

Postbus 260
2600 AG DELFT

tel. 015-285 0125
fax. 015-285 0126
vortech@vortech.nl

Detailontwerp uniformering WAQUA/TRIWAQ

Technisch Rapport TR06-05 versie 0.2 (28 juli 2006)

Datum

28 juli 2006

Auteur(s)

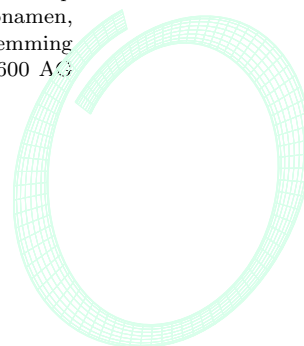
Bas van 't Hof en Edwin Vollebregt

In opdracht van

Rijkswaterstaat/RIKZ, overeenkomst RKZ-1629

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 2006.



Log-sheet

Versie	Auteur	Datum	Opmerkingen	Review
0.1	EV, BvtH	12-07-2006	Eerste analyse en plan van aanpak	
0.2	EV, BvtH	20-07-2006	Verdere uitwerking van strategie en aanpak	

Inhoudsopgave

Log-sheet	2
1 Inleiding	5
2 Opname niet-hydrostatische druk	7
2.1 Inleiding	7
2.2 Nieuwe mogelijkheden in de niet-hydrostatische versie van TRIWAQ	7
2.3 Aanpak voor de opname van niet-hydrostatisch rekenen	8
2.4 Aandachtspunten en mogelijke verbetering	9
2.5 Plan van aanpak en tijdschatting	12
2.6 Aanpassingen in de programmatuur	18
2.7 Gedetailleerde beschrijving van de aanpassingen	18
2.7.1 De routine <code>washdy</code>	18
2.7.2 Nieuwe arrays	19
2.7.3 Nieuwe invoer	22
2.7.4 Aanpassingen in bestaande rekenroutines	23
2.7.5 Verwerking van niet-hydrostatische druk in bestaande rekenroutines	25
3 Opname horizontaal $k - \epsilon$ model en parallellisatie	27
3.1 Inleiding	27
3.2 Analyse van benodigde werkzaamheden	27
3.3 Plan van aanpak en tijdschatting	28
4 Uniformering WAQUA/TRIWAQ: aanpassing van arrays	30
4.1 Inleiding	30
4.2 Vermijden gebruik van <code>mnmaxk</code> -arrays	30
4.3 Uniformeren van arrays	32

4.4	Tijdschatting	34
5	Uniformering WAQUA/TRIWAQ: migreren eenvoudige functionaliteit	36
5.1	Beschikbaar maken van functionaliteit in WAQUA en TRIWAQ	36
5.2	Mergen van eenvoudige rekenroutines	38
5.3	Aandachtspunten	41
5.4	Tijdschatting	42
6	Uniformering WAQUA/TRIWAQ: mergen van het echte rekenhart	43
6.1	Inleiding	43
6.2	Hoofdlijnen migreren rekenroutines	43
6.3	Verdere uitwerking migreren rekenroutines	44
6.4	Globale tijdschatting	50
	Referenties	51

Hoofdstuk 1

Inleiding

In 2004 heeft VORtech met assistentie van WL|Delft Hydraulics het droogvalproject uitgevoerd. Hierbij zijn de data-structuren van WAQUA/TRIWAQ een stuk dichter naar elkaar toe gebracht. Ook zijn de droogvalalgoritmes van WAQUA en TRIWAQ op elkaar afgestemd.

Midden 2005 heeft VORtech in het kader van de bid-procedure voor het SIMONA beheer en onderhoud een concept offerte geschreven voor de uniformering van WAQUA en TRIWAQ. In april 2006 heeft Mustapha Elorche in het verlengde hiervan een werkplan voor de uniformering aan VORtech toegestuurd. Hierin zijn ten opzichte van eerdere plannen twee nieuwe activiteiten toegevoegd: opname van horizontale $k - \epsilon$ model en opname van niet hydrostatisch rekenen. Daarbij zijn zes fasen voorgesteld, waarvan de laatste opname van de geüniformeerde code in de moederversie is.

In overleg is deze fasering aangepast. Het lijkt ons verstandig dat er ook tussentijds delen van het project worden opgenomen. Anders is er aan het einde een grote integratieslag nodig, omdat er ook in andere activiteiten aan WAQUA/TRIWAQ wordt gewerkt. Verder is de focus van de eerste fase (functioneel en technisch ontwerp) wat aangepast. Hierin wordt vooral de strategie bepaald en worden de belangrijkste consequenties daarvan uitgewerkt. Het daadwerkelijk gedetailleerd ontwerp wordt dan later per onderdeel apart bepaald. De reden hiervoor is dat de broncode in alle stappen flink wijzigt en dat niet alle gevolgen daarvan nu al goed kunnen worden overzien. Dus ook niet hoe de stappen doorwerken in het vervolg.

De fasering voor het totale project is nu:

1. fase 1: globaal functioneel/technisch ontwerp
2. fase 2: integratie en opname van niet-hydrostatisch rekenen.
3. fase 3: integratie en opname van horizontaal $k - \epsilon$ model
4. fase 4: uniformering WAQUA/TRIWAQ: aanpassen van arrays
5. fase 5: uniformering WAQUA/TRIWAQ: migreren eenvoudige functionaliteit
6. fase 6: uniformering WAQUA/TRIWAQ: samenvoegen van echte rekenroutines

Qua uitvoering door VORtech kunnen er verschillende medewerkers tegelijk worden ingezet: Bas van 't Hof, Edwin Vollebregt, Jeroen Gerrits en Erwin Loots. Om hun capaciteit te kunnen benutten moeten de verschillende fases deels parallel aan elkaar worden uitgevoerd. Jeroen en Erwin kunnen gemakkelijk bijdragen aan stappen 2, 4 en 5, en minder gemakkelijk meewerken aan stappen 3 en 6. Stappen 5 en 6 zijn afhankelijk van stap 4. Stappen 2 en 3 moeten niet te veel overlappen met stappen 4, 5 en 6.

In het overleg van 13 juli bij RIKZ heeft Mustapha Elorche aangegeven dat gebruikers uitkijken naar de resultaten van stap 5. Verder is het gewenst dat stap 2 snel wordt uitgevoerd omdat deze dan nog (grotendeels) door Marcel Zijlema kan worden begeleid. Naar aanleiding hiervan is het voorstel om tegelijkertijd te beginnen met stappen 2, 3 en 4, zodra stap 4 klaar is te beginnen met stap 5, en om stap 6 als laatste te doen.

Hoofdstuk 2

Opname niet-hydrostatische druk

2.1 Inleiding

In de afgelopen jaren heeft Marcel Zijlema een speciale versie van TRIWAQ ontwikkeld met de mogelijkheid van “niet-hydrostatisch rekenen” [4]. Deze versie is midden 2005 nog geactualiseerd ten opzichte van de moederversie van de programmatuur. Er wordt gevraagd om de uitbreidingen op te nemen in de moederversie, zodat ze later voor bijvoorbeeld voor koelwaterberekeningen voor het Noordzeekanaal kan worden gebruikt.

2.2 Nieuwe mogelijkheden in de niet-hydrostatische versie van TRIWAQ

In dit project hebben we de code bestudeerd waarin niet-hydrostatische effecten kunnen worden berekend met TRIWAQ. Er zit een groot aantal uitbreidingen in deze code. Een groot gedeelte van deze uitbreidingen lijkt niet direct verband te houden met niet-hydrostatisch rekenen: ze kunnen ook interessant zijn voor hydrostatische berekeningen, en ook is het denkbaar dat betekenisvolle niet-hydrostatische berekeningen kunnen worden uitgevoerd zonder gebruik te maken van alle uitbreidingen.

De volgende uitbreidingen zijn gevonden in de niet-hydrostatische versie van TRIWAQ:

- De eigenlijke niet-hydrostatische vergelijkingen.

De verticale stroomsnelheden worden hierbij berekend met behulp van de impulsvergelijking in de z -richting. De stroming wordt na elke tijdstap weer divergentievrij gemaakt door een zogenaamde *drukcorrectiestap*. Hiervoor wordt een Poisson-vergelijking opgelost voor het hele 3D-veld.

- Behoudende upwind advection-operatoren.

De advection wordt berekend met behulp van een nieuw discretisatieschema dat onder andere verschilt van de gebruikelijke discretisaties in dat het impulsbehoudend is. De nieuwe advection-operator maakt gebruik van upwind discretisaties.

- Alternatief bodemwrijvingsmodel.

In de code wordt een aangepaste term gebruikt voor de berekening van de bodemwrijving, waarin ook de bodemhelling wordt verrekend.

- “Full Reynolds stresses” voor viscositeit.

Hiermee wordt een alternatieve formulering geboden voor de (turbulente) horizontale viscositeit.

- Een nieuw model is beschikbaar voor verticale viscositeit.

Via de invoer kan een model geselecteerd worden voor de verticale viscositeit dat is bedoeld voor de simulatie van eddy viscositeit in brekende golven.

- Er is een nieuw model gemaakt voor de productie van turbulente energie.

- Een alternatieve manier voor het langzaam aanzetten van randvoorwaarden.

In plaats van de bekende *smoothing*-optie voor de randvoorwaarden, kan de randvoorwaarde geleidelijk worden ingeschakeld via een “ramp function”. De gebruikte functie (die gebruik maakt van de `tanh`-functie) is gladder dan de functie die wordt gebruikt bij *smoothing* (een lineair oplopende functie).

- Sponge layers.

Nabij de randen van het domein kan een zogenaamde *sponge layer* worden gedefinieerd. Een sponge layer is een gebied waarin een extra term wordt toegevoegd aan de impulsvergelijking waardoor de snelheid en energie afnemen.

- Nieuwe randvoorwaarde: Sommerfeld’s stralingsconditie.

Sommerfeld’s stralingsconditie is een zwak-reflecterende randvoorwaarde die is gebaseerd op een verband tussen de waterstand en de waterstandsgradiënt.

- Het is mogelijk gemaakt om een randvoorwaarde op te geven voor de turbulente intensiteit op instroomranden.

2.3 Aanpak voor de opname van niet-hydrostatisch rekenen

De nieuwe functionaliteiten zijn niet allemaal voldoende ver uitgewerkt (bijv. alleen voor linker randen, of met *ad hoc* aansturing in de simulatie-invoerfile) en gedocumenteerd om direct te kunnen worden opgenomen in de moederversie. Daarom is er overlegd met RIKZ over welke features moeten worden overgenomen en hoe de huidige implementatie verder moet worden verfijnd.

Het combineren van de nieuwe functionaliteiten met domein decompositie (inclusief proceskoppeling, niet-hydrostatisch in het ene en hydrostatisch in het andere domein) en Kalman filtering vergt een aanzienlijke extra inspanning. Dit valt buiten de mogelijkheden van het huidige project en wordt daarom vooralsnog niet uitgewerkt. Dit kan naderhand, wanneer

we meer kennis van de nieuwe functionaliteit hebben opgedaan, in een aparte fase of project wordt gedaan.

De opname van de uitbreidingen van de hydrodynamische versie van TRIWAQ in de moeder-versie van WAQUA/TRIWAQ in SIMONA zal met de volgende stappen worden gedaan:

1. In overleg met Marcel Zijlema is besloten welke uitbreidingen op welke manier zullen worden opgenomen in de nieuwe versie.
2. De wijzigingen van de niet-hydrostatische versie worden in de nieuwste versie van WAQUA/TRIWAQ geïntegreerd, zonder herstructurering, behoudens wijzigingen waarvan besloten is dat ze niet moeten worden overgenomen. De eerste geïntegreerde versie wordt getest (vergeleken met de testversie van niet-hydrostatisch rekenen), en op een branch ingecheckt in het versiebeheersysteem Subversion.
3. De eerste geïntegreerde versie wordt stap voor stap aangepast tot een testversie die alleen de gewenste nieuwe features bevat en waarin die features op de gewenste manier zijn geïmplementeerd. Hierbij wordt ook aandacht besteed aan (validatie van de) parallellisatie van de programmatuur en worden de gewenste aanpassingen aan de documentatie uitgewerkt.
4. De nieuwe versie van de programmatuur wordt grondig getest, zowel wat betreft de nieuwe functionaliteit en documentatie, als via regressietesten voor oude functionaliteit.
5. Na acceptatie van de nieuwe versie worden de uitbreidingen in de moederversie geïntegreerd.

