feitelijk n aparte deelmodellen. De configuratiefile bevat dan onder het oorspronkelijke domein n “bladeren” die de verschillende deelmodellen vertegenwoordigen.

Als de gebruiker in de configuratiefile het oorspronkelijke model selecteert, dan krijgt hij invoerschermen voor het domein als geheel: de waarden die hier ingevuld worden gelden voor alle deeldomeinen tenzij bij een deeldomein iets anders is gespecificeerd. Als de gebruiker een deeldomein selecteert dan krijgt hij invoerschermen voor het deeldomein, waarin in eerste instantie de defaults uit het globale domein zijn opgenomen. Deze kan de gebruiker “overriden” door er andere waarden in te vullen. IPW moet hierbij in de gaten houden dat het model als geheel consistent blijft.

Bij een deeldomein kan de gebruiker ook nog een partitionering opgeven ten behoeve van parallel rekenen. Het is gewenst dat deze partitionering alleen binnen het deeldomein gewijzigd kan worden: wijzigingen tasten de grenzen van het deeldomein niet aan. Dit is anders dan momenteel in VISIPART het geval is, waar de gebruiker zonder het in de gaten te hebben het aantal lagen in een gebiedje kan veranderen.

Wil men nu tevens horizontale verfijning gebruiken, dan kan men aan de configuratiefile een nieuw model toevoegen op het niveau van het oorspronkelijke model. Daarbij kan men ook de aansluiting van de roosters opgeven (zoals nu al in IPW mogelijk is). Dit nieuwe model kan men op dezelfde wijze verder opdelen als het eerste model. In eerste instantie gebeurt het editen van de modelinvoer per domein apart. Gaandeweg worden er steeds meer mogelijkheden geschapen om meerdere domeinen tegelijkertijd te visualiseren, en om gegevens van het ene domein te kopiëren naar het andere. Ook kunnen steeds meer consistentie-controles worden ingebouwd. Zodra het gehele model (inclusief DDHOR, DDVERT en parallel rekenen) op deze manier is gedefinieerd, kan men met één druk op de knop files genereren waarmee zonder verdere bewerking een run kan worden uitgevoerd.

2.4.2 Postprocessing

Voor postprocessing heeft de ervaring geleerd dat het niet wenselijk is om daarvoor een standaard pakket te ontwikkelen: er zullen altijd gebruikers zijn die dingen willen visualiseren die met het standaardpakket net niet mogelijk zijn, zeker in het licht van snelle ontwikkelingen aan de functionaliteit van de rekenprogrammatuur. Daarom lijkt het beter om uit te gaan van de ‘stekker’-gedachte: SIMONA biedt een aantal standaard tools om informatie op gestructureerde manier op te vragen, op basis waarvan gebruikers hun eigen visualisatietools kunnen samenstellen.

Het is wenselijk dat gebruikers een gekoppelde run (parallel, DDHOR, DDVERT, procesvariëaties) als één geheel kunnen visualiseren. Daarbij zou de gebruiker direct toegang moeten hebben tot informatie over hoe een bepaald deel van het domein is doorgerekend, om een

correcte interpretatie van de resultaten mogelijk te maken. Dit vereist dat deze configuratie op een gestructureerde manier is opgeslagen.

Bij het tonen van beelden van het gehele domein, moet zichtbaar zijn dat er in bepaalde delen met andere instelling gerekend is. In een zoutheidsplaatje zou bijvoorbeeld een gebied dat zonder zout is doorgerekend ook als zodanig herkenbaar moeten zijn. Dit om incorrecte interpretatie van resultaten te voorkomen. Het visualiseren kan worden vergemakkelijkt door het inactieve gedeelte van een DDHOR-domein weg te laten uit de resultaat-SDS-file van een berekening. Dit lijkt goed mogelijk op basis van de generieke mechanismen van COPPRE en COPPOS. Men hoeft zich dan geen zorgen meer te maken over artefacten in plaatjes die voortkomen uit het weergeven van informatie voor de inactieve gedeelten.

2.5 Uitwerking van de aanpak met configuratiefile

Centraal in de hierboven beschreven aanpak voor uitwerking van de preprocessing staat een configuratiefile waarin de gehele gekoppelde berekening wordt beschreven. In deze paragraaf wordt de betekenis van die configuratiefile verder uitgewerkt.

Het idee is om aan de huidige configuratiefile voor DDHOR en DDH+V verwijzingen toe te voegen naar de gebruikte simulatie invoerfiles:

```
DDHOR_DOMAINS
```

```
D 1 : GLOBAL
      NAME   = 'kuststrook',
      SIMINP = 'siminp.ks3'
      RUNID  = 'ks3', EXP = 'ks3'
```

Verder kan hierin de informatie over verticale verfijning worden toegevoegd. Ten eerste is dit de roosteropdeling voor verticale verfijning (optie `-decomp` van de runprocedure). Daarnaast betreft dit het aantal lagen per deeldomein en de include-files voor laagafhankelijke informatie. De configuratiefile geeft dan waarden die prevaleren boven de feitelijke waarde in een siminp-file. De siminp-file bevat dan een normale inhoud (zonder %-variabelen) zodat IPW deze file ook gewoon kan lezen en schrijven. Middels de configuratiefile worden bepaalde elementen uit de siminp-file vervangen. Anders gezegd: de siminp-file bevat de waarden van variabelen die invariant zijn over de verschillende deeldomeinen en de variaties worden ondergebracht in de configuratiefile. Een mogelijke invulling voor de DDVERT-informatie is

```
DDVERT_DIVISION
      NDOM   = 2
      DECOMP = 'areas.kust'
DDVERT_DOMAINS
      SB 1 : KMAX = 1, LET FILE 'layers.include' BE 'layers.kmax1'
      SB 2 : KMAX = 3, LET FILE 'layers.include' BE 'layers.kmax3'
```

Er kan worden afgesproken om in de siminp-file steeds de informatie op te nemen die geldt voor de fijnste verticale laagverdeling.

Merk op dat in dit voorstel wordt afgestapt van de %-variabelen. De namen van variabelen hoeven dan niet meer steeds het format "naam.%DOM" te hebben: de gebruiker is vrij

om de namen naar eigen believen aan te passen. Daarentegen moeten de namen nu wel per deeldomein in de configuratiefile worden opgegeven. De runprocedure kan uit de configuratiefile afleiden wat er in de siminp-file momenteel wordt gebruikt, en waarin dit per deeldomein moet worden vervangen. Dit parsen van de configuratiefile is wel wat lastig, maar heeft als voordeel dat er een siminp-file wordt gegenereerd die direct door `siprep` kan worden ingelezen, dus waarvoor er geen aanpassingen in WAQPRE zijn vereist. Een voordeel van deze opzet van de configuratiefile is dus dat de modulariteit van het systeem gehandhaafd blijft.

Met behulp van de configuratiefile kan het opstarten van domein decompositie en parallelle berekeningen worden vergemakkelijkt. Er hoeven minder parameters te worden doorgegeven aan de run-procedures. Alle configuratie-informatie wordt op een centrale plek opgeslagen, en kan hierdoor door verschillende programma's worden opgevraagd. Een voordeel daarbij is dat de verschillende soorten runs (DDHOR, DDVERT en parallel) op een meer uniforme manier kunnen worden opgegeven.

Het configuratiebestand kan in principe willekeurig complexe koppelingsinformatie gaan bevatten. Daarbij moet er wel voor worden opgepast dat er niet te veel structuren van de siminp-file worden overgenomen. Dat zou leiden tot een beheersprobleem omdat aanpassingen aan het siminp-formaat dan gevolgen kunnen hebben voor het formaat van de configuratiefile. Dit wordt momenteel omzeild door met file-namen te werken in plaats van de inhoud van de files zelf in de configuratiefile op te nemen (met uitzondering van de waarde van KMAX).