2.4 Aandachtspunten en mogelijke verbetering

De volgende onderdelen worden (nog) niet opgenomen in de code. In plaats daarvan zal een error of een warning worden gegeven wanneer de gebruikersinvoer vraagt om een berekening die nog niet (voldoende) wordt ondersteund:

1. In principe is het mogelijk om niet-hydrostatische berekeningen “exact” te restarten, maar dat is nog niet volledig geïmplementeerd. Het is gewenst dat dit bij de opname wel wordt gedaan. Hiervoor moeten de relevante arrays op de SDS-file worden gezet en bij een restart worden gelezen door WAQPRE.
2. Parallellisatie is voldoende efficiënt en correct uitgevoerd. Alleen de preconditioner is niet geparallelliseerd. Hierdoor zijn er in parallele berekeningen soms (iets) meer iteraties nodig, maar dat is niet erg. Mogelijk zijn wel aanpassingen aan de parallellisatie nodig vanwege recente ontwikkelingen in TRIWAQ. Dit moet bij de opname worden nagegaan.
3. Domein decompositie met horizontale verfijning zou zeer nuttig kunnen zijn bij niet-hydrostatisch rekenen, om de rekestijd te kunnen beperken. Het voert echter te ver om dit mee te nemen in de huidige fase/het huidige project. Indien de gebruiker toch

probeert deze twee mogelijkheden te combineren dan zal er een foutmelding worden gegeven (COEXEC).

4. Proceskoppeling kan zeer nuttig zijn bij niet-hydrostatisch rekenen, omdat niet-hydrostatische effecten slechts in een klein gedeelte van het domein optreden en flink wat extra rekentijd kosten. Het voert echter te ver om dit uit te werken in de huidige fase/het huidige project.
5. Niet-hydrostatisch rekenen wordt nog niet gecombineerd met het Kalman-filtering. Wanneer beide opties worden geselecteerd in een simulatie-invoerfile dan wordt er een foutmelding gegeven (WAQPRE).
6. De sponge layer is belangrijk omdat daarmee reflecties op randen kunnen worden tegengegaan die anders lastig/storend zijn in niet-hydrostatische berekeningen. De berekening van de afstand tot de rand in `trsfm` is echter nog niet volledig ontwikkeld:
 - de berekening van de afstand tot het einde van de rij is in geval van kromlijnige roosters heel onnauwkeurig;
 - de sponge layer zit altijd aan het einde van een rij/kolom, nooit aan het begin; bovendien aan het einde van alle rijen en kolommen waar geen koppeling is met een ander domein.

Er wordt voor gekozen om deze uitbreiding lagere prioriteit te geven. Alleen als de andere werkzaamheden voor het niet-hydrostatisch rekenen voldoende snel verlopen wordt de sponge layer in dit project meegenomen. Over de preciese manier waarop ondersteuning gewenst is (aansturing vanuit de invoer, met name op welke randen en met welke afstand tot de rand) moet dan nog worden nagedacht.

7. De effecten van barriers ontbreken nog in de nieuwe advection-operator. Deze operator is echter van groot belang voor het niet-hydrostatisch rekenen en moet daarom worden opgenomen in de moederversie van de programmatuur. Daarbij moet de correcte afhandeling van barriers worden toegevoegd, analoog aan de afhandeling van barriers in de bestaande advection-operator.

De implementatie van advection-operators zal wat generieker worden gemaakt ter voorbereiding op nog meer schema's in de toekomst. In hydrostatische berekeningen wordt het bestaande schema de default, in niet-hydrostatische berekeningen is gebruik van de behoudende operator verplicht.

8. In de testversie met niet-hydrostatisch rekenen wordt het full Reynolds model op een *ad-hoc* manier geactiveerd: door een negatieve eddy viscosity op te geven. Bij de opname van het full Reynolds model moet hiervoor een *switch*-keyword (VISCOSITY - FULL-REYNOLDS) worden geïntroduceerd.
9. Op dit moment heeft het full Reynolds model nog geen invloed op de transportvergelijking. Echter, als het model een grote mate van turbulente menging voorspelt, dan zou dat ook voor opgeloste stoffen moeten gelden. Dit zal bij de opname worden opgelost door een warning te geven wanneer het full Reynolds model wordt gecombineerd met transport.

10. Op dit moment wordt Sommerfeld's conditie op een *ad-hoc* manier geselecteerd, door een negatieve reflectiecoëfficiënt op te geven. Deze nieuwe randvoorwaarde is bedoeld om verstoringen in de simulatie te verminderen, maar werkt niet zo goed als verwacht. De sponge layer levert op een andere manier een betere demping van verstoringen op. Ze hoeft daarom niet te worden opgenomen in de moederversie van de programmatuur.
11. Bij de nieuwe randvoorwaarde voor de turbulente intensiteit op instroomranden wordt verondersteld dat er instroom is aan het begin van roosterlijnen en niet aan het einde. Deze functie is alleen geïmplementeerd ten behoeve van specifieke testen en hoeft niet te worden opgenomen in de moederversie van de programmatuur.
12. In het turbulentiemodel is een nieuwe formulering voor de productie van turbulente energie geïmplementeerd. Deze is echter alleen voor 2DV modellen uitgewerkt en wordt daarom nog niet overgenomen in de moederversie van de programmatuur. Wel zal een warning worden gegeven wanneer het full Reynolds model wordt gecombineerd met turbulentie.
13. De nieuwe *ramp function* heeft hetzelfde doel als de bestaande *smoothing*-optie, maar geeft een geleidelijkere overgang. De verbetering is heel belangrijk voor modellen waarin korte golven een rol spelen en moet daarom worden opgenomen in WAQUA/TRIWAQ. Hierbij moet worden nagedacht hoe en hoe lang de bestaande smoothing-optie wordt gehandhaafd:
 - SMOOTHING selecteert per direct de nieuwe ramp-function. Dit veroorzaakt veranderingen in de uitkomsten van bestaande modellen. Voor compatibiliteit wordt het keyword OLD_SMOOTHING toegevoegd, en dit keyword wordt na circa een jaar verwijderd.
 - RAMP_SMOOTHING wordt toegevoegd, SMOOTHING houdt zijn huidige betekenis, en wordt na circa een jaar verwijderd.
 - SMOOTHING en RAMP_SMOOTHING worden naast elkaar geïmplementeerd en blijven permanent bestaan.

Onze voorkeur gaat uit naar de eerste optie.

14. Er is een index set $kmax4=(-1:kmax+2)$ toegevoegd. Deze index set is slechts bedoeld om bepaalde berekeningen (matrix-vector operaties) efficiënter te kunnen programmeren, vergelijkbaar met $(-2:mmax+3)$. In de extra lagen $-1, 0, kmax+1, kmax+2$ staan altijd nullen. Deze index-set zal worden omzeild door `cocupd` handig aan te roepen.
15. Er is nog slechts weinig documentatie van de nieuwe berekeningen. Marcel Zijlema heeft ons zijn aantekeningen doen toekomen. Deze zullen met het artikel [4] als basis dienen voor de technische documentatie.
16. In berekeningen met niet-hydrostatische druk wordt de laagdikte in een u -punt anders berekend dan in de standaard-code. Dit is bedoeld voor de impulsbehoudende advection-operator. De nieuwe formule zal worden geïmplementeerd in de berekening van array `hkuh`, zodat ze op minder plekken in de code tot verschillen leidt. Ze zal steeds worden geactiveerd wanneer de behoudende advection-operator wordt gebruikt.

17. In de uitbreidingen wordt veel gebruik gemaakt van `irogeo`-loops. Waar mogelijk worden dergelijke loops vervangen door de modernere `nmloop`-loops, die sneller kunnen worden verwerkt en waarmee eenvoudiger en duidelijker kan worden aangegeven om welke punten het precies gaat.
18. Het nieuwe viscositeitsmodel voor brekende golven wordt momenteel geactiveerd door een waarde voor parameter `FV` op te geven in `TURBULENCE_MODEL - EMPIRICAL_CONSTANTS`. Het lijkt ons gewenst om hier een andere naam voor te hanteren, of om een switch `BREAKING_WAVES` toe te voegen voor activeren van dit submodel.
19. Bij het bestuderen van de niet-hydrostatische berekeningen hebben we diverse mogelijkheden voor verbeteringen bedacht. Met name zou de drukcorrectie $A \cdot q$ kunnen worden doorgegeven naar subroutines `trscue` en `trsumo` in plaats van de matrix A en de druk q . Dit vereenvoudigt deze routines en kan worden gebruikt om de benodigde hoeveelheid geheugen te verminderen. Ook zouden belangrijke grootheden (m.n. q en $A \cdot q$) uit het niet-hydrostatische model als uitvoer op de SDS-file kunnen worden gezet.

2.5 Plan van aanpak en tijdschatting

In paragraaf 2.3 is de globale aanpak voor de opname van de uitbreidingen van de testversie met niet-hydrostatisch rekenen besproken. Globaal is de strategie om eerst snel een versie te maken die kan worden vergeleken met de testversie, en dan deze versie stap voor stap te verbeteren. Dit wordt met de volgende stappen bereikt:

1. Recht-toe-recht-aan opname zonder herstructurering van de aangeleverde programma-tuur (met name `irogeo`-loops) van:
 - de behoudende advectionoperator (nog niet mergen met bestaande operator, nog geen ondersteuning van barriers, afwijkende laagdiktes nog niet via `hkuh`);
Aanpassingen beperken zich tot `trsumo` en `trscue`. Aansturing via een `intgda`-entry, hard-coded gezet in initialisatiefase.
Ik denk dat de aanpassingen (bijna) triviaal kunnen worden overgenomen. Daarna flink testen, zodat ook informatie wordt verkregen over het effect van deze nieuwe advection-operator. De testresultaten kunnen vast worden opgenomen in het testverslag.
Programmeren: 1 uur
Testen: 6 uur
 - het nieuwe bodemwrijvingsmodel;
Aanpassingen beperken zich ook hier tot `trsumo` en `trscue`. Aansturing weer via een `intgda`-entry, hard-coded gezet in initialisatiefase.
Ik denk weer dat de aanpassingen (bijna) triviaal kunnen worden overgenomen. Daarna flink testen, zodat ook informatie wordt verkregen over het effect van deze nieuwe bodemwrijving. De testresultaten kunnen vast worden opgenomen in het testverslag.

Programmeren: 1 uur

Testen: 4 uur

- het full Reynolds model (met nieuwe aansturing in WAQPRE en de benodigde warnings);

Aanpassingen zijn niet alleen nodig in `trsumo` en `trscue`, maar ook in WAQPRE, en in het doorsturen van de invoer naar de berekeningen.

Ik denk weer dat de aanpassingen (bijna) triviaal kunnen worden overgenomen. Daarna flink testen, zodat ook informatie wordt verkregen over het effect van dit nieuwe viscositeitsmodel. De testresultaten kunnen vast worden opgenomen in het testverslag.

Programmeren: 3 uur

Testen: 5 uur

- het nieuwe viscositeitsmodel voor brekende golven;

Aanpassingen kunnen denk ik ongewijzigd worden overgenomen uit `trptm2`, `wassfc`, `awasre`, `wasxru`, `wasxrv` en `trsviz`.

Wel kan moet denk ik het keyword een beetje worden aangepast, omdat er een heel nieuw model wordt geactiveerd door het invullen van het onbeduidend lijkende keyword FV. Misschien moet er een swtch-keyword BREAKING_WAVES bij komen, zodat het lijkt of FV een subkeyword is van BREAKING_WAVES.

Ook hier moet weer worden gekeken wat het effect is van het nieuwe model. De testresultaten kunnen vast worden opgenomen in het testverslag.

Programmeren: 3 uur.