Toekomstige richtingen voor verdere ontwikkeling van de configuratiefile zijn om hierin ook nieuwere vormen van koppeling (proceskoppelingen) te configureren, om eventueel gebruik te gaan maken van XML, en om een aparte applicatie te ontwikkelen om de consistentie van een configuratie te controleren. Het eerste idee stemt overeen met de aanpak binnen OMS en HarmonIT, waarbij ook bepaalde informatie over een gekoppelde run in een aparte configuratiefile wordt bewaard. Een voordeel van XML is dat er gemakkelijk browsers voor te krijgen/maken zijn en dat het de standaard binnen OMS lijkt te gaan worden. Het doel van een aparte controle-module is tenslotte om COEXEC te verlossen van allerhande checks die er eigenlijk niet in thuishoren.

2.6 Korte bespreking van mogelijke alternatieven

Aanleiding tot het schrijven van dit hoofdstuk waren de twee vragen die in de inleiding zijn genoemd: het oplossen van de problemen met %-variabelen in IPW en het kunnen opslaan van alle informatie over DDVERT-berekeningen. Naast de hierboven gepresenteerde aanpak met een configuratiefile kunnen er ook verschillende alternatieven worden bedacht. In deze paragraaf worden een aantal alternatieven kort besproken, en afgezet tegen de hierboven gepresenteerde benadering:

- Het runscript van IPW arbitraire dummywaarden te laten invullen voor de %-variabelen.
- De SIMONA preprocessor aanpassen zodanig dat de %-variabelen direct als zodanig ingelezen kunnen worden.
- Het aantal deeldomeinen, de AREAS-informatie, en alle laagafhankelijke informatie expliciet opnemen in de siminp-file.

Deze worden nu verder besproken.

2.6.1 Dummy-waarden invoegen voor %-variabelen

Een eerste soort van alternatieven bestaat uit het maken van dusdanige aanpassingen dat IPW niet hoeft te weten dat er DDVERT wordt gebruikt. IPW werkt dan met een enkele waarde voor KMAX en de daarmee samenhangende keuze voor de include-files. Deze waardes moeten dan in de siminp-file worden ingevuld voordat deze door IPW wordt ingelezen.

Het grote nadeel van dit soort alternatieven is dat de gebruiker na afloop altijd een aantal zaken handmatig moet veranderen. IPW weet namelijk niet waar het %-variabelen moet invullen in de siminp-file die door hem wordt weggeschreven. Verder kan de gebruiker IPW alleen voor een gedeelte van het domein gebruiken. Dit is lastig voor het opgeven van informatie die per deeldomein verschillend is: ruwheidsformulering, wel/geen overlaten, en de initiële toestand die per laag verschillend kan zijn. Tenslotte komt hierin het bewaren van de configuratie-informatie voor DDVERT niet aan de orde.

Op langere termijn is het daarom onontkoombaar dat IPW wordt uitgebreid met kennis van samengestelde berekeningen.

2.6.2 Aanpassen SIMONA preprocessor

Een tweede soort van alternatieven gaat uit van uitbreiding van de generieke preprocessor van SIMONA met de mogelijkheid om macro's te definiëren. Dit is nuttige functionaliteit, onder andere ook voor het geautomatiseerd afregelen van modellen met het WAQAD-systeem.

Het afhandelen van macro's in `siprep` is een omvangrijke ingreep omdat er stukjes tekst moeten worden opgeslagen die later weer moeten kunnen worden teruggeroepen. Merk op dat bij de definitie van macro's `siprep` nog niet kan weten om wat voor soort informatie het gaat. De ingreep lijkt enigszins op het naprogrammeren van `cpp` of `sed` binnen SIMONA. Voordeel hiervan is wel extra flexibiliteit met betrekking tot het opgeven en gebruiken van macro's in include-files.

Zodra macro's kunnen worden opgegeven in een siminp-file komt de vraag hoe dit te gebruiken is binnen DDVERT. IPW moet dan op de een of andere manier de variabele informatie voor de deeldomeinen beheren en opslaan, en de run-procedure `waqpre.run` moet de informatie voor een van de deeldomeinen doorgeven aan `siprep` in WAQPRE. Ten opzichte van de huidige situatie valt er dus met deze oplosrichting weinig te verbeteren.

2.6.3 KMAX-en en AREAS-informatie opnemen in siminp-file

Een derde soort oplosrichtingen gaat uit van het opnemen van alle informatie van DDVERT-berekeningen in de simulatie invoerfile. Dit kan worden gedaan door een nieuw keyword op het hoogste niveau toe te voegen voor de DDVERT-informatie. Onder dit keyword worden dan de roosteropdeling en het aantal deeldomeinen gedefiniëerd, en vervolgens de waarde van KMAX en te gebruiken laagafhankelijke informatie per deeldomein gespecificeerd.

Binnen WAQPRE kan er op twee manieren met dit nieuwe keyword worden omgegaan. WAQPRE kan worden uitgebreid om SDS-files te genereren voor alle deeldomeinen, of kan voor ieder deeldomein apart worden opgestart. Dit laatste heeft de voorkeur omdat dit minder ingrepen in WAQPRE vereist en omdat de globale SDS-files voor alle deeldomeinen dan niet gelijktijdig te hoeven worden opgeslagen. De benodigde aanpassingen aan WAQPRE lijken dan enigszins op de uitbreiding van `siprep` uit de vorige paragraaf: WAQPRE leest de

laagafhankelijke informatie (definitie van macro's) apart van de rest van de modelinvoer in, en roept deze definities op zodra om de betreffende informatie (gebruik van macro's) wordt gevraagd.

Deze oplosrichting lijkt sterk op die van de voorgestelde configuratiefile, vrijwel dezelfde informatie wordt nu in de `siminp`-file opgenomen. Wel is er nu veel meer werk nodig voor het aanpassen van WAQPRE. Verder is de hier beschreven aanpak niet uitbreidbaar naar DDHOR. DDVERT en DDHOR blijven hiermee in de beleving van de gebruiker duidelijk aparte werkwijzen vragen. Het enige voordeel is dat de run-procedure de configuratiefile niet meer hoeft te parsen; het selecteren van de goede informatie per deeldomein wordt nu door WAQPRE zelf gedaan.

2.7 Werkplan

Uitwerking van de hierboven beschreven aanpak op basis van een configuratiefile vergt vooral aanpassingen aan IPW voor het ondersteunen van de configuratiefile en van meerdere domeinen. Verder zijn er kleine aanpassingen nodig aan de run-procedures (minder opties, parsen configuratiefile), COPPRE (algemene i.p.v. eigen configuratiefile), COEXEC (uitgebreidere configuratiefile), en aan de documentatie.

De aanpassingen kunnen stapsgewijs worden ingevoerd. Daarbij stellen wij de volgende ordening van stappen voor:

1. Vaststellen van het format van de configuratiefile en uitproberen van de afhandeling hiervan binnen COEXEC. Invoeren van de configuratiefile in de runprocedures `waqpre.run`, `waqpro.run` en in partitioner COPPRE, in plaats van de nu gebruikte opties. Aanpassen van de gebruikersdocumentatie voor domein decompositie berekeningen.

Na deze aanpassingen kan IPW overweg met `siminp`-files die voor DDVERT gebruikt worden, hoewel de specifieke DDVERT modellering (verschillende aantallen lagen, laagverdelingen, AREAS-definitie) nog niet via IPW is te doen.

2. Uitbreiden van COPPRE en/of van `waqpre.run` voor hiërarchisch partitioneren. Dit moet het mogelijk maken dat in COPPRE de DDVERT-opdeling apart wordt ingelezen van partitionering per DDVERT deeldomein voor parallel rekenen.

Dit is nodig indien de configuratiefile de voorgestelde hiërarchische opbouw krijgt. Los daarvan is het ook prettig voor de gebruiker als hij de opdeling van DDVERT-deeldomeinen voor parallel rekenen met de standaard methodes kan laten doen.

3. Aanpassen van IPW en bijbehorende documentatie voor DDVERT: het kunnen inlezen en wegschrijven van een configuratiefile voor een som met alleen verticale verfijning. IPW moet hiervoor intern het aantal deeldomeinen kennen, en de laagafhankelijke informatie moet per deeldomein afzonderlijk kunnen worden opgegeven.
4. Uitbreiden van IPW voor het lezen en editen van de configuratiefile voor DDHOR, en voor het naast elkaar visualiseren van verschillende DDHOR-domeinen: bodemligging, ligging van open en gesloten randen, van overlaten, e.d.
5. Uitbreiden van IPW met de functionaliteit van VISIPART, om binnen IPW de opdeling van domeinen te kunnen veranderen.

6. Uitbreiden van IPW met mogelijkheden om de consistentie van verschillende (deel-) domeinen te controleren of af te dwingen.
7. Verder denkbare ontwikkelingen, zoals bijvoorbeeld een grafische schil rondom de configuratiefile, of introductie van XML en een conversieprogramma van XML naar **siprep**-formaat.

Hoofdstuk 3

Werkplan met betrekking tot nieuwe koppelingen

3.1 Inleiding

Op dit moment is domein decompositie met verticale en horizontale verfijning feitelijk gerealiseerd. Een belangrijk voordeel van deze ontwikkelingen is dat de hoeveelheid rekenwerk voor bepaalde modellen sterk kan worden gereduceerd. Immers: de roosterfijnheid kan per deeldomein zo laag mogelijk gekozen worden, in plaats van de resolutie in het hele domein bepaald te laten worden door het gebied dat de hoogste eisen stelt. Ook is de flexibiliteit vergroot. Bijvoorbeeld zorgt het gelijktijdig kunnen gebruiken van horizontale en verticale verfijning ervoor dat het 3D-Kuststrook model kan worden gekoppeld aan het horizontaal fijnere Zeedelta (DDVERT model met 1 laag voor de rivieren) waarbij de 3D-effecten ten gevolge van de zoet-zout overgangszone kunnen worden opgelost, terwijl er toch overlaten kunnen worden gebruikt in de rivieren in het Zeedelta-model. Merk op dat overlaten alleen in 2D-berekeningen beschikbaar zijn.

De mogelijkheden van domein decompositie kunnen op verschillende manieren worden uitgebouwd. De performance-winst kan verder worden opgevoerd door in bepaalde domeinen een deel van de procesmodellen uit te schakelen (denk aan transport of het $k - \epsilon$ turbulentiemodel) indien het doorrekenen van de processen daar niet *per sé* nodig is. Performance-winst kan ook worden behaald door de tijdstap per domein apart te kunnen kiezen, waarbij grove domeinen doorgaans met een grotere tijdstap kunnen worden doorgerekend dan fijnere domeinen. Voorts zou men voor gebieden die met 1 laag kunnen worden doorgerekend kunnen volstaan met een WAQUA som in plaats van een (tragere) TRIWAQ som. Dit laatste is een voorbeeld van een algemenere wens om verschillende modellen aan elkaar te kunnen koppelen (denk aan: WAQUA koppelen aan SIMPAR of SOBEK, of TRIWAQ aan een willekeurig model van een ander instituut).

De flexibiliteit voor gebruikers kan ook op verschillende manieren worden vergroot. Bijvoorbeeld is het voor sommige (DDHOR-)koppelingen wenselijk om roosters in bolcoördinaten te kunnen koppelen aan roosters in kromlijnige coördinaten (CSM8 \leftrightarrow Zuidelijke Noordzee). Een andere wens is om bij het maken van schematisaties roosterlijnen in de x -richting te kunnen koppelen aan roosterlijnen in de y -richting. Verder kunnen de mogelijkheden ten aanzien van het gebruik van overlaten en dynamische barriers verder worden uitgebreid.

Dit hoofdstuk geeft een beknopt werkplan voor het realiseren van deze nieuwe vormen van koppeling. Deze kunnen vanuit het oogpunt van de gebruiker in drie categoriën worden verdeeld, met oplopende graad van moeilijkheid.

1. Uitbreidingen van de huidige domein decompositie functionaliteit. Hieronder vallen:
 - Koppeling van ongelijksoortige roosters (bolcoördinaten aan kromlijnige coördinaten).
 - Koppeling van stukken van eenzelfde rooster die niet naast elkaar liggen.
 - Dynamische barrieresturing over meerdere DDHOR-domeinen.
 - Verbeterde ondersteuning voor overlaten op DDHOR-interfaces.
 - Performanceverbetering door andere koppelingsalgoritmes te gebruiken.
2. Per domein aan- of uitzetten van verschillende procesmodellen. Hieronder vallen:
 - Aan- en uitzetten van procesmodellen die per roosterpunt berekend worden zonder dat waarden uit buurpunten nodig zijn (k-Nikuradse, SVWP).
 - Per domein aan- of uitzetten van het transportmodel.
 - Per domein aan- of uitzetten van het $k - \epsilon$ turbulentiemodel.
3. Koppelen van rekenprocessen met verschillende tijdstappen, en andere koppelingen op tijdstapbasis. Dit omvat:
 - Koppelen van domeinen met verschillende tijdstappen.
 - Koppelen van x -roosterlijnen aan y -roosterlijnen.
 - WAQUA koppelen aan TRIWAQ.
 - WAQUA/TRIWAQ koppelen aan SIMPAR, SOBEK, Delft3D-MOR, of POM.

3.2 Indeling van dit hoofdstuk

In de volgende paragraaf komen eerst de uitbreidingen van de huidige domein decompositie functionaliteit aan de orde. Deze kunnen grotendeels al binnen de huidige programmatuur worden gerealiseerd.

Daarna zal besproken worden hoe procesvariëaties zoals het aan/uitschakelen van transport- en turbulentiemodellen kunnen worden gerealiseerd. Ook dit is met een relatief beperkte inspanning te realiseren. Tot op dit punt kunnen de wensen nog goed binnen de huidige structuren worden vervuld.

Daarna zal een beeld geschetst worden van hoe er in het algemeen met ongelijke rekenprocessen kan worden omgegaan. Dit vergt grotere ingrepen in de programmatuur, en tot op zekere hoogte ook het ontwikkelen van nieuwe numerieke methoden. Daarom is hiermee waarschijnlijk de grootste inspanning en ook doorlooptijd gemoeid.

Tenslotte zal op hoofdlijnen een werkplan worden opgesteld, waarin een mogelijke volgorde voor de verschillende stappen wordt aangegeven.

3.3 Uitbreidingen van de huidige domein decompositie functionaliteit

In deze paragraaf worden een aantal uitbreidingen besproken van de huidige domein decompositie functionaliteit.

3.3.1 Koppeling van kromlijnige roosters aan bolcoördinaten

In de huidige programmatuur wordt op verschillende plaatsen een conversie uitgevoerd tussen kromlijnige coördinaten en bolcoördinaten. De precieze formules die hiervoor kunnen worden toegepast zijn bekend bij de Meetkundige Dienst en worden in een generieke transformatietool geïmplementeerd. Op basis van deze omrekenformules is het zeer eenvoudig om de koppeling van krom- aan bolcoördinaten toe te staan.

3.3.2 Koppeling van verschillende stukken van hetzelfde domein

Een mogelijkheid van de geïmplementeerde vormen van domein decompositie is om onnatuurlijke knikken en oprekking in de gebruikte rekenroosters te vermijden. Zulke in principe ongewenste features zijn soms nodig voor het maken van een geheel rooster dat gebruikt kan worden in sequentiële simulaties (d.i. zonder domein decompositie). Bijvoorbeeld om stukken van roosters aan elkaar te verbinden die in werkelijkheid dicht bij elkaar liggen, maar in termen van (m, n) -coördinaten niet.

Een nadeel van de huidige domein decompositie programmatuur voor deze toepassingen is dat de gebruiker aparte siminp-files moet maken voor de verschillende gedeeltes van het rooster. Bijvoorbeeld voor ieder verbindingskanaaltje apart, of een enkele siminp-file met daarin meerdere van zulke kanaaltjes. Het zou veel mooier zijn als alle verbindingskanaaltjes in een enkele siminp-file konden worden opgenomen.

Deze aanpassing vereist dat het mogelijk wordt gemaakt dat verschillende stukken van hetzelfde domein (= siminp-file) aan elkaar gekoppeld kunnen worden. Dit vergt aanpassingen aan het matchingsgedeelte voor DDHOR-berekeningen (ook matches toestaan tussen “koppelbare randen” van hetzelfde domein) en aan de communicatiebibliotheek (geen PVM-boodschappen versturen/ontvangen in geval buurman = zelf). Verder verandert er het een en ander aan de voorwaarden waarmee wordt herkend of er bijvoorbeeld wel/geen DDHOR-communicatie nodig is. Nu is zo’n voorwaarde dat het aantal domeinen (siminp-files) gelijk is aan 1, hiervoor moet een geschikt alternatief worden geïmplementeerd.