Testen: 4 uur.

- Smoothing met ramp-function;

De uitbreiding wordt voorlopig ongewijzigd overgenomen.

Om te testen hoeft alleen met wat print-statements gecontroleerd te worden of de juiste werking wordt verkregen.

Programmeren: 1 uur

Testen: 1 uur

- het niet-hydrostatisch rekenen (inclusief foutmeldingen voor `dom.dec` en Kalman).

De parallellisatie-dingen worden ongewijzigd overgenomen. Er wordt voorlopig alleen sequentieel getest.

De routines kunnen denk ik gewoon worden overgenomen.

Deze moeten daarna natuurlijk wel worden ingepast, en via WAQPRE worden aangestuurd. Arrays moeten worden aangemaakt. De druk-correctieterm wordt ingevoerd in `trsumo` en `trscue`.

Programmeren: 1 dag.

Testen: in de volgende activiteit.

2. Uitgebreid testen van de nieuwe functionaliteit in de nieuwe versie ten opzichte van de testversie.

Het niet-hydrostatische model wordt aangezet voor beide testen.

De resultaten worden vergeleken met die van het prototype.

De testen uit de voorgaande stappen worden nog eens over gedaan: er wordt uitgezocht welke invloed op de uitkomsten elk van de aanpassingen heeft.

Het voorlopige testrapport wordt nog eens doorgewerkt om de resultaten op te nemen.

Testen: 2 dagen

Rapportage: 4 uur.

3. Invoering van `nmloop` in plaats van `irogeo`-loops, testen.

Deze aanpassingen leveren geen enkele verandering van de resultaten. Wel wordt de code korter en duidelijker.

Het gaat om 87 `irogeo`-loops, die wellicht niet allemaal kunnen worden vervangen.

Programmeren: 4 uur

Testen: 4 uur

4. Verfijning van de behoudende advectionoperator en testen:

- verwerken van afwijkende laagdiktes via `hkuh`;

Bij de aanpassing van de laagdiktes moet zeer nauwkeurig te werk worden gegaan. Eerst wordt gecontroleerd of de oude formulering van `hkuh` nog ergens gebruikt wordt in geval van behoudende advection. Als dat niet zo is, moet de berekening van `hkuh` worden aangepast. Vervolgens wordt gecontroleerd of de programmatuur nog alles het zelfde doet.

Dit kan denk ik in 1 dag.

- ondersteunen van barriers;

Ik denk dat het nuttig is om de barriers eerst goed te maken. Daardoor wordt het duidelijker wat de generieke advection-operator allemaal moet kunnen.

In TRIWAQ bestaan de aanpassingen voor barriers aan de advection alleen maar daaruit, dat tweede orde centrale differenties worden gebruikt. Dat is heel eenvoudig toe te voegen aan de beschikbare code.

Programmeren: 2 uur.

Testen: 4 uur.

- aansturing via de simulatie-invoerfile;

De advection wordt instelbaar gemaakt met het nieuwe keyword, `CONSERVATIVE_ADVECTION`.

Deze moet worden aangemaakt, ingelezen, opgeslagen, gelezen en doorgegeven.

Daarna moet worden gecontroleerd dat de resultaten het zelfde zijn.

Programmeren: 4 uur.

- samenvoegen van verschillende advection-operatoren (verminderen dubbele code); Dit is in feite een optionele uitbreiding. Het zal de code sterk vereenvoudigen, en ook de mogelijkheid geven om andere advection-operatoren eenvoudig in te bouwen

voor toekomstige ontwikkelingen. Echter, het levert niet direct extra mogelijkheden op.

Het ontwerp van een flexibel systeem van ondersteuning van advection-operatoren en het analyseren van bestaande operatoren om ze in de nieuwe, generieke mal te stoppen moet zorgvuldig gebeuren.

Ik denk dat daar 1 dagen mee gemoeid zijn. Daarna moet er nog getest worden. Ik denk dat dat in een dag kan.

Samen: 2 dagen

5. Verfijning van de niet-hydrostatische berekeningen. Met name het doorgeven van de gradient $A \cdot q$ naar de discretisatie van de impulsvergelijking in plaats van apart doorgeven van A en q .

6. Er moet worden nagegaan of recente ontwikkelingen binnen TRIWAQ aanpassingen in de parallelisatie nodig maken. Bijwerken van de data-analyse van TRIWAQ en vergelijken van sequentiële en parallelle berekeningen.

Een data-analyse kan Edwin snel maken: in twee uur denk ik. Daarna moet een communicatieschema worden opgesteld en vergeleken met het schema in het prototype. Ik denk dat de parallelisatie een dag kost. Daarna moet een (aantal) test(en) worden uitgevoerd totdat de parallelle resultaten zeer goed overeenkomen met de sequentiële. Ook moet een DDVERT-test worden uitgevoerd. Dit kost ook weer een dag, denk ik.

Samen 2 dagen

7. Aanpassen van de gebruikersdocumentatie van TRIWAQ, vermelden van de nieuwe opties.

Er komen niet zo veel nieuwe keywords bij: tussen 10 en 15. Deze moeten in Word worden beschreven. Dit kan denk ik in een dag.

8. Aanpassen van de technische documentatie van TRIWAQ, beschrijven van de nieuwe formules die zijn geïmplementeerd.

Marcel Zijlema heeft ons zijn aantekeningen doen toekomen. Deze zullen als basis dienen voor de technische documentatie, die voor de niet-hydrostatische berekeningen nog moet worden gemaakt.

Als de technische documentatie door VORtech moet worden gemaakt, zal daar ruim de tijd voor moeten worden genomen. Ik stel me voor: 4 dagen studeren, 2 dagen opschrijven/intern reviewen/verbeteren, 1 dag Marcel Zijlema laten lezen, 1 dag verbeteringen doorvoeren.

Samen: 8 dagen, waarvan 1 door Marcel Zijlema.

9. Rapportage over de werkzaamheden en de uitgevoerde testen.

In de voorgaande activiteiten is al het een en ander genoemd over rapportage. Als dat inderdaad is gedaan, kan de rapportage op dit moment worden afgemaakt in 1 dag.

10. Opname en verfijning van smoothing with ramp-function, inclusief gebruikersdocumentatie.

Het aanmaken, lezen en opslaan van de benodigde keywords kan in een halve dag worden uitgevoerd. Het toevoegen van de nieuwe smoothing-techniek kan ook in een halve dag.

Daarna moet een test worden uitgevoerd, waaruit het effect van de nieuwe smoothing blijkt. Dat kan in een halve dag.

Samen: 1.5 dag

11. Mogelijk maken van exacte restarts van niet-hydrostatische berekeningen, inclusief testen.

In mijn aantekeningen heb ik al een aantal arrays aangemerkt:

- *Verticale stroomsnelheden in interfaces: `whphysi(fullbox_s * kmax0)`. Dit array kan het best worden opgenomen onder de naam `RESTART_NONHYDROSTATIC_WPHYSI`, om vewarring te voorkomen met `SOLUTION_FLOW_WPHYS`. In de code wordt het `wphysz` genoemd, maar dat lijkt te veel op een typfout.*
- *De preconditioner: array `prec(25*kmax4*fullbox_*)` Dit array kan worden opgeslagen in `RESTART_NONHYDROSTATIC_PREC`.*
- *De arrays `q(fullbox_s * kmax4)`, `dqdx(fullbox_u * kmax)` en `dqdy(fullbox_v * kmax)`. Deze arrays kunnen worden opgeslagen in `RESTART_NOHYDROSTATIC_PRESSURE`, `RESTART_NOHYDROSTATIC_DQDX` en `RESTART_NOHYDROSTATIC_DQDY`.*

Van al deze arrays moeten de extra lagen `-1,0`, `kmax+1`, `kmax+2` niet op de SDS-file worden opgeslagen. Ook moeten ze worden geconverteerd van `fullbox` naar `mmaxk`.

Het aanmaken van een SDS-array, en het kopiëren in geval van een restart, en het inlezen in de initialisatiefase van WAQPRO kost samen een dag. Echter, daarna moet worden getest en geknutseld totdat de restart exact dezelfde resultaten geeft als de oorspronkelijke run. Hiervoor moet je denk ik 2 dagen rekenen.

Samen: 3 dagen

12. Opname en verfijning van de sponge layer, inclusief gebruikersdocumentatie.

Alleen als de andere werkzaamheden voor het niet-hydrostatisch rekenen voldoende snel verlopen wordt de sponge layer in dit project meegenomen. Over de preciese manier waarop ondersteuning gewenst is (aansturing vanuit de invoer) moet dan nog worden nagedacht.

Stap	Omschrijving	Inspanning
1.	Voorlopige opname	
	Behoudende advection	7 uur
	Bodemwrijving	5 uur
	Full Renolds model	8 uur
	Viscositeit voor breaking wave model	7 uur
	Smoothing met ramp functie	2 uur
	Niet-hydrostatisch rekenen	8 uur
2.	Testen	20 uur
3.	<code>nmloop</code> -loops	8 uur
4.	Verfijning behoudende advection	
	Laagdikttes via <code>hkuh</code>	8 uur
	Ondersteuning van barriers	6 uur
	Aansturing via <code>siminp</code> -file	4 uur
	Generieke advection-codering (optioneel)	16 uur
5.	Herstructurering niet-hydro berekening	16 uur
6.	Evaluatie en perfectie parallelisatie	16 uur
7.	Gebruikersdocumentatie	8 uur
8.	Technische documentatie	64 uur
9.	Rapportage	8 uur
10.	Verfijning smoothing met ramp-functie	12 uur
11.	Exacte restarts	24 uur
12.	Opname sponge layer	24 uur
	Totaal	281 uur

Het totaal voor niet-hydrostatische TRIWAQ komt daarmee op 34 dagen, waarvan 1 door Marcel Zijlema.

Voor deze stappen verwachten we de volgende tijd nodig te hebben:

Stap	Omschrijving	Inspanning
1.	recht-toe-recht-aan integratie	5 dagen
2.	uitgebreid testen t.o.v. testversie	3 dagen
3.	invoeren <code>nmloop</code> i.p.v. <code>irogeo</code> -loops	1 dag
4.	verfijnen behoudende advection-operator	2 dagen
5.	verfijnen niet-hydrostatische berekeningen	2 dagen
6.	verifiëren parallelisatie nieuwe functionaliteit	2 dagen
7.	aanpassen gebruikersdocumentatie	1 dag
8.	uitbreiden technische documentatie	7 dagen
9.	rapportage werkzaamheden en testen	1 dag
10.	opname smoothing with ramp-function	1 dag
11.	mogelijk maken exacte restarts	3 dagen
12.	opname en verfijning sponge layer	3 dagen
	projectleiding en onvoorzien 15%	5 dagen
	TOTAAL	38 dagen

2.6 Aanpassingen in de programmatuur

De ons toegestuurde code voor berekeningen met niet-hydrostatische druk bevat de volgende aanpassingen met betrekking tot de standaard-SIMONA versie, te weten `simona0506`:

- In `wassim` is een call toegevoegd naar de routine `washdy`.
- Er zijn nieuwe arrays toegevoegd.
- Er zijn keywords toegevoegd aan `waquaref.tab`.
- Er zijn uitbreidingen gemaakt in bestaande routines.

Van deze uitbreidingen moet worden vastgesteld in hoeverre ze horen bij de niet-hydrostatische berekening. Mogelijk gaat het om uitbreidingen of verbeteringen die ook voor hydrostatische berekeningen kunnen worden gebruikt.

Omgekeerd is het ook mogelijk gewenst om de oorspronkelijke berekening te kunnen gebruiken in niet-hydrostatische berekeningen.

- Er is ondersteuning gemaakt voor niet-hydrostatisch rekenen in de bestaande routines.
- Er zijn timers toegevoegd in `csitim.i` en `cwatim.i`.
- Extra uitvoer naar de message file wordt gegenereerd door de routines `waspif`, `waspst` en `waspsi`.
- De koproutines `waschf`, `wasei3`, `wasmaf`, `wasxru`, `wasxrv`, `wastur`, `wasspu`, `wasspv` en `wassff` zijn aangepast om de rekenroutines correct aan te roepen.