3.3.3 Dynamische barriersturing over meerdere domeinen

Het is gewenst om bij dynamische barriersturing voor het sturen van een barrier in het ene domein gebruik te kunnen maken van condities in een ander domein. Technisch gesproken is dit niet zo heel moeilijk, maar er is wel wat werk mee gemoeid.

Uitgangspunt is dat de siminp-files per domein onafhankelijk van elkaar moeten worden behandeld door WAQPRE. Daarbij moet de structuur van de siminp-file zodanig worden uitgebreid dat er kan worden verwezen naar externe condities. Die condities volgen dan uit

de siminp-files van de andere domeinen en uit de configuratiefile voor de gekoppelde berekening (zie werkplan pre- en postprocessing). Tijdens de initialisatiefase van de gekoppelde berekening wordt alle aangeleverde informatie aan elkaar gepast.

Het mechanisme dat wij voorstellen voor het opgeven van de externe referenties is als volgt. In plaats van de nummers van de te gebruiken punten of curves voor de barriersturing zouden we namen willen gebruiken.

```
CONDITION
  IF ( CO 1 P 1 LAYER 1 LT 1000. ) THEN
    TS1
  ELSEIF ( LEVEL 'Hoek v Holland 1' LT 25.0 ) THEN
    TS2
  ENDIF
```

Hierin verwijst P 1 naar een locatie binnen het eigen domein, en is 'Hoek v Holland 1' een externe referentie. WAQPRE moet worden uitgebreid om deze externe referenties te herkennen en te bewaren.

De namen kunnen in de configuratiefile worden overschreven zoals in Hoofdstuk 2 ook voor andere informatie is voorgesteld:

```
DOMAIN 3
  NAME = 'kust'
  LET 'Hoek v Holland 1' BE 'Hoek van Holland'
```

In de initialisatiefase van een gekoppelde berekening wordt er uitgezocht welke andere domeinen de benodigde informatie moeten toeleveren. Daarbij moeten ook de nodige controles worden ingebouwd. Tenslotte veranderen de communicaties die voor de barriersturing worden gebruikt enigszins.

3.3.4 Ondersteuning voor overlaten op DDHOR-interfaces

In het project DDHOR+VERT zijn complicaties ontdekt met overlaten in DDHOR-berekeningen. De moeilijkheid is dat interpolaties in DDHOR min of meer uitgaan van *continuïteit* van de oplossing, terwijl het overlatenmodel juist een *discontinue* aansluiting van de stroomtoestanden links en rechts van een overlaat mogelijk maakt. In de overlaatroutines worden nu in bepaalde omstandigheden twijfelachtige waarden gebruikt voor de waterstanden aan weerszijden van overlaten en voor het debiet over een overlaat. Dit leidt tot een verkeerde inschatting van het energieverlies over de overlaat, vooral in niet-stationaire omstandigheden.

Er zijn twee aanpassingen nodig om overlaten op DDHOR-interfaces beter te ondersteunen:

1. verbieden van diagonale overlaten waarbij het u -punt bij een ander domein hoort dan het v -punt van de overlaat. Zulke gecombineerde overlaten leveren allerlei moeilijkheden op, zowel wiskundig als in de programmatuur;
2. gebruiken van *samecell* interpolatie in plaats van *bilineaire* interpolatie voor waterstandpunten in de guard band direct naast een overlaat op de DDHOR-interface. Een eigenschap van de samecell-methode is namelijk dat hij alleen berekende waarden uit het buurdomein gebruikt. Hiermee wordt het gebruik van waterstanden aan weerszijden van de interface omzeild.

De eerste aanpassing hangt samen met een moeilijkheid in het genereren van goede invoer voor DDHOR-berekeningen. Als de echte ligging van overlaten voor ieder domein apart wordt afgebeeld op het rooster, dan is het mogelijk dat er op DDHOR-interfaces gaten ontstaan.

3.3.5 Performanceverbeteringen

Naast de hierboven besproken functionele uitbreidingen zou het gebruik van de huidige domein decompositie functionaliteit ook kunnen worden vergemakkelijkt door de performance te verbeteren. Dit kan worden bereikt door aanpassing van de numerieke koppelingsalgoritmen.

Ten eerste kan het aantal iteraties voor de continuïteitsvergelijking worden verminderd door slimmere koppelingsvergelijkingen te gebruiken (bijv. op basis van Riemann-invarianten), of door een slimmer stopcriterium te implementeren. Dit heeft direct effect, omdat de continuïteitsvergelijking een van de meest rekenintensieve stappen van WAQUA/TRIWAQ is.

Verder kan de zogenaamde “Wang-solver” worden geïmplementeerd in plaats van de huidige block-Jacobi koppeling voor de continuïteitsvergelijking. Wang’s algoritme is sterk gerelateerd aan de Schur-complement methode en is een directe solver voor de gekoppelde vergelijkingen van de verschillende domeinen. Met deze methode wordt de convergentie voor de iteratieve berekening van waterstanden verbeterd en kunnen eventuele tijdstaprestricties ten gevolge van domein decompositie koppelingen worden omzeild. In het kader van het OMS-project is besproken dat de Wang-solver binnen de huidige architectuur goed in een apart rekenproces kan worden geïmplementeerd.

Tenslotte kan de huidige koppelingsmethode worden versneld door niet alle deeldomeinen gelijktijdig een iteratie uit te laten voeren, maar door dit meer sequentiëel te laten doen. In de theorie van iteratieve procedures komt dit overeen met de Jacobi en Gauss-Seidel methodes, ookwel resp. “simultaneous displacements” en “successive displacements” genoemd, waarbij de laatste vaak twee keer zo snel convergeert als de eerste. In de theorie van domein decompositie worden de twee methodes respectievelijk “additive Schwarz” en “multiplicative Schwarz” genoemd.

Om een multiplicative Schwarz methode te kunnen gebruiken moet de communicatiebibliotheek worden uitgebreid. In plaats dat in een update-operatie alle rekenprocessen tegelijkertijd randwaardes aan elkaar versturen moet nu steeds slechts een van de processen zijn waardes versturen. Verder moet de implementatie van de iteratieve solvers in WAQUA/TRIWAQ worden aangepast om de nieuwe update-operatie te gebruiken en om de gewenste volgorde van iteraties van verschillende deeldomeinen te bereiken.

Deze verschillende methoden om de koppelingsalgoritmen te verbeteren zijn vooral in combinatie met andere ontwikkelingen van belang. Bijvoorbeeld valt te verwachten dat de gebruiker steeds vaker verschillende gedeeltes van het te simuleren gebied in aparte deelroosters zal gaan onderbrengen (zie ook paragraaf 3.3.2), en dat de interfaces tussen verschillende deelroosters steeds langer zullen worden. Dit laatste speelt bijvoorbeeld wanneer het zomer- en winterbed van een rivier apart worden beschreven met gebruik van horizontale verfijning. Dit leidt tot het op steeds meer lokaties gebruiken van de domein decompositie koppeling, dus moet deze niet onnodig veel extra iteraties introduceren.

Verder moet de communicatie van randwaardes efficiënt worden geïmplementeerd. In een aantal gevallen zou je het gebruik van PVM willen omzeilen, en in plaats daarvan een “*in memory*” mechanisme willen gebruiken binnen een enkel rekenproces. Dit zou goed kunnen

worden gebruikt binnen een opzet van WAQUA/TRIWAQ (of OMS-FLOW) waarin er per rekenproces meerdere aparte deelroosters worden doorgerekend (multi-block architectuur).

3.4 Per domein aan- of uitzetten van verschillende procesmodellen

In deze paragraaf worden verschillende mogelijkheden besproken om bepaalde procesmodellen per domein aan- of uit te zetten, of per domein te variëren. Dit kan enerzijds een forse besparing opleveren in de rekestijd, en anderzijds de mogelijkheden voor de gebruiker vergroten.

3.4.1 Aan/uitzetten van procesmodellen die per roosterpunt werken

Procesmodellen die voor alle roosterpunten onafhankelijk van elkaar worden uitgerekend kunnen gemakkelijk worden gevarieerd per domein. Dit geldt bijvoorbeeld al voor de ruwheidsformulering. Met de introductie van DDVERT is het mogelijk geworden om in het ene deeldomein Chezy en in het andere White-Colebrook te gebruiken.

Hetzelfde geldt voor “space varying wind and pressure” (SVWP) en DDHOR. De programmatuur staat toe dat voor het ene domein een enkele tijdserie wordt opgegeven voor windsnelheid en -richting, terwijl in het andere domein SVWP wordt gebruikt. Wel is het zo dat zulke variaties in de procesmodellen per deeldomein kunnen leiden tot sprongen in de oplossing over de deeldomeinen. Immers: links en rechts van een interface wordt een andere bodemruwheid of een andere luchtdruk gebruikt. Het is aan de modelleur om rekening te houden met deze sprongen bij het interpreteren van resultaten, en om de aansluiting zo mooi mogelijk te maken.

Soms lijkt een procesmodel strikt lokaal te zijn, maar wordt er bij precieze beschouwing toch informatie van naburige roosterpunten gebruikt. Dit was voorheen zo voor de berekeningen voor de k -Nikuradse ruwheidsformulering. Daar werden de resulterende waarden van de White-Colebrook coëfficiënt gecommuniceerd tussen naburige deeldomeinen om communicatie elders in het programma uit te sparen. Door deze communicatie was het niet meer mogelijk om in het deeldomein wel en in het andere geen k -Nikuradse berekening te gebruiken. Door aanpassingen aan de rekenroutines in het project DDHOR+VERT, het verwijderen van de zogenaamde “redundante berekening”, is de communicatie van de k -Nikuradse berekening komen te vervallen. Het is hierdoor nu wel mogelijk geworden om de deze berekening per domein apart te activeren.

3.4.2 Per domein aan/uitzetten van het $k - \epsilon$ turbulentiemodel

Het per domein kunnen aan- of uitzetten van het $k - \epsilon$ turbulentiemodel is onder andere interessant in combinatie met DDVERT. Bijvoorbeeld wordt de turbulentieberekening momenteel ook uitgevoerd in deeldomeinen waar er slechts 1 laag wordt gebruikt, terwijl dat eigenlijk betekenisloos is. Maar dit is nu nodig omdat er nog geen variatie is toegestaan.

Het per domein aan/uitzetten van het turbulentiemodel is ingrijpender dan de in de vorige paragraaf besproken variaties. Dit komt doordat er in het turbulentiemodel ruimtelijke dis-

cretisaties worden gebruikt, waardoor er speciale maatregelen moeten worden getroffen op de interfaces tussen domeinen. Er moet daar dan ook randafhandeling worden geïmplementeerd.

In de huidige situatie levert ieder buurdomein de waardes die nodig zijn voor de randafhandeling voor de interfaces tussen de domeinen. Dat zou kunnen worden voortgezet door per domein zonder turbulentiemodel een eenvoudig constant vertikaal profiel voor k en ϵ te hanteren. In het algemeen is dit echter geen goed uitgangspunt: het is niet voor alle procesmodelvarianties mogelijk om een geschikte randvoorwaarde te betrekken uit een buurdomein dat met een heel ander procesmodel werkt. Bovendien komt dit de flexibiliteit niet ten goede: je wilt niet ieder model opzadelen met kennis van alle grootheden die in de gekoppelde modellen worden gebruikt.

Daarom moet het uitgangspunt zijn dat een model zelf de randvoorwaarden moet verzorgen op interfaces waar het buurdomein geen informatie kan leveren. Zaken die daarbij aan de orde komen zijn:

1. Hoe komt een domein te weten dat het buurdomein geen turbulentie gebruikt.
2. Welke randvoorwaarden worden er gehanteerd op interfaces tussen domeinen met en zonder turbulentiemodel.
3. Hoe worden in de code de randvoorwaarden aangepast als de buurman geen turbulentie kent.
4. Hoe worden de communicaties in het turbulentiemodel ingeperkt.

Ad 1. Bepalen of buurdomeinen wel of geen turbulentie gebruiken

Het bepalen of (deel)domeinen wel of geen turbulentieberekening uitvoeren kan op dezelfde manier gebeuren waarop ze nu van elkaar het aantal lagen en de laagverdelingen te weten komen: met behulp van globale communicatie.

Ad 2. Welke randvoorwaarden moeten worden gebruikt

Het ligt voor de hand om randvoorwaarden op te laten geven via de bestaande mechanismes in de programmatuur voor open randen. Voor het turbulentiemodel is dit een soort Neumann-randvoorwaarde: $\partial k / \partial x = \dots = \partial \epsilon / \partial y = 0$.

Ad 3. Corrigeren van het randtype

Er zijn in principe twee plekken waar de randtypes die in de programmatuur gehanteerd worden kunnen worden aangepast: in COPPRE en in WAQPRO. In COPPRE is echter nog niet bekend welke buurdomeinen wel of geen turbulentie berekenen. De correctie van de randtypes zal daarom in WAQPRO moeten worden ondergebracht.

Er zal dan bepaald moeten worden welke randpunten grenzen aan domeinen waarin geen turbulentie berekend wordt en voor die punten moet de randcode aangepast worden, alsmede de randvoorwaarde. Nadeel van deze methode is wel dat eventuele problemen met de opgegeven randvoorwaarden pas laat worden gedetecteerd. Daarom is het wenselijk om ook in de module die de consistentie van domeinen bepaalt te controleren of er op alle betreffende subdomeinranden tijdseries zijn gedefiniëerd.

Ad 4. Inperken communicaties

Bij het opstellen van stencils voor de communicatiebibliotheek wordt er van uitgegaan dat alle deeldomeinen dezelfde procesmodellen gebruiken. Dat betekent dat er bij communicatie in het turbulentiegedeelte ook informatie verwacht wordt van deeldomeinen die geen turbulentie berekenen.

Er zijn een aantal manieren denkbaar om hiermee om te gaan. In de eerste plaats zou men aparte stencils voor het turbulentiegedeelte kunnen maken, waarbij maskers worden gebruikt om communicatie met deeldomeinen die geen turbulentie gebruiken uit te sluiten. Dit is een niet erg elegante oplossing: er zouden meer stencils moeten worden gecreëerd dan strikt noodzakelijk is, en die krijgen dan namen als `stcmax_turbulence`.

Een betere oplossing is om het concept van procesgroepen te gaan gebruiken. Dit wordt nu in globale communicaties gebruikt voor min of meer hetzelfde doel. Bijvoorbeeld kunnen alle deeldomeinen die iets met de barriëresturing te maken hebben met elkaar communiceren, zonder dat de overige processen hiervan af hoeven te weten. Deze procesgroepen zouden ook in de update-operatie kunnen worden ingevoerd.

Een laatste oplossing is om gebruik te maken van het koppelingsalgoritme zoals dat is ontwikkeld voor parallel Kalman. Daarbij wordt in een externe file gespecificeerd welke communicaties een proces allemaal uitvoert en in welke volgorde. Daarbij wordt dan ook vastgelegd welke communicaties uit het ene proces zijn verbonden met welke communicaties uit het andere proces. Het is denkbaar dat in de specificatie van communicaties per proces bepaalde stukken conditioneel worden gemaakt, waarbij de conditie op runtime bepaald wordt. De communicatiebibliotheek (die de inhoud van het koppelingsalgoritme kent) kan dan op runtime bepalen of een buurproces een communicatie zal uitvoeren of niet.

Het tweede alternatief lijkt geschikt te zijn voor het huidige doeleinde en is redelijk elegant. Voor andere toekomstige uitbreidingen is echter toch ook het koppelingsalgoritme van parallel Kalman nodig. Keuze voor de derde mogelijkheid zou dus prettig zijn, omdat hiermee de aanpak voor parallel Kalman en die voor domein decompositie bij elkaar worden gebracht. Bovendien sluit deze aanpak goed aan bij de ideeën die binnen OMS ontwikkeld worden.