2.7 Gedetailleerde beschrijving van de aanpassingen

2.7.1 De routine `washdy`

In `wassim` is een call toegevoegd naar de routine `washdy`. Dit is een koproutine die calls bevat naar de nieuwe routines

- `trswmo`: stel de impulsvergelijking in z -richting op en los die op.
- `trspoi`: stel een Poisson vergelijking op voor de druk-correctie en los die op.
- `trshdu`: verwerk de aangepaste druk in de stroomsnelheden in alle 3 richtingen.

Deze routines bevatten updates en kunnen dus worden gebruikt in parallele berekeningen. Wel worden nog de oude `irogeo`-loops gebruikt. Deze moeten worden vervangen door nieuwe `nmloop`-loops.

Waarschijnlijk is een data-analyse nodig, die kan worden opgenomen in de appendix voor de Technische Documentatie van TRIWAQ [?]. Hierdoor kan de parallellisatie misschien efficiënter worden en/of kunnen fouten in de parallellisatie worden gevonden.

Domein-decompositie wordt nog niet ondersteund.

2.7.2 Nieuwe arrays

De volgende nieuwe variabelen zijn geïntroduceerd:

- **ihydro**: flag for non-hydrostatic flow computation (1/0=yes/no)
staat op CONTROL_FLOW_ICONTB(51), intgda(171,2).
- **maxit**: maximum number of iterations
staat op CONTROL_FLOW_ICONTB(52), intgda(172,2).
- **ifpco**: mfrequency with which postconditioner is applied
staat op CONTROL_FLOW_ICONTB(53), intgda(173,2).
- **iamout**: control parameter indicating the amount of output required
staat op CONTROL_FLOW_ICONTB(54), intgda(174,2).
- **imeth**: control parameter indicating the kind of approximation of vertical pressure derivative (1=standard, 2=box)
staat op CONTROL_FLOW_ICONTB(55), intgda(175,2).
- **icreep**: calculation of horizontal non-hydrostatic pressure gradient due to sigma transformation or not (1=yes, 0=no)
staat op CONTROL_FLOW_ICONTB(56), intgda(176,2).
- **nfhydr**: integer integration step for starting non-hydrostatic flow computation
staat op CONTROL_FLOW_ICONTB(57), intgda(177,2).
- **mitpoi**: minimal number of iterations `trspoi`
staat op `intgda(412,2)`. Er is inmiddels een array `MINIT` aangemaakt, waar deze waarde aan kan worden toegevoegd.
- **whpys**(`fullbox_s * kmax0`): physical w-velocity.
Aangemaakt in `wasfgd`, INFOR index in `intgda(461,2)`.

Ik denk dat dit array in `RESTART_FLOW` moet worden opgenomen. Het is bovendien verstandig om de naam te veranderen in `wphys` om het te kunnen onderscheiden van `SOLUTION_FLOW_WPHYS`. De naam `wphys` wordt al gebruikt in de code.

- **zkso**(`fullbox_s * kmax0`): layer-interfaces at previous timestep
Aangemaakt in `wasfgd`, INFOR index in `intgda(462,2)`.

Dit is een type array dat in `WAQUA` normaal nooit nodig is. De reden dat hij nu wel nodig is is omdat de drukcorrectie eenmaal per hele tijdstap wordt uitgevoerd. Daardoor is het normale p/h-schema niet meer voldoende.

- `q(fullbox_s * kmax4)`: non-hydrostatic pressure
Aangemaakt in `wasfgd`, INFOR index in `intgda(463,2)`. Ik denk dat ook dit array in `RESTART_FLOW` moet worden opgenomen.
- `dq(fullbox_s * kmax4)`: non-hydrostatic pressure correction.
Aangemaakt in `wasfgd`, INFOR index in `intgda(464,2)`.
- `gmat(3*6*kmax4*fullbox_*)`: gradient matrices.
Aangemaakt in `wasfgd`, INFOR index in `intgda(465,2)`.
- `prec(25*kmax4*fullbox_*)`: preconditioner matrix.
Aangemaakt in `wasfgd`, INFOR index in `intgda(466,2)`.
Dit array wordt alleen gebruikt in `trspoi`: op geselecteerde tijdstippen wordt dit array 'beter' gemaakt (i.e. een iteratie van een proces wordt uitgevoerd waarmee een optimale ILU-decompositie wordt verkregen). Verder wordt het alleen gelezen.
Dit array moet aan `RESTART_FLOW` worden toegevoegd, of er moet iets slims worden gedaan met herstarts.
De precieze index set waarop dit array leeft moet nog worden uitgezocht.
- `sponu(fullbox_u)`: damping function in u-points for sponge layer
Aangemaakt in `wasfgd`, INFOR index in `intgda(467,2)`, gevuld in `trsfm`, daarna alleen gebruikt in `trscue` en `trsumo`.
- `sponv(fullbox_u)`: damping function in v-points for sponge layer
Aangemaakt in `wasfgd`, INFOR index in `intgda(468,2)`, gevuld in `trsfm`, daarna alleen gebruikt in `trscue` en `trsumo`.
- `theta`: parameter in the time integration
Opgeslagen in `CONTROL_FLOW_RCONTA(71)`, `reagda(61,2)`.
- `reps1`: relative accuracy
Opgeslagen in `CONTROL_FLOW_RCONTA(72)`, `reagda(62,2)`.
- `reps2`: relative accuracy wrt right hand side
Opgeslagen in `CONTROL_FLOW_RCONTA(73)`, `reagda(63,2)`.
- `relax`: relaxation parameter meant for iterative solvers
Opgeslagen in `CONTROL_FLOW_RCONTA(74)`, `reagda(64,2)`.
- `splenu`: length of sponge layer in x-direction
Opgeslagen in `CONTROL_FLOW_RCONTA(75)`, `reagda(65,2)`.
- `splenv`: length of sponge layer in y-direction
Opgeslagen in `CONTROL_FLOW_RCONTA(76)`, `reagda(66,2)`.

- **fv**: empirical constant for eddy viscosity model
Opgeslagen in CONTROL_FLOW_RCONTA(67), reagda(29,2).
- **rtens**: turbulent intensity at inflow
Opgeslagen in CONTROL_FLOW_RCONTA(66), reagda(40,2).

Het array **gmat** moet vermoedelijk anders worden gestructureerd, omdat de entries **gmat(1,::,::)** leven op **fullbox_u**, de entries **gmat(2,::,::)** op **fullbox_v** en de entries **gmat(3,::,::)** op **fullbox_s**. Een array met een zodanige structuur kan niet goed worden gecommuniceerd.

De kolommen van dit array noem ik vanaf nu

```
gmatu(:,::,:) = gmat(1,::,::),
gmatv(:,::,:) = gmat(2,::,::),
gmatw(:,1:2,:) = gmat(3,1:2,::,::).
```

Het gebruik van de arrays **gmat**, **q** en **dq** is interessant. In een tijdstap wordt het gebruikt op de volgende manieren:

- In **trscue** wordt **gmatv*q** gebruikt.
- In **trsumo** wordt **gmatu*q** gebruikt.
- In **trscue** wordt **gmatu*q** gebruikt.
- In **trsumo** wordt **gmatv*q** gebruikt.
- In **trswmo** wordt **gmatw*q** gebruikt.
- In **trspoi** worden **gmatu**, **gmatv** en **gmatw** gevuld.
- In **trspoi** **dq** berekend.
- In **trshdu** worden **gmatu*dq**, **gmatv*dq** en **gmatw*dq** gevuld,
- In **trshdu** wordt **q:=q+dq** aangepast.

Volgens mij kunnen deze waarden het beste op de volgende manier worden opgeslagen:

- De arrays **q**, **dqdx=gmatu*q**, **dqdy=gmatv*q** en **dqdz=gmatw*q** worden toegevoegd aan **RESTART_FLOW**. Het array **q** wordt aangepast, en de arrays **dqd*** worden gevuld in **trshdu**.
- De arrays **gmatu**, **gmatv** en **gmatw** worden als **simm02**-arrays aangemaakt in **washdy**: ze bestaan slechts in routine **trshdy**.
- Het **simm02**-array **dq** bestaat in de routines **trspoi** en **trshdu**.

2.7.3 Nieuwe invoer

De volgende nieuwe keywords zijn geïntroduceerd:

- TURBULENCE_MODEL:EMPIRICAL_CONSTANTS:FV
gelezen in `trptm2`, opgeslagen in
`fv = CONTROL_FLOW_RCONTA(67) = reagda(29,2)`.
- TURBULENCE_MODEL:EMPIRICAL_CONSTANTS:INTENSITY
gelezen in `trptm2`, opgeslagen in
`rtens = CONTROL_FLOW_RCONTA(66) = reagda(40,2)`.
- HYDRODYNAMIC:THETA
gelezen in `wap073`, opgeslagen in
`theta = CONTROL_FLOW_RCONTA(71) = reagda(61,2)`.
- HYDRODYNAMIC:MAXITER
gelezen in `wap072`, opgeslagen in
`maxit = CONTROL_FLOW_ICONTB(52) = intgda(172,2)`.
- HYDRODYNAMIC:RELACCURACY
gelezen in `wap073`, opgeslagen in
`reps1 = CONTROL_FLOW_RCONTA(72) = reagda(72,2)`.
- HYDRODYNAMIC:RHSACCURACY
gelezen in `wap073`, opgeslagen in
`reps2 = CONTROL_FLOW_RCONTA(73) = reagda(73,2)`.
- HYDRODYNAMIC:RELAXATION
gelezen in `wap073`, opgeslagen in
`relax = CONTROL_FLOW_RCONTA(74) = reagda(64,2)`.
- HYDRODYNAMIC:PREC_FREQUENCY
gelezen in `wap072`, opgeslagen in
`ifpco = CONTROL_FLOW_ICONTB(53) = intgda(173,2)`.
- HYDRODYNAMIC:AMOUNT_OUTPUT
gelezen in `wap072`, opgeslagen in
`iamout = CONTROL_FLOW_ICONTB(54) = intgda(174,2)`.
- HYDRODYNAMIC:APPROXIMATION
gelezen in `wap072`, gecodeerd opgeslagen in
`imeth = CONTROL_FLOW_ICONTB(55) = intgda(175,2)`.

- HYDRODYNAMIC:SPONGE_U
gelezen in wap073, opgeslagen in
splenu = CONTROL_FLOW_RCONTA(75) = reagda(65,2).
- HYDRODYNAMIC:SPONGE_V
gelezen in wap073, opgeslagen in
splenv = CONTROL_FLOW_RCONTA(76) = reagda(66,2).
- HYDRODYNAMIC:CREEP
gelezen in wap072, opgeslagen in
icreep = CONTROL_FLOW_ICONTB(56) = intgda(176,2).
- HYDRODYNAMIC:START
gelezen in wap072, gecodeerd opgeslagen in
nfhydr = CONTROL_FLOW_ICONTB(57) = intgda(177,2).

2.7.4 Aanpassingen in bestaande rekenroutines

De volgende aanpassingen zijn doorgevoerd in de bestaande berekeningen. Van deze uitbreidingen moet worden vastgesteld in hoeverre ze horen bij de niet-hydrostatische berekening. Mogelijk gaat het om uitbreidingen of verbeteringen die ook voor hydrostatische berekeningen kunnen worden gebruikt.

Omgekeerd is het ook mogelijk gewenst om de oorspronkelijke berekening te kunnen gebruiken in niet-hydrostatische berekeningen.

1. De *sponge layer* is een nieuwe faciliteit, om 'zachte randen' te modelleren. In de 'sponge layer' wordt de waarde `sponu` opgeteld bij `bbk` zodat daar de snelheid gedempt wordt.
aanpassing in `trsfm`, `trscue`, `trsumo`.
2. In het geval van niet-hydrostatische berekening wordt een andere formulering gebruikt voor de advection. Deze wordt genoemd
a conservative formulation for advection is applied, including the corresponding curvature term. Upwind differences are used depending on the geometry, first order upwind or zero near boundary and second order upwind (BDF or LUDS) interior. Effect of barriers is not taken into account.
De effecten van barriers moeten in deze advection worden opgenomen.
aanpassing in `trscue`, `trsumo`.
3. Er is een viscositeitsmodel geïntroduceerd dat 'full Reynold stresses' heet.
Het model wordt geactiveerd door een negatieve waarde te kiezen voor de horizontale eddy viscosity coefficient
`vicouv= vico= reagda(13,2) = CONTROL_FLOW_RCONTA(42)`,
de waarde van het keyword `FLOW:PROBLEM:VISCOSITY:EDDYVISCOSITY`.