3.4.3 Per domein aan/uitzetten van het transportmodel

Voor het aan- en uitzetten van het transportmodel gelden in grote lijnen dezelfde overwegingen als zojuist genoemd voor het aan/uitzetten van het turbulentiemodel. Alleen het opgeven van randvoorwaarden voor interfaces van domeinen waar het buurdomein geen transportberekening doet (ad 2. in paragraaf 3.4.2) is moeilijker. Dit is omdat er voor het transportmodel, in tegenstelling tot het turbulentiemodel, wel randwaardes worden opgegeven in de invoerfile.

Het ligt wederom voor de hand om randvoorwaarden op te laten geven via de bestaande mechanismes in de programmatuur voor open randen. Voor het transportmodel is dit ongestoord transport bij uitstroming en de Thatcher-Harleman formulering met constituent-return time bij instroom. De vraag doet zich dan voor hoe de gebruiker deze extra randvoorwaarde specificeert. WAQPRE staat alleen randvoorwaarden toe op de enclosure, terwijl er nu ook randvoorwaarden nodig kunnen zijn in het binnengebied, op de rand van het actieve gedeelte van het rooster.

In eerste instantie kan de mogelijkheid van aan/uitschakelen van transportberekening beperkt worden tot situaties waarin er op openingen wordt gekoppeld. Dit betekent dat een

model met transportberekening in zijn geheel meedoet in de DDHOR-berekening, of dat aan de kant waar er een stuk van wordt afgesneden het buurdomein ook transportberekening heeft. Voor de koppeling van Kuststrook aan Zeedelta betekent dit dat de transportberekening van Kuststrook kan worden uitgeschakeld als het gehele zeegebied van Zeedelta wordt gebruikt.

Pas zodra het bovenstaande mechanisme is geïmplementeerd komt het aan de orde om extra randvoorwaarden te kunnen opgeven in het binnengebied van een domein. Hiervoor zijn er ruwweg twee mogelijkheden: de extra tijdseries kunnen worden opgenomen in de siminp-file of in de configuratiefile zoals in Hoofdstuk 2 is beschreven. Het eerste is ongewenst omdat er dan in de siminp-file tijdseries terechtkomen die alleen betekenis hebben voor een speciale roosteropdeling en procesmodelvariatie. Het tweede is evenmin gewenst omdat er dan gedeeltes van de referentietabel van WAQUA worden herhaald in de configuratiefile. Een mogelijke middenweg is dat in de configuratiefile de naam wordt opgegeven van het bestand met de extra tijdseries, en dat dit op de een of andere manier wordt ingevoegd aan de tijdseries in het siminp-bestand.

3.4.4 Variëren van het aantal getransporteerde stoffen

Een verfijning van de hierboven beschreven uitbreiding is om niet zozeer alle domeinen ofwel met ofwel zonder transport door te rekenen, maar om per domein het aantal constituents te kunnen variëren dat wordt doorgerekend. Bijvoorbeeld: in alle domeinen wordt het transport van zware metalen doorgerekend, maar alleen bij de kust wordt gerekend met zouttransport.

Deze procesvariatie leidt tot grote moeilijkheden wanneer wordt toegestaan dat de volgorde van de constituents verschilt per domein. Dan wil bijvoorbeeld het ene domein eerst zout doorrekenen (en daarbij communiceren met de buurdomeinen met zout), terwijl de ander eerst de metalen wil doen. Voor de gebruiker is het veel minder belastend om de volgorde van de gebruikte constituents gelijk te maken.

Het variëren van het aantal constituents kan in principe langs dezelfde lijn aangepakt worden als het in zijn geheel aan/uitschakelen van transport, maar levert wel extra complicaties. Met name moeten de randtypes per constituent aangepast kunnen worden: voor sommige constituents is een opening een echte rand waar speciale randvoorwaarden worden opgegeven, voor andere stoffen is het een subdomeinrand waarvoor er wordt gecommuniceerd. Daarnaast zullen er ook per constituent procesgroepen gedefinieerd moeten worden.

Het is aan te bevelen om deze optie pas te implementeren als het aan/uitzetten van transport als geheel al is geïmplementeerd. Maar bij het ontwerpen daarvan zouden mogelijke variaties in het aantal constituents al wel meegewogen moeten worden.

3.5 Koppelen van ongelijksoortige rekenprocessen

Tot nu toe zijn vormen van koppeling aan de orde geweest waarbij de rekenprocessen nog veel op elkaar lijken en hooguit een deel van de algoritmiëk van het rekenproces aan- of uitgeschakeld kan zijn. In deze paragraaf wordt ingegaan op koppelingen waarbij de berekeningen in de verschillende gekoppelde processen niet meer gelijk met elkaar oplopen, maar waarbij de verschillende modellen binnen een tijdstap verschillende berekeningen uitvoeren.

In het geval van domein decompositie gaat dit over bijvoorbeeld koppeling van WAQUA/-TRIWAQ aan een ander waterbewegingsmodel (SOBEK, POM, WAQUA aan TRIWAQ), of

over de koppeling van domeinen die een verschillende tijdstap gebruiken, of over de koppeling van x - aan y -roosterlijnen. In dit laatste geval is er van ongelijksoortige berekeningen per tijdstap sprake omdat er in WAQUA/TRIWAQ een ADI-schema wordt gebruikt. Maar de problematiek van deze vormen van domein decompositie is sterk gelijk aan de problematiek van verschillende soorten van proces-koppeling: WAQUA \leftrightarrow Delft3D-MOR, WAQUA \leftrightarrow SIMPAR, etc.

Dit soort koppelingen is al uitgebreid onderzocht bij het realiseren van een parallelle versie van het Kalman filter en krijgt ook veel aandacht in het Open Model Systeem (OMS) project. De voorstellen die hier gedaan worden zijn in lijn met die ontwikkelingen.

3.5.1 Algemene aanpak

Bij het realiseren van verschillende vormen van proces-koppeling spelen verschillende aspecten een rol:

1. Er moet een numerieke methode worden verzonnen om de deelmodellen aan elkaar te koppelen,
2. de methode moet worden geïmplementeerd in de verschillende modules voor de verschillende deelmodellen, en
3. deze aparte modules moeten zodanig worden opgezet dat ze flexibel inzetbaar zijn in verschillende configuraties.

De laatste twee punten zijn een nadrukkelijk onderdeel van onze aanpak voor de parallelle versie van de Kalman-filtering programmatuur en van de software architectuur van OMS. Dit wordt enigszins uitgewerkt in de volgende paragraaf. In dit werkplan gaan we verder vooral in op het eerste punt, consequenties voor WAQUA/TRIWAQ van numerieke koppelingen op tijdstapbasis.

3.5.2 De Parallel Kalman aanpak

In veel gevallen leidt de koppeling van minder op elkaar gelijkende rekenprocessen ertoe dat de numerieke berekeningen binnen het ene proces niet meer samenvallen met die in een aangrenzend proces. Men zal dus voor specifieke koppelingen bepaalde communicaties aan- of uit moeten zetten: binnen een parallelle of domein decompositie run wordt bijvoorbeeld gewoon op sub-iteratieniveau gecommuniceerd, bij een koppeling naar een ander pakket (bijvoorbeeld WAQUA ten opzichte van SOBEK) wordt op een ander niveau gecommuniceerd.

Om te voorkomen dat men in de code elke koppeling expliciet moet uitprogrammeren (“als ik aan WAQUA gekoppeld ben, dan moet ik hier communiceren”), is bij het ontwikkelen van een parallelle versie van het Kalman filter het idee van een configuratiefile ontstaan. Dit idee is in OMS overgenomen.

Het idee is dat een code uiteindelijk een aantal specifieke communicatie-calls bevat. Zo staat er bijvoorbeeld aan het einde van elke tijdstap een communicatieroutine waarmee de toestand in de buurt van de randen wordt weggestuurd naar wie er maar in geïnteresseerd is. Vervolgens staat er een communicatieroutine die de toestand net over de randen ontvangt van een willekeurig ander proces, indien dat aanwezig is. Of deze communicatieroutines

daadwerkelijk iets verzenden of ontvangen, wordt extern geconfigureerd. In een configuratiefile wordt aangegeven dat het wegsturen en ontvangen met het ene proces niet hoeft te gebeuren, maar met een ander proces wel.