In een definitieve versie moet dit model op een andere manier worden geselecteerd.
aanpassing in `trscue`, `trsumo`.

4. In het geval van niet-hydrostatische druk wordt er een ander model gebruikt voor bodemwrijving.

aanpassing in `trscue`, `trsumo`.

5. In de berekeningen met niet-hydrostatische druk wordt de laagdikte in een u-punt (bijna?) consequent berekend uit het gemiddelde van laagdiktes in twee waterstandspunten (gemiddelde differentie van `zks`) in plaats van het array `hkuk`.

aanpassing in `trscue`, `trsumo`.

6. Een nieuwe formulering, Sommerfeld's condition', is geïntroduceerd voor de zwak reflecterende randvoorwaarden.

In plaats van een verband tussen de stroomsnelheid en de waterstand levert deze formulering een verband tussen waterstand en zijn normaalafgeleide.

Deze randvoorwaarde wordt geselecteerd met een negatieve waarde voor `ciral`, de waarde van het keyword REFL in de invoer.

aanpassing in `trsrv2`.

7. Op gesloten randen wordt de normaalafgeleide van de waterstand op nul gezet. Dit is een andere randvoorwaarde dan wordt gebruikt in de standaard-code.

Er moet worden nagegaan of deze aanpassingen ongewenste effecten kan hebben, anders kan hij worden overgenomen in de standaard-versie.

Ook moet worden nagegaan of een aanpassing nodig is bij (permanente) schotjes, waar deze randvoorwaarde immers niet wordt voorgeschreven.

8. In `trstur` wordt een randvoorwaarde opgesteld voor de turbulent intensity op instroomranden.

Hierbij wordt echter verondersteld dat er instroom is aan het begin van roosterlijnen en niet aan het einde. In een definitieve versie moet deze veronderstelling worden losgelaten.

aanpassing in `trstur`.

9. In niet-hydrostatische berekeningen wordt een heel andere formulering gebruikt voor de productie van turbulente energie (`tkepro`) dan in standaard-berekeningen.

De gebruikte term vind ik een beetje vreemd want 'asymmetrisch': er komt wel een term `dudx` in voor, maar geen term `dvdv`.

aanpassing in `trstur`.

10. Wanneer het keyword TURBULENCE_MODEL:EMPIRICAL_CONSTANTS:FV is gezet, wordt een heel ander model gebruikt voor de berekening van de verticale eddy viscositeitscoëfficiënt `vicow` dan in andere berekeningen. Hierbij wordt een zeer kleine viscositeitscoëfficiënt gebruikt in alle lagen behalve de bovenste.

aanpassing in `trsviz`.

11. De minimum- en maximumwaarden voor de turbulente tijdschaal (variabele `tscale`) zijn vrijgegeven. Het lijkt om een vergissing te gaan, omdat de betreffende regel is uitgecommentarieerd zonder verder commentaar, en dus ook in geval van hydrostatische berekeningen is aangepast.

aanpassing in `trsviz`.

12. Een correctie met de 'ramp function for smoothing incident waves' is toegevoegd aan de routine `wasfov`.

In het begin van de simulatie wordt de berekende randvoorwaarde slechts gedeeltelijk opgelegd (vermenigvuldiging met een getal dat oploopt van 0 naar 1).

Hierbij moet worden onderzocht of deze ramp function geen nare bijverschijnselen kan hebben als de randvoorwaarden niet om de waarde 0 heen liggen, maar om een andere waarde.

Een ander probleem dat moet worden onderzocht is het effect van deze optie in combinatie met restarts.

Tot slot vraag ik mij af hoe deze functionaliteit zich verhoudt tot de reeds bestaande *smoothing*.

aanpassing in `wasfov`.

2.7.5 Verwerking van niet-hydrostatische druk in bestaande rekenroutines

1. De berekening van de verticale snelheden in `SOLUTION_FLOW_WPHYS` komt in het geval van niet-hydrostatische berekening neer op de interpolatie van het array `wphys`.
aanpassing in `trscof` (berekening van `TIMEHISTORIES_FLOW_ZCRWPH`)
aanpassing in `trscpw` (berekening van `WPHYSF`, daarna `SOLUTION_FLOW_WPHYS`).
2. De druk-krachten worden gecorrigeerd met `gmat*q`. Kennelijk staat het array `q` voor het verschil tussen hydrostatische en niet-hydrostatische druk.
aanpassing in `trscue`, `trsumo`.
3. In `trssuw` wordt de verticale stromsnelheid `omega` niet berekend in niet-hydrostatische berekeningen, omdat deze snelheid immers volgt uit de impulsvergelijking `trswmo`.
aanpassing in `trssuw`.
4. De arrays `wphys`, `q`, `dq`, `gmat` en `prec`, `sponu`, `sponv` en `zkso` worden gecreëerd en geïnitieerd in `wasfgd`.
aanpassing in `wasfgd`.
5. In de routines `wasifc` en `wasixs` wordt de index set `kmax4` gedefinieerd en vermenigvuldigd met enkele `fullbox`-sets. Daarna worden er enkele stencils op actief gemaakt.

De index-set `kmax4=(-1:kmax+2)` bestaat uit de lagen en enkele extra lagen, waarschijnlijk voor de benadering van randvoorwaarden. Uiteraard is het moeilijk om te bedenken hoe arrays op deze index set moeten worden geïnterpoleerd.

Men zou kunnen uitzoeken of de extra lagen wel echt worden gebruikt.

aanpassing in `wasixs` en `wasixs`.

6. In `wassfc` worden de nieuwe variabelen in `intgda` en `reagda` opgeslagen, en wordt een foutmelding gegeven als niet-hydrostatische berekening wordt gecombineerd met domein decompositie.

aanpassing in `wassfc`.

7. In `waspif` wordt de `minit`-waarde geïntialiseerd voor de routine `trspoi`

aanpassing in `waspif`.

Hoofdstuk 3

Opname horizontaal $k - \epsilon$ model en parallelisatie

3.1 Inleiding

In 1999-2000 is er in TRIWAQ door WL/RIKZ een dieptegemiddelde $k - \epsilon$ turbulentie-model (kortweg horizontaal $k - \epsilon$ model genoemd) in een speciale versie van TRIWAQ geïmplementeerd. Begin 2001 heeft VORtech deze uitbreidingen in de DDVERT-versie van WAQUA/TRIWAQ geïntegreerd en de parallelisatie van deze functionaliteit verzorgd [1]. Inmiddels is de moederversie van WAQUA/TRIWAQ sterk doorontwikkeld. Het horizontaal $k - \epsilon$ model moet hier nu aan worden toegevoegd.

3.2 Analyse van benodigde werkzaamheden

De wijzigingen aan WAQUA/TRIWAQ voor implementatie van het dieptegemiddelde $k - \epsilon$ turbulentie-model (kortweg horizontaal $k - \epsilon$ model genoemd) worden beschreven in [1]. Deze beschrijving is *up-to-date*: de op te nemen bron-code met horizontaal $k - \epsilon$ model verschilt nauwelijks van de parallelle versie die begin 2001 is gerealiseerd. In de op te nemen versie zijn kleine verbeteringen doorgevoerd. Bijvoorbeeld is de variabelenaam `ihortu` ingevoerd zoals wordt voorgesteld in [1].

In de moederversie van WAQUA/TRIWAQ zijn een aantal aspecten die een rol spelen bij de implementatie van het horizontaal $k - \epsilon$ -model gewijzigd ten opzichte van de situatie van begin 2001. Daarom moet het werk van de opname van toen vrijwel volledig opnieuw worden gedaan. Wel kan de aanpak van toen goed worden hergebruikt.

Aspecten die een rol spelen zijn:

- In WAQUA/TRIWAQ is domein decompositie met horizontale en verticale verfijning geïmplementeerd. Hiervoor is de manier waarop loops worden geïmplementeerd gewijzigd. In plaats van `irogeo` plus `irobou`, met speciale randcodes voor subdomein interfaces, worden nu vooral `nmixst` en `nmloop` gebruikt.

- De redundante berekening van de impulsvergelijking in `trsumo` wordt niet meer gebruikt. De data-analyse van de $hor-k\epsilon$ berekeningen moet daarom opnieuw worden gedaan.
- De droogvalalgoritmes van WAQUA/TRIWAQ zijn gereorganiseerd. Hiervoor zijn ook de kenmerkarrays gewijzigd. Dit moet in de nieuwe routines voor het horizontaal $k - \epsilon$ model worden doorgevoerd.
- De arrays `umean` en `vmean` die voor het horizontaal $k - \epsilon$ model nodig zijn worden ook bij de uniformering van WAQUA en TRIWAQ aangepast.

We stellen voor dat het horizontaal $k - \epsilon$ model in eerste instantie niet met domein decompositie met horizontale verfijning mag worden gecombineerd. Daarvoor is aanvullende analyse en programmeerwerk nodig, met name omdat er in het horizontaal $k - \epsilon$ model met dieptepunten wordt gewerkt, wat binnen DDHOR nog niet eerder aan de orde is geweest. Deze uitbreiding is daarom vrij omvangrijk en kan daarom beter in een aparte fase of apart project worden uitgewerkt.

3.3 Plan van aanpak en tijdschatting

De aanpak die we voorstellen voor de opname van het horizontaal $k - \epsilon$ model is als volgt:

1. Toevoegen van de nieuwe data-structuren en variabelen in WAQPRE.
2. Uitwerken van een generieke aanpak voor dieptepunten t.b.v. sequentiëel/parallel rekenen. Invoeren van arrays `kcd` en `idpgeo` in respectievelijk WAQPRE en WAQPRO. Invoeren van extra index-sets in `nmloop` in WAQPRO.
3. Toevoegen van de nieuwe berekeningen in WAQPRO, met benodigde herstructurering n.a.v. wijzigingen in de moederversie. Toevoeging van een controle/foutmelding voor de combinatie met DDHOR.
4. Valideren van de integratie voor sequentiële berekeningen.
5. Maken van een data-analyse van de nieuwe berekeningen in het rekengedeelte van TRIWAQ, en bepalen waar in de code welke communicaties moeten worden toegevoegd.
6. Valideren van de parallelisatie van het horizontaal $k - \epsilon$ model.
7. Testen van het horizontaal $k - \epsilon$ model in combinatie met verticale verfijning.
8. Aanpassen gebruikersdocumentatie, technische documentatie, documentatie m.b.t. parallel rekenen ([2]).
9. Rapportage over de werkzaamheden en testen.

We verwachten dat voor deze stappen de volgende hoeveelheid tijd is vereist:

Stap	Omschrijving	Inspanning
1.	integratie wijzigingen WAQPRE	4 uur
2.	uitwerken aanpak dieptepunten	4 uur
3.	integratie/verfijning wijzigingen WAQPRO	12 uur
4.	testen sequentiële berekeningen	8 uur
5.	parallellisatie nieuwe berekeningen	12 uur
6.	testen parallelle berekeningen	8 uur
7.	testen verticale verfijning	4 uur
8.	aanpassen documentatie	16 uur
9.	rapportage	4 uur
	onvoorzien 10%	8 uur
TOTAAL		80 uur

Hoofdstuk 4

Uniformering WAQUA/TRIWAQ: aanpassing van arrays

4.1 Inleiding

Een belangrijk deel van de verschillen tussen WAQUA en TRIWAQ heeft te maken met dat er iets andere arrays worden gebruikt. Daardoor moet er bij het programmeren steeds worden nagedacht welke arrays wel en niet bestaan, en moet er in koproutines steeds onderscheid tussen WAQUA en TRIWAQ worden gemaakt. Het stroomlijnen van de data-structuur levert daarom een grote vereenvoudiging voor programmeurs op. Deze stap kan goed apart worden uitgevoerd en is een nuttige voorbereiding op de daadwerkelijke uniformering van de rekenroutines. Ze heeft geen invloed op de simulatieresultaten.