De verwachting is dat er in de code uiteindelijk een beperkt aantal aanroepen van communicatieroutines komt te staan, waarmee alle in praktijk voorkomende koppelingen geconstrueerd kunnen worden.

3.5.3 Koppeling op tijdstapbasis met gelijke tijdstappen, 1 : 1

In de huidige implementaties van parallel rekenen en van domein decompositie in WAQUA/TRIWAQ wordt er binnen ieder deeldomein dezelfde reeks van stappen doorlopen: per tijdstap oplossen van de u -impulsvergelijking, oplossen van de continuïteitsvergelijking en v -impulsvergelijking, etc. Binnen ieder van die tijdstappen wordt er gecommuniceerd.

Deze intense koppeling is niet te handhaven wanneer WAQUA/TRIWAQ wordt gekoppeld aan een ander soort waterbewegingsmodel. Bijvoorbeeld gebruikt SOBEK niet dezelfde onderverdeling van iedere tijdstap in deelstapjes. Dus als men WAQUA aan SOBEK zou willen koppelen, dan moet er een nieuwe numerieke methode voor de koppeling worden verzonnen. Wat hiervoor voor de hand ligt is om gelijke tijdstappen te gebruiken in beide deelmodellen, om randwaardes aan het begin en einde van iedere tijdstap met elkaar uit te wisselen, en om een extra iteratieproces rondom iedere tijdstap te plaatsen, waarmee wordt geïtereerd tot over en weer consistentie bereikt is. Meer algemeen: de modellen worden op tijdstapniveau aan elkaar gekoppeld.

Bij tijdstapkoppeling is het belangrijkste issue dat een domein in staat moet zijn om een tijdstap te herhalen omdat er op tijdstapniveau een iteratieproces nodig is. Concreet betekent dit dat er bij zo'n rollback allerlei zaken teruggezet moeten worden in een toestand van een tijdstip geleden (met name randvoorwaarden en de uitgangstoestand).

Dit is voor WAQUA/TRIWAQ tot op zekere hoogte geïmplementeerd binnen de KALMINA-programmatuur voor data-assimilatie. In het RRSQRT Kalman filter is het namelijk nodig om naast de tijdstap voor de modeltoestand zelf, ook tijdstappen uit te voeren voor de zgn. error-modes. Dit zijn verstoringen van de modeltoestand die de onzekerheid van de voorspelling beschrijven. Een verschil met domein decompositie is echter wel dat er in het Kalman filter geen iteratie over de tijdstap wordt uitgevoerd. Het iteratieproces op tijdstapniveau voor domein decompositie stelt extra eisen aan de rollback. Met name wil men zo veel mogelijk informatie van vorige buiten-iteraties hergebruiken om iteratieprocessen binnen de tijdstap te versnellen.

3.5.4 Koppeling op tijdstapbasis met ongelijke tijdstappen, 1 : n

Bij ongelijke tijdstappen per domein werkt het buiten-iteratieproces in termen van het domein met de grootste tijdstap. Als daarin een tijdstap wordt herhaald, dan zullen de domeinen met fijnere tijdstap terugmoeten naar een grootste tijdstap geleden.

In dit geval moet nagedacht worden over de betekenis van de randvoorwaarden die uit het domein met de grotere tijdstap komen. Om die zinvol als randvoorwaarde te kunnen gebruiken, moeten deze in de tijd geïnterpoleerd worden naar het tijdstip waarop ze in het domein met een kleine tijdstap gebruikt worden. In dat kader is het denkbaar dat men binnen één tijdstapiteratie eerst het domein met grote tijdstap zal doorrekenen (met randvoorwaarden

uit het domein met kleine tijdstap uit de vorige iteratie) en daarna het domein met een kleine tijdstap met randvoorwaarden die zojuist binnen deze iteratie door het domein met grote tijdstap zijn berekend. Algemeen gesteld: er moet nagedacht worden over de volgorde waarin (delen van) tijdstappen worden doorgerekend om te proberen om zo snel mogelijk convergentie te bereiken. Dit is een onderzoek op zich.

De vraag doet zich daarbij voor waar de tijdsinterpolatie moet worden uitgevoerd. Het ligt voor de hand om dit in de communicatiebibliotheek onder te brengen, analoog aan de ruimtelijke interpolatie. Daarvoor zou analoog aan de roostermatching een tijdstapmatching moeten gebeuren. Indien dit volledig doorgezet wordt, kan de communicatiebibliotheek ook zelf in de gaten houden of er bepaalde waarden van de buurman verwacht worden of niet. Daarvoor is het dan wellicht nodig om het begrip tijdstap op te rekken om ook iteratieniveau's te omvatten.

Voor het realiseren van tijdstapkoppeling kan het beste een twee-fase aanpak gevolgd worden. In eerste instantie zou er een 1 : 1 tijdstapkoppeling gerealiseerd kunnen worden om issues op te lossen die samenhangen met het introduceren van een buiteniteratie (met name de rollbackvoorzieningen) en de daarmee verbonden administratie. Daarna zou een 1 : n tijdstapkoppeling kunnen worden gerealiseerd om interpolatieissues aan te pakken.

3.5.5 Koppeling tussen WAQUA en TRIWAQ

De koppeling tussen WAQUA en TRIWAQ is interessant omdat WAQUA momenteel een factor twee tot vier sneller rekent dan TRIWAQ, en omdat WAQUA extra functionaliteit bevat voor riviermodellen. Bijvoorbeeld onderkent het model voor barriers in WAQUA superkritische stroomtoestanden, en zijn er speciale randvoorwaarden voor rivieren in gedefinieerd (QH- en QAD-randen). Dus voor bijvoorbeeld het Zeedelta-model is het momenteel gewenst om de riviergedeeltes met WAQUA te kunnen simuleren.

De extra functionaliteit van WAQUA zou geen reden moeten zijn om een koppeling tussen WAQUA en TRIWAQ na te streven. In plaats daarvan zou de ontbrekende functionaliteit ook in TRIWAQ moeten worden geïmplementeerd voor 2D berekeningen. Dit zou sterk worden vereenvoudigd als WAQUA en TRIWAQ inwendig op elkaar worden afgestemd. Met name moet hiervoor het gebruik van indirecte adressering via de LGRID-tabel uit WAQUA worden verwijderd. Dit is de lijn die momenteel in het OMS-project wordt gevolgd.

De betere performance van WAQUA ten opzichte van TRIWAQ is wel een argument om de koppeling na te streven. Maar dit betekent wel dat de koppeling intensiever moet zijn dan op de hierboven beschreven tijdstapbasis. Het gebruiken van een buiten-iteratie op tijdstapniveau is namelijk nadelig voor de performance, en dit zal het performancevoordeel van het gebruik van WAQUA compleet overschaduwten. Om een strakkere koppeling te bereiken zal de algoritmieken van beide programma's veel op elkaar moeten worden afgestemd.

Een aantal zaken komt hierbij in elk geval aan de orde: conversie van `fullbox` naar `mnmaxk`-formaat, de verschillende stencils in WAQUA en TRIWAQ, de verschillende iteratieprocessen, en variaties in de procesmodellen.

Conversie van `fullbox` naar `mnmaxk`-formaat

Het eerste punt bij het koppelen van WAQUA met TRIWAQ is dat deze nu verschillende opslagstructuren gebruiken voor roosterfuncties zoals de waterstand: TRIWAQ gebruikt

zogenaamde `fullbox`-arrays, WAQUA gebruikt indirecte adressering naar het zogenaamde `mmaxk`-formaat. Dit levert geen grote moeilijkheden voor het realiseren van de koppeling. De uitgebreidere definitie van het begrip `indexset` die bij de parallelisatie van het Kalman filter is geïntroduceerd is hiervoor direct te gebruiken. Het ene programma levert dan zijn informatie op de ene manier, terwijl het andere programma het op de door hem gewenste andere manier ontvangt. De conversie vindt plaats binnen COCLIB en wordt met krachtige, generieke mechanismen geconfigureerd.