Het stroomlijnen van de datastructuur bestaat uit twee onderdelen:

1. Verder verstoppert van `mnmaxk`-arrays in WAQPRO, verwijderen uit `intgda`;
2. Gelijktrekken van de data-structuren van WAQUA en TRIWAQ, met name door WAQUA te baseren op de TRIWAQ-arrays.

We verwachten dat de tweede stap in WAQUA beperkte gevolgen voor de performance en rekenresultaten heeft. Het resultaat hiervan is een versie die kan worden opgenomen in beheer en onderhoud. Het aantal regels code verandert nog niet wezenlijk.

Deze twee onderdelen worden in de volgende twee paragrafen verder uitgewerkt.

4.2 Vermijden gebruik van `mnmaxk`-arrays

In het verleden werd er in WAQUA met indirecte adressering via `lgrid` gerekend terwijl er in TRIWAQ “full box” arrays werden gebruikt. In het droogvalproject van 2004 zijn WAQUA en TRIWAQ op dit punt geuniformeerd door het verwijderen van `lgrid` uit de rekenroutines [3]. Op de SDS-file (en in Kalman state-vectoren) wordt echter nog steeds (terecht) met indirecte adressering via `lgrid` gewerkt.

Een ongewenst gevolg van deze gang van zaken is dat er in het programma WAQPRO van veel arrays twee varianten bestaan: het array dat van de SDS-file afkomstig is of daarnaar weggeschreven wordt, en het fullbox array dat in de berekeningen wordt gebruikt. In naamgeving worden deze arrays onderscheiden door een enkele letter, bijvoorbeeld arrays `sep` en `sepf`. Het zou de leesbaarheid van de code sterk verbeteren wanneer de SDS-arrays volledig worden verstoppt. Hiervoor kan een kleine schil rond de lees- en schrijfroutines van SIMONA worden ontworpen.

Het lijkt ons nuttig om deze routines nu als eerste te ontwikkelen en in te voeren. Hiermee kan het data-model van WAQUA/TRIWAQ (alle `infor`-pointers in array `intgda`) in een keer flink worden opgeschoond. De mooie namen voor variabelen zonder “f” aan het eind kunnen dan voortaan voor de fullbox-arrays worden gebruikt. En in dit project moeten we toch alle koproutines aanpassen en kunnen dan gelijk de nieuwe namen worden ingevoerd.

De benodigde stappen hiervoor zijn als volgt:

1. Programmeren van een schilletje rond het maken van fullbox-arrays:

```
call wasrdf(nmax, mmax, 1, 1, mnmaxk, defval, iadlnd,  
           'SOLUTION_FLOW_SEP', ilgrid, intgda(252,2), name1(1:ilen))
```

Dit maakt een fullbox-array aan, zet de `infor`-index in `intgda`, haalt de opgegeven karakteristiek op, en expandeert deze met `wag1fr` naar het nieuwe array. `ilgrid` is de `infor`-index van het `lgrid`-array.

2. Programmeren van een schilletje rond het wegschrijven van fullbox-arrays:

```
call waswrf(nmax, mmax, 1, 1, mnmaxk, intgda(252,2), ilgrid,  
           'SOLUTION_FLOW_SEP', name1(1:ilen))
```

Dit activeert het fullbox-array en `mnmaxk`-array, en comprimeert de data in het eerste naar het tweede array met `wagf1r`.

3. Vervangen van het aanmaken en vullen van fullbox-arrays in het initialisatiegedeelte door aanroepen van `wasrdf`.
4. Vervangen van het vullen en wegschrijven van `mnmaxk`-arrays in het outputgedeelte door aanroepen van `waswrf`.
5. Nalopen van heel `intgda` (`awasin.i`), bepalen wat `infor`-indices van `mnmaxk`-arrays zijn, kijken of deze entries nog worden gebruikt. Zo niet: verwijderen van deze entries uit `intgda`.
6. Maken van verdere aanpassingen om het gebruik van `mnmaxk`-arrays te verminderen en verdere entries te kunnen weghalen uit `intgda`.
7. In de hele code vervangen van namen met “f” aan het einde door namen zonder f.

4.3 Uniformeren van arrays

In deze stap worden de data-structuren van WAQUA en TRIWAQ gelijkgetrokken en worden er wat algemene verbeteringen doorgevoerd.

In deze stap beginnen we met het doorvoeren van wat verbeteringen aan de data-structuur in TRIWAQ:

1. Aanmaken van arrays `hksp/hksh`, `hkup/hkuh` en `hkvp/hkvh` als semi-permanente arrays in TRIWAQ.

Deze arrays worden momenteel steeds opnieuw als werkarrays aangemaakt en herberekend. In eerste instantie kunnen de werkarrays worden vervangen door semi-permanente arrays. Vervolgens kan worden gekeken op welke plekken de arrays al valide inhoud bevatten en niet opnieuw hoeven te worden bepaald.

2. Invoeren van “aliassen” in TRIWAQ: zorgen dat `sepf` en `seh` niet apart worden opgeslagen, maar verwijzen naar de bovenste laag van arrays `zksp` en `zksh`. Hiervoor worden op de posities van `sepf` en `seh` in `intgda` de INFOR-pointers van `zksp` en `zksh` gezet.

Deze stap is nodig om de performance van WAQUA op peil te houden. Een belangrijk verschil tussen WAQUA en TRIWAQ is dat er in WAQUA minder tussen verschillende arrays wordt gekopieerd.

3. Geven van een meer permanente status aan de arrays `hu/hv`, `umean/vmean`, `qx/qy` die gaan over de gehele waterkolom.

Deze arrays zullen een eigen positie krijgen in `intgda`. Dit in plaats van de huidige situatie waarin de pointer naar sommige 2D arrays van WAQUA op dezelfde plek wordt opgeslagen als de pointer naar 3D arrays in TRIWAQ.

4. In geval van een laag: laten samenvallen van arrays per laag en arrays voor de totale waterkolom. Dit regelen we door op twee posities in `intgda` dezelfde INFOR-pointer op te slaan.

- `qx` \equiv `qxk`, `qy` \equiv `qyk`.

In WAQUA heeft array `qy` nu al hetzelfde adres als `qxk`. Er is geen `qx`-array?

- `umean` \equiv `up` of `umean` \equiv `uh`, `vmean` \equiv `vp` of `vmean` \equiv `vh`.

De arrays `umean` en `vmean` worden nog heel weinig gebruikt, maar zijn later van belang voor het generaliseren van WAQUA-functionaliteit.

- `kfup` \equiv `kfbu` of `kfuh` \equiv `kfbu`, `kfvp` \equiv `kfbv` of `kfvh` \equiv `kfbv`.

De arrays `kfbu` en `kfbv` hoeven niet te worden aangemaakt in WAQUA of in TRIWAQ met een laag. [Deze arrays komen voor in de volgende routines:](#)

- [wasfgd](#) en [trsfin](#);

[De arrays worden aangemaakt en gevuld met de waarde -1. Dit kan worden overgeslagen in geval van 2D berekening.](#)

- `trscue`

In geval van 2D berekening kan het overeenkomstige schotjes-array worden meegegeven, door te stellen

`ipkfbv=ipkfv` in `wasspu`, en
`ipkfbu=ipkfuh` in `wasspv`.

De routine `trsbar` doet een aanpassing aan `kfbu`. Deze aanpassing kan er in de huidige versie toe leiden dat de arrays `kfbu` en `kfu` verschillen. Door de juiste droogval- en onderloopcriteria kan worden voorkomen dat dit gebeurt. Dit is in WAQUA al het geval. Daarna hoeft `trsbar` niet meer te worden aangepast.

- `trssuw`

In geval van 2D berekening kan het overeenkomstige schotjes-array worden meegegeven, door te stellen

`ipkfbu=ipkfuh` in `wasspu`, en
`ipkfbv=ipkfup` in `wasspv`.

- `trsdif`

In geval van 2D berekening kan het overeenkomstige schotjes-array worden meegegeven, door te stellen

`ipkfbu=ipkfuh`, `ipkfbv=ipkfvh` in `wastru`, en
`ipkfbu=ipkfup`, `ipkfbv=ipkfv` in `wastrv`.

- `hu` \equiv `hukh` of `hu` \equiv `hukp`, `hv` \equiv `hvk` of `hv` \equiv `hvkp`. Hierbij moet worden uitgezocht op welke tijdsniveaus de arrays `hu` en `hv` zijn gedefinieerd.

5. Introductie van WAQUA-arrays in TRIWAQ:

In TRIWAQ moeten waarschijnlijk de arrays die in WAQUA voor barriers worden gebruikt (`lgubar`, `basepa`, ...) worden geïntroduceerd.

- `lgubar`, `lgvbar`
- `basepa`, `basepb`, ...

Daarna maken we de data-structuur in WAQUA hetzelfde als die in TRIWAQ:

6. Aanmaken van arrays met 3D-informatie in WAQUA-berekeningen.

Dit gaat enerzijds om verticale snelheden en grootheden van het turbulentiemodel (`w`, `vicow`, `rtur`). Anderzijds gaat het om de posities van laaginterface en de laagdiktes (`zksp/zksh`, `zkup/zkuh`, `zkvp/zkvh`, de bovengenoemde `hk`-arrays, en bijvoorbeeld `hlay` en `indlay`). Tenslotte worden er voor barriers schotjes per laag gebruikt (`kfbu/v`). Deze arrays worden in `wasfgd` aangemaakt met conditie “alleen voor TRIWAQ”.

Een aantal van deze arrays wordt niet gebruikt wanneer `kmax=1`, maar worden geïntroduceerd ter vereenvoudiging van de programmatuur. Het alternatief is om deze arrays volledig te gaan omzeilen in WAQUA en TRIWAQ-berekeningen met 1 laag. Dit gaat echter wel om veel plekken in de programmatuur, onder andere bij het bepalen van time-histories en in het Kalman-filter. Het niet aanmaken van deze arrays (uitsparen geheugenruimte) leidt daarom tot extra complexiteit in de programmatuur.

7. Overall vervangen van het gebruik van `dps`, `dpu` en `dpv` door het gebruik van de onderste laag uit de `zk`-arrays.
8. Invoeren van de nieuwe arrays in rekenroutines van WAQUA, met name de `zk`- en `hk`-arrays in `wassuc`.

Er zal voor worden gezorgd dat `hu` en `hv` een alias worden van `hkuh` en `hkvp`.

Waar nu `hu` wordt berekend moet eerst de waterstand in het u -punt worden bepaald en opgeslagen in het array `zkuh`, en kan daarna de totale doorstroomhoogte `hu` worden bepaald.

Deze arrays worden gebruikt in

- `trsfm` (arrays vullen)
- `trscf`, lijkt op `wagcof`.
- `trscot`, lijkt op `wagcot`.
- `trscpw`, lijkt op `wasfit`.
- `trscue`, lijkt op `wasuxc`.
- `trsdif`, lijkt op `wasdfc`.
- `trsfm`, lijkt op `wasfit`.
- `trsim`, Lagrangiaanse integratie, bestaat nog niet
- `trslag`, Lagrangiaanse integratie, bestaat nog niet voor WAQUA
- `trschz`, lijkt met `trsczw` samen op `wasvlu` en `wasvlu`.
- `waskv1` bevat een WAQUA-deel en een TRIWAQ-deel
- `waskch` bevat een WAQUA-deel en een TRIWAQ-deel
- `waseva` bevat een WAQUA-deel en een TRIWAQ-deel
- `wasmaf` (uitvoer naar LAYER_INTERFACES)
- `trsmea` (berekening `umean/vmean`)
- `trssuw` lijkt op `wassuc`.
- `trstur` turbulentiemodel, alleen in TRIWAQ
- `trsviz` turbulentiemodel, alleen in TRIWAQ
- `waskg5` bevat een WAQUA-deel en een TRIWAQ-deel
- `wasust`, user transport (door gebruiker in te vullen)
- `waszks`, `zks` aanmaken voor WAQUA, voor `wasust`

4.4 Tijdschatting

Voor het verder verstoppen van `mnmaxk`-arrays verwachten we de volgende tijd nodig te hebben:

Stap	Omschrijving	Inspanning
1, 3	wasrdf : lezen fullbox-arrays	8 uur
2, 4	waswrf : schrijven fullbox-arrays	8 uur
5, 6	opschonen van intgda , verdere aanpassingen	8 uur
7.	aanpassen variabelenamen, uitleg	12 uur
	projectleiding, onvoorzien 20%	8 uur
TOTAAL		44 uur

Het uniformeren van de data-structuren van WAQUA en TRIWAQ is wat moeilijker in te schatten. Globaal rekenen we hiervoor op de volgende inspanning:

Stap	Omschrijving	Inspanning
1.	aanmaken/invoeren hk -arrays	1 dag
2.	invoeren aliassen sepf/seh	2 dagen
3.	aanmaken/invoeren arrays hele waterkolom	2 dagen
4.	uitsparen 3D arrays in 2D berekeningen	2 dagen
5.	aanmaken WAQUA barrier-arrays in TRIWAQ	1 dag
6.	aanmaken 3D arrays in TRIWAQ	1 dag
7.	vervangen dps/u/v door zks/u/v	1 dag
8.	invoeren TRIWAQ-arrays in WAQUA	6 dagen
	onvoorzien 20%	2 dagen
	projectleiding 10%	1 dag
TOTAAL		18 dagen

Hoofdstuk 5

Uniformering WAQUA/TRIWAQ: migreren eenvoudige functionaliteit

In deze stap worden berekeningen van WAQUA en TRIWAQ samengevoegd/geuniformeerd voor zover dat geen gevolgen heeft voor de rekenresultaten. Enerzijds betreft dit het beschikbaar maken van functionaliteit uit het ene model in het andere. Anderzijds betreft dit het samenvoegen van routines die niet tot de echte rekenkern behoren en die in aparte varianten voor WAQUA en TRIWAQ bestaan.

5.1 Beschikbaar maken van functionaliteit in WAQUA en TRIWAQ

Enerzijds zal functionaliteit die nu in een van de twee modellen beschikbaar is en gemakkelijk kan worden generaliseerd ook voor het andere model beschikbaar worden gemaakt. Dit betreft:

1. In TRIWAQ-berekeningen kan harmonische analyse van de vertikaal gemiddelde snelheid `umean/vmean` mogelijk worden gemaakt. Dit is een gemakkelijke aanpassing in een koproutine en verwijderen van een controle uit `WAQPRE`. De meeste tijd gaat nog zitten in het testen van de aanpassingen.

Deze aanpassing is verbazend eenvoudig. Hij vergt volgens mij niet meer dan 3 acties:

- *vervang een gedeelte van `wasofd` door een call naar `wasofd`: dan hoeven aanpassingen in `wasbti` nog maar een keer.*
- *gebruik `umean` en `vmean` in de aanroep naar `wasuwl` in `wasbti`.*
- *Haal de foutmelding weg uit `wap072` die harmonische analyse verbiedt in TRIWAQ*

Dit kan allemaal in 1 uur.

Hierna moet getest worden. Dat is veel lastiger: twee modellen met Harmonische analyse moet gevonden worden (bijvoorbeeld in de testbank): een met en een zonder splitting.

Hiervan wordt een TRIWAQ model met een laag gemaakt en een met twee lagen. Daarna wordt de harmonische analyse van TRIWAQ vergeleken met die van WAQUA. Tot slot wordt er voor gezorgd dat de modellen in de testbank komen en dat de harmonische uitvoer wordt meegenomen in de controles.

Om dit allemaal te doen is zeker 6 uur nodig.

2. In TRIWAQ-berekeningen met meerdere lagen kunnen QH-openingen worden toegestaan door de vertikaal gemiddelde snelheid `umean/vmean` te gebruiken. Ook dit is een eenvoudige aanpassing.

Ook hiervoor is maar heel weinig programmeren nodig:

- `wasfqh` gebruik `umean` en `vmean` in plaats van `up` en `vp`.
- `wap058`: verwijder foutmelding 2532.

Daarna moet er getest worden. Dit moet op dezelfde manier als bij harmonische analyse, maar is eenvoudiger omdat de uitvoer al wordt vergeleken in de testbank. Er moet dus alleen een QH-model worden genomen, waarvan een meerlagenmodel wordt gemaakt.

Programmeren: 1 uur, testen: 4 uur.

3. De berekening van Lagrangiaanse tijdsintegralen wordt in WAQUA mogelijk gemaakt. Zodra de data-structuur van WAQUA op die van TRIWAQ is afgestemd vergt dit alleen nog het uitschakelen van bepaalde controles en het wegschrijven van resultaten in WAQUA-runs.

Voor de berekening van Lagrangiaanse integralen worden arrays gebruikt die nu nog niet beschikbaar zijn in WAQUA-berekeningen. Op het moment dat we deze uitbreiding gaan doorvoeren zijn al deze integralen wel beschikbaar. Daarom hoeven alleen de verbodsbepalingen in `wap072` en `wasfbr` te worden opgeheven, en moet `waswti` een klein beetje aangepast worden zodat de resultaten ook op SDS-file komen in WAQUA berekeningen.

Als test moet er een Lagrangiaanse som worden gevonden in de testbank (die zijn er, dat weet ik gewoon). Daarvan moeten een test met 1 laag in TRIWAQ en een WAQUA-test worden gemaakt. Daarna kunnen de Lagrangiaanse arrays met elkaar worden vergeleken.

De WAQUA-test moet aan de testbank worden toegevoegd.

Ik weet niet of het gewenst is dat `slib3d` het doet met WAQUA-uitvoer. Als dat zo is, moet er nog meer getest worden.

Programmeren: 1 uur.

Testen WAQUA/TRIWAQ: 4 uur

Testen SLIB3D: 4 uur

4. De berekening van incrementele uitvoer t.b.v. animaties blijkt al in TRIWAQ te worden toegestaan. Dit punt uit eerdere plannen komt dus te vervallen.

Incrementele uitvoer is reeds beschikbaar voor TRIWAQ. Overigens is wel de indentatie in routine `wap174` helemaal in de war.

5. De berekening van maximale waarden gedurende de simulatie is momenteel in TRIWAQ alleen toegestaan wanneer er slechts een laag wordt gebruikt. We stellen voor om in deze berekening de verticaal gemiddelde snelheid te gebruiken en deze optie met meer lagen toe te staan. Dit is een kleine ingreep.

Volgens mij wordt maximum value berekening wel toegestaan in TRIWAQ, als `kmax`. eq. 1. Wel wordt in `wap174` een warning gegeven en wordt de invoer genegeerd. Dit moet worden weggehaald.

Verder moet de routine `wascmx` de routines `umean` en `vmean` gaan gebruiken in plaats van `upf` en `vpf`.

Ook in de routine `wascmx` is de indentatie in de war.

Testen: door een `maxval`-test te nemen en uit te breiden tot een 2-laags test. De resultaten moeten sterk lijken op de 1-laags test.

De testbank moet waarschijnlijk zo worden aangepast dat de `maxval`-resultaten worden vergeleken.

Maxval kan in een groter aantal testen worden toegevoegd, zodat deze functionaliteit voortaan grondig wordt getest.

Programmeren: 1 uur

Testen: 4 uur

De volgende stap wordt nog niet uitgevoerd:

- Het temperatuurmodel dat in WAQUA zit zou ook in TRIWAQ beschikbaar kunnen worden gemaakt.

Uitvoeren hiervan is niet opportuun omdat het temperatuurmodel in een ander kader sterk wordt aangepast.

5.2 Mergen van eenvoudige rekenroutines

Anderzijds is er een groot aantal routines die niet tot de echte rekenkern behoren en die in aparte varianten voor WAQUA en TRIWAQ bestaan. Deze routines kunnen gemakkelijk worden samengevoegd:

1. Samenvoegen van routines `wagcof`, `trscof` en `trpcof`, berekenen time-histories FLOW. Deze kunnen worden samengevoegd tot een enkele routine `trgcof` in WAQGEN. Dit leidt tot een minimaal verschil op het starttijdstip, omdat `trpcof` rotatie net iets anders berekent dan `trscof`.

Er zijn momenteel 3 routines die (te?) veel op elkaar lijken: `trpcof`, `trscof` en `wagcof`. Deze verschillen op de volgende manier van `trscof`:

- `trpcof`:
 - gebruikt `mnmaxk`-arrays*
 - berekening van rotatie gaat net anders.*

- **wagcof**:
alleen 2D-berekeningen.

*De routine **wagcof** wordt alleen gebruikt door **waqpre** en **waqpro** (niet door **sds2mat** oid).*

Ik heb de routines al enigszins aangepast in deze report-directory, zonder hun werking te veranderen.

*Het lijkt me mogelijk om de routine **trscot** te verplaatsen naar **waqgen** en in alle gevallen te gebruiken. Wel moet **WAQPRE** dan aansturen met **fullbox**-arrays. Dit is in de aanroep van **wagcof** al gedaan.*

*Ik heb er natuurlijk al wat tijd in zitten, maar ik denk dat dit in 2 uur kan. Edwin: vind ik optimistisch wg. **mnmaxk** en wg. goed testen gevolgen rotatie.*

2. Samenvoegen van routines **wagcot**, **trscot** en **trpcot**, berekenen time-histories TRANSPORT. Deze routines worden samengevoegd tot een enkele routine **trgcot** in **WAQGEN**, analoog aan de routines voor FLOW hierboven.

*Ook de routine **wagcot** wordt enkel aangeroepen vanuit **waqpre** en **waqpro**.*

*Ook hier zijn drie routines: **trpcot**, **trscot** en **wagcot**.*

Ik veronderstel dat de situatie bijna het zelfde is als hierboven.

Daarom denk ik dat ik deze werkzaamheden in 4 uur kan (2 uur meer dan hierboven, omdat ik daar immers al de routines heb voorbereid)

3. Samenvoegen van routines **waspcf** en **trspcf**, printen van computed data (maps) voor FLOW naar de report-file. Hierbij moeten kleine verschillen in de report-file tussen **WAQUA** en **TRIWAQ** met 1 laag worden geëlimineerd. Bijvoorbeeld drukt **TRIWAQ** bij gebruik van 1 laag nog steeds het laagnummer af en **WAQUA** niet. Hierdoor zullen er verschillen ontstaan in de precieze layout van de report-file ten opzichte van de huidige versie.

Ook hier heb ik al een beetje geoefend. De routines zijn al grotendeels met elkaar in overeenstemming. Verschillen zijn:

- ***TRIWAQ** drukt laagnummers af.*
- ***TRIWAQ** drukt reals af voor een klein aantal variabelen*
- *De bottom roughness heet **CMANU** in **WAQUA** en **ROUGHU** in **TRIWAQ***
- ***TRIWAQ** drukt extra variabelen af:*
 - ***W**-snelheden*
 - *vertical viscosities*
 - *bottom stress velocities*
 - *Richardson numbers*
- *Er zitten volgens mij fouten in **waspcf**, want soms wordt **waspcr** aangeroepen met een **fullbox** array, in plaats van een **(nmax,mmax)**-array*

Als al dit verschillende gedrag (behalve de fouten natuurlijk) moet worden behouden, wordt dit een beetje vervelende functie.

Toch kan dit allemaal in weinig tijd gedaan worden: ik denk in 2 uur.

4. Samenvoegen van routines `waspc` en `trspc`, printen van computed data (maps) voor TRANSPORT naar de report-file. Deze routines kunnen eenvoudig worden gecombineerd. Het belangrijkste verschil is dat TRIWAQ verticale diffusie-coëfficiënten afdruckt en WAQUA niet. Deze print zal worden uitgezet in TRIWAQ-berekeningen met 1 laag.

Het gaat hier om heel eenvoudige functies. Ik denk dat deze functies in 1 uur in elkaar geschoven kunnen worden. TRIWAQ drukt verticale diffusie-coëfficiënten af, WAQUA niet

5. Samenvoegen van routines `wasphf` en `trspfhf`, printen van history data voor FLOW naar de report-file. Een verschil tussen deze routines is de uitvoer die WAQUA voor barriers wegschrijft. Deze uitvoer willen we door een aparte routine `wasphb` laten schrijven. Verder drukt TRIWAQ ook grootheden per laag af, deze zullen bij gebruik van 1 laag worden uitgezet.

Ik heb de routines ook weer voorbereid. De volgende verschillen:

- WAQUA heeft uitgebreidere FORMATS
- WAQUA drukt ook barrier-dingen af
- TRIWAQ drukt behalve stroomsterkte `zcur`, ook de componenten `zcuro`, `zcurv` en `zcurw` af, per laag.

Het is denk ik een goede manier om twee routines te maken: eentje zonder barriers en een met.

Het samenvoegen van deze routines kost dan denk ik 3 uur.

6. Samenvoegen van routines `wasph` en `trspht`, printen van history data voor TRANSPORT naar de report-file. Hiervoor geldt m.m. hetzelfde als hierboven voor `wasphf` en `trspfhf`.

In deze routines verschillen geloof ik alleen de FORMATS, die in WAQUA uitgebreider zijn, en de lagen die alleen in TRIWAQ worden afgedrukt.

Het samenvoegen van deze routines kost dan denk ik 2 uur.

7. Samenvoegen van routines `wasei2` en `wasei3` voor het berekenen van Euleriaanse tijdsintegralen in respectievelijk WAQUA en TRIWAQ. Het belangrijkste verschil hiertussen is dat WAQUA arrays SEPNOW en DEPINT gebruikt en wegschrijft naar de SDS-file, terwijl TRIWAQ array INTEGRAL_FLOW_ZK gebruikt en schrijft.

We stellen voor om `wasei3` uit te breiden zodat SEPNOW en DEPINT in alle gevallen naast INTEGRAL_FLOW_ZK wordt aangemaakt en op de SDS-file gezet. Daarna kan deze routine ook in WAQUA worden gebruikt.

De WAQUA-routines kunnen dan vervallen.

Dit kan denk ik in 6 uur.

Onderscheid tussen WAQUA en TRIWAQ waar vooralsnog niets aan wordt gedaan is:

- In de dynamische barriesturing is niet duidelijk of in WAQUA en TRIWAQ precies dezelfde formules worden gebruikt (precieze tijdsniveaus).
- In de Chezy-arrays worden verschillende waarden voor het landpunt gebruikt (0 en 9999.).
- Vertikale verfijning is alleen mogelijk in TRIWAQ (ook subdomeinen met 1 laag) en niet in WAQUA.

5.3 Aandachtspunten

- Er wordt gevraagd om een inventarisatie te maken van de functionaliteiten (zoals display e.d.) in TRIWAQ die in de afgelopen jaren niet gebruikt zijn.
 - Het *niet* opgeven van USE_ZEROS in Kalman berekeningen wordt niet gebruikt.
 - Van alles wat in DISPLAYS staat weet ik niet hoeveel dat gebruikt wordt.
 - Harmonische metingen en satellietmetingen kunnen niet worden verwerkt door OBS2SDS. Daarom zijn de routines in WAQPRO die dit soort metingen verwerken al jaren niet gebruikt of getest.
Moeten deze worden bewaard in afwachting van een OBS2SDS die wel dergelijke metingen kunnen verwerken, of kan deze (ongebruikte, ongeteste) functionaliteit vervallen?
 - Er wordt overwogen om CDCON te verwijderen. Dit lijkt een goed idee, ook omdat er in de handleiding staat dat de werking van andere waarden dan CDCON=0 niet betrouwbaar is.
- De aanroep van de routine HATYAN moet worden aangepast aan de TRIWAQ norm.
- De aanroep van Nikuradse in TRIWAQ zou moeten worden aangepast.
- Welke functionaliteiten komen verder in aanmerking voor uniformering?

Ik heb de hele code doorgespit en de volgedne punten gevonden:

- Kan routine `trscof` gebruik maken van `trscpw` voor de berekening van de verticale stroomsnelheden?
- De volgende routines berekenen laagdiktes: `trslav`, `trscah`, `trslv1`. Kunnen die gemerged worden?
- De onderste en bovenste laag van het array `wf` bevatten altijd nullen. Door dit te gebruiken kunnen de volgende routines efficiënter en/of eenvoudiger worden: `waskrf`, `waskpi`, `waskf1`, `waskt6`

- Door de introductie van laaginterfacearrays `zk*` kunnen vereenvoudigingen worden doorgevoerd in de routines `wasksv`, `waskv1`, `waskxp`, `waskch`, `waskcw`, `waskfu`, `waskg2`, `waskg5`, `waskgs`.
- In de routines `wassff` en `waschf` komt een verschil voor tussen TRIWAQ en WAQUA met betrekking tot de gebruikte tijdsnivo's. In `wascht` is een dergelijk verschil opgeheven. Deze verschillen kunnen worden opgeheven.

5.4 Tijdschatting

Voor het beschikbaar maken van WAQUA-functionaliteit in TRIWAQ verwachten we de volgende tijd nodig te hebben:

Stap	Omschrijving	Inspanning
1.	harmonische analyse vert.gem. snelheid	8 uur
2.	QH-openingen o.b.v. vert.gem. snelheid	8 uur
3.	Lagrangiaanse tijdsintegratie in WAQUA	8 uur
4.	incrementele uitvoer in TRIWAQ	-
5.	berekening maximale waarde vert.gem. snelheid	8 uur
	projectleiding, onvoorzien 10%	4 uur
TOTAAL		36 uur

Het uniformeren van de eenvoudige rekenroutines van WAQUA en TRIWAQ begroten we als volgt:

Stap	Omschrijving	Inspanning
1.	samenvoegen <code>wagcof</code> , <code>trscof</code> en <code>trpcof</code>	8 uur
2.	samenvoegen <code>wagcot</code> , <code>trscot</code> en <code>trpcot</code>	8 uur
3.	samenvoegen <code>waspcf</code> en <code>trspcf</code>	8 uur
4.	samenvoegen <code>waspct</code> en <code>trspct</code>	8 uur
5.	samenvoegen <code>wasphf</code> en <code>trspfhf</code>	8 uur
6.	samenvoegen <code>wasphf</code> en <code>trspht</code>	8 uur
7.	uitbreiden <code>wasei3</code> , elimineren <code>wasei2</code>	12 uur
	projectleiding, onvoorzien 20%	12 uur
TOTAAL		72 uur

Hoofdstuk 6

Uniformering WAQUA/TRIWAQ: mergen van het echte rekenhart

6.1 Inleiding

Zoals het project nu is ingedeeld (zie hoofdstuk 1) wordt het daadwerkelijk samenvoegen van de rekenroutines van WAQUA en TRIWAQ tot het laatst uitgesteld. Dit werk wordt wel voorbereid in de andere fases, met name door het uniformeren van de data-structuren. Bij het daadwerkelijk samenvoegen van rekenroutines spelen echter de performance en de rekenresultaten een belangrijke rol.

We pleiten ervoor om niet-wezenlijke verschillen in de numerieke schema's weg te poetsen ten voordele van een eenvoudigere implementatie. Op deze manier wordt een zo groot mogelijke reductie van complexiteit en aantal regels broncode bereikt zonder dat dat tot beperkingen voor de gebruiker leidt. Hierdoor worden er waarschijnlijk wel wat extra verschillen ten opzichte van de huidige rekenresultaten geïntroduceerd.

6.2 Hoofdpijnen migreren rekenroutines

In dit hoofdstuk presenteren we ons huidige plan voor de samenvoeging van de rekenroutines van WAQUA en TRIWAQ. In deze paragraaf geven we de hoofdpijnen, in de volgende paragraaf met wat meer detail. We hebben het plan intern nog verder uitgewerkt om zo veel mogelijk moeilijkheden op het spoor te komen. We verwachten echter niet dat het plan zoals het er nu staat wordt uitgevoerd. Daarvoor is de materie te complex, niet genoeg te overzien. Bovendien veranderen de rekenroutines nog door de introductie van niet-hydrostatisch rekenen, het horizontaal $k - \epsilon$ model, en de uniformering van arrays.

De voorgestelde aanpak is om de grote rekenroutines een voor een volledig te uniformeren:

1. Samenvoegen van routines voor bodemwrijving en overlaten;
2. samenvoegen van routines voor de impulsvergelijking;

3. samenvoegen van routines voor de gekoppelde impuls- en continuïteitsvergelijkingen;
4. samenvoegen van routines voor het stoftransport.

Hiervan is de eerste stap nog redelijk eenvoudig, maar zijn met name de tweede en derde stap behoorlijk complex. Bij deze stappen zullen de twee aparte implementaties eerst in meerdere stappen worden aangepast, zodat ze meer op elkaar gaan lijken. Pas als een aantal grote verschillen is geëlimineerd zullen de routines daadwerkelijk worden samengevoegd.

De afhandeling van barriers verschilt sterk tussen WAQUA en TRIWAQ, ook wanneer daarin 1 laag wordt gebruikt. Aan deze verschillen wordt nog niets gedaan. Afhankelijk van het model dat in de simulatieinvoerfile wordt geselecteerd worden ofwel de ene ofwel de andere formuleringen gebruikt.

6.3 Verdere uitwerking migreren rekenroutines

Het daadwerkelijk samenvoegen van de echte rekenroutines zal met de volgende stappen worden gedaan:

1. Samenvoegen van routines `wasvlu/wasvlu` enerzijds en `trschz/trsczw` anderzijds voor de berekening van Chezy-coëfficiënten en het energieverlies tengevolge van overlaten.
 - (a) Vergelijken en samenvoegen van de berekening van de bodemruwheid in WAQUA en TRIWAQ.
 - (b) Vergelijken en samenvoegen van de berekeningen voor overlaten in WAQUA en TRIWAQ. Hier zit de implementatie van WAQUA overzichtelijker in elkaar (aparte routines voor u - en v -richting). Indien/voor zover mogelijk zal daarvan worden uitgegaan of bij aangesloten.
 - *Verwijderen van functionaliteit die niet met overlaten samenhangt uit `wasvlu` en `wasvlu`.*
 - *Aanroep naar `trschz` in geval van WAQUA.*
 - *Gebruik van `wagcvl` in `trschz`*
 - *Vervangen van de call naar `trsczw` door een call naar `wasvlu` of `wasvlu`.*

Ik denk dat alles bij elkaar in een halve dag kan (inclusief testen).

2. Zelfstandig opgeloste impulsvergelijking, subroutines `wasuxc` en `trscue/trsjam`:
 - (a) Verwijderen van de viscositeitskruistermen uit WAQUA en TRIWAQ, om de volgende stap sterk te vergemakkelijken.
weghalen is makkelijk. Wel moet een error in WAQPRE worden toegevoegd. Alles bij elkaar twee uur.

- (b) Reorganiseren van subroutine `wasuxc` in een gedeelte waarin coëfficiënten worden bepaald en een gedeelte waarin de iteratiemethode wordt uitgevoerd.

*Hiervoor moeten de werkarrays `aa`, `bb`, `cc`, `dd` omgezet worden naar `fullbox-arrays`. Ook moeten er `coefficientenarrays` worden gemaakt voor de waarden uit andere roosterlijnen. Deze heten dan `b[ud]*x`, net als in `trscue`.*

Hoe deze waarden moeten worden gevuld kan worden gevonden door in `wasuxc` te zoeken naar `up`.

Vervolgens moet loop 1000 worden gekopieerd. Uit de eerste loop wordt het oplosgebouwen weggehaald (alles vanaf iets voor loop 800). Deze eerste kopie wordt buiten de iteratie gehaald.

*In de tweede loop wordt het modelvergelijkingen-gedeelte vervangen door het invullen van `bux*up` etcetera. Het `vegen` moet een klein beetje anders worden uitgevoerd zodat het array `cc` niet wordt