Verschillende stencils

Een tweede moeilijkheid is dat WAQUA en TRIWAQ in de communicaties voor verschillende groottheden verschillende stencils gebruiken. TRIWAQ communiceert bijvoorbeeld zoutconcentraties voor drie rijen en kolommen van het rooster en WAQUA slechts voor twee. Dit moet op elkaar worden afgestemd, en hiervoor moet enige vorm van overleg tussen WAQUA en TRIWAQ geprogrammeerd worden. Daarbij moet ieder proces aangeven waar het informatie van het andere proces nodig heeft. Ook dit is al grotendeels uitgezocht in het kader van de parallelisatie van het Kalman filter.

Verschillende iteratieprocedures

Een derde aspect is dat de iteratieprocedures in WAQUA en TRIWAQ van elkaar verschillen. Bijvoorbeeld gebruikt TRIWAQ voor het stelsel vergelijkingen voor het transportmodel een “red-black Jacobi methode”, terwijl WAQUA de zogenaamde BJ-TWGE methode gebruikt. Verschillen tussen deze methodes zijn dat er in TRIWAQ twee keer per iteratie wordt gecommuniceerd, eerst voor “rode” en dan voor “zwarte” roosterpunten, en in WAQUA slechts een keer, en dat het aantal iteraties in WAQUA per rij apart wordt bepaald terwijl dit in TRIWAQ voor alle rijen hetzelfde is. Koppeling van de twee modellen vereist daarom aanpassingen aan de algoritmieken: de communicatiepunten voor randinformatie, en de afstemming van stop-criteria en het aantal iteraties dat wordt uitgevoerd.

De verschillen in de iteratieprocedures hangen verder samen met verschillen in de droogvalmodellering. Koppeling van de huidige waterbewegingsmodules van WAQUA en TRIWAQ zou daarom aanzienlijke moeilijkheden met zich meebrengen. Om de koppeling te vergemakkelijken zouden WAQUA en TRIWAQ daarom eerst op dit punt op elkaar moeten worden afgestemd (zie ook [8]).

Procesmodelvarianties

Tenslotte vereist de koppeling van WAQUA aan TRIWAQ dat de voorzieningen voor procesvarianties uit paragraaf 3.4 worden geïmplementeerd. Het is namelijk nodig om het $k - \epsilon$ turbulentiemodel te kunnen beperken tot de domeinen waar met TRIWAQ wordt gerekend, aangezien deze module niet in WAQUA is geïmplementeerd.

3.5.6 Koppeling van x -roosterlijnen aan y -roosterlijnen

Bij presentaties over domein decompositie hebben verschillende gebruikers aangegeven behoefte te hebben aan de mogelijkheid om modellen met verschillende orientaties aan elkaar te kunnen koppelen. Dit is met name van belang voor het maken van kromlijnige roosters

rond eilanden of voor verbindingskanalen. Bedenk bijvoorbeeld een eiland in de vorm van een halve waterdruppel waar roosterlijnen omheen lopen; twee roosterlijnen die parallel aan elkaar lopen bij de punt, komen haaks op elkaar te staan aan de andere zijde.

Het aan elkaar koppelen van domeinen met verschillende orientaties is moeilijk vanwege het belang van “rekenrichting” in de numerieke methode, de ADI-aanpak. Een geschikte manier om met de verschillende orientaties om te gaan is nu om de rekenmodellen per domein als compleet verschillend van elkaar te beschouwen, en om te koppelen op tijdstapbasis.

Er kan dus aan de koppeling van x -roosterlijnen aan y -roosterlijnen worden gewerkt zodra 1 : 1 tijdstapkoppeling is gerealiseerd. Wel worden er extra issues geïntroduceerd door de verschillende orientaties. Dit vereist uitbreidingen aan de procedure voor het uitzoeken van de koppelingen tussen de verschillende domeinen, en aan de mechanismes voor het initialiseren van de communicatiebibliotheek. Bij dit laatste lijkt de uitgebreidere definitie van indexsets uit parallel Kalman te kunnen worden gebruikt.

3.6 Werkplan

Ten aanzien van het realiseren van nieuwe koppelingen wordt het volgende stappenplan voorgesteld:

1. Toestaan van het per deeldomein in- of uitschakelen van de k -Nikuradse ruwheidsberekening. Dit is al mogelijk met de huidige programmatuur, maar is alleen nog niet getest.
2. Realiseren van de mogelijkheid om kromlijnige roosters aan bolcoördinaten te kunnen koppelen. Dit betreft weinig werk.
3. Implementeren van de mogelijkheid om transport en turbulentie per deeldomein aan of uit te kunnen zetten. In eerste instantie voor het transportgedeelte alleen waar er wordt gekoppeld op openingen. Dit levert met relatief weinig werk een flinke performancewinst.
4. Implementeren van 1 : 1 tijdstapkoppeling. Dit wordt enerzijds gedaan om x - aan y -roosterlijnen te kunnen koppelen, anderzijds als voorbereiding op 1 : n tijdstapkoppeling.
5. Maken van een prototype met 1 : n tijdstapkoppeling. Het doel hiervan is om inzicht te krijgen in de problematiek: tijdsintegratie-schemas, tijdsinterpolatie in communicaties. Pas zodra hier wat ervaring mee is opgedaan kunnen goede schema's worden bedacht. Het is ook voordelig om de issues beter in kaart te hebben op het moment dat er in het kader van OMS aan wordt gewerkt.
6. Dynamische barrieresturing over meerdere DDHOR domeinen realiseren.
7. Operationaliseren van de 1 : n tijdstapkoppeling.
8. WAQUA-TRIWAQ koppeling realiseren. Dit wordt zoveel mogelijk uitgesteld omdat WAQUA en TRIWAQ eerst intern op elkaar zouden moeten worden afgestemd (gebruik van LGRID, droogvalmodellering). Daarna zou het resterende deel van de ontwikkeling van de koppeling overbodig geworden kunnen zijn in verband met de komst van OMS.

Referenties

- [1] B.H.M. Gerritsen, W. Zijl, F.J.T. Floris, and J.L. van der Meij. Invoering domein-decompositie RIKZ, flexibele koppeling roosters en procesmodellen. Technical Report GG R 96-76 C, TNO-GG, P.O.Box 6012, 2600 JA Delft, December 1996.
- [2] B. van 't Hof, C. van Velzen, and E.A.H. Vollebregt. Cursusboek module 2, hands-on training domein decompositie. In *Cursus "Domein decompositie en parallel rekenen in SIMONA"*. VORtech Computing, Oktober 2002.
- [3] M.R.T. Roest and E.A.H. Vollebregt. Ontwikkeling domein decompositie op middellange termijn. Technical Report TR98-04, VORtech Computing, Postbus 260, 2600 AG Delft, Maart 1998. In opdracht van RIKZ.
- [4] M.R.T. Roest, E.A.H. Vollebregt, and B. van 't Hof. Cursusboek module 1, introductie domein decompositie en parallel rekenen in SIMONA. In *Cursus "Domein decompositie en parallel rekenen in SIMONA"*. VORtech Computing, Oktober 2002.
- [5] M.R.T. Roest, E.A.H. Vollebregt, and B. van 't Hof. Cursusboek module 3, programmeren aan parallel WAQUA en TRIWAQ. In *Cursus "Domein decompositie en parallel rekenen in SIMONA"*. VORtech Computing, Oktober 2002.
- [6] E.A.H. Vollebregt, B. van 't Hof, and M.R.T. Roest. Ontwerp van de DDVERT functionaliteit in de standaardversie van WAQUA/TRIWAQ. Technical Report TR01-02, VORtech Computing, Postbus 260, 2600 AG Delft, Januari 2001. In opdracht van RIKZ.
- [7] E.A.H. Vollebregt, M.R.T. Roest, and B. van 't Hof. Detailontwerp domein decompositie met horizontale verfijning. Technical Report TR01-06, VORtech Computing, Postbus 260, 2600 AG Delft, Nederland, April 2001. In opdracht van RIKZ.
- [8] E.A.H. Vollebregt and J.A.Th.M. van Kester. Inventarisatie van problemen en oplossingen m.b.t. droogvallen en onderlopen in WAQUA en TRIWAQ. Technical Report TR02-13, VORtech Computing, P.O.Box 260, 2600 AG Delft, the Netherlands, November 2002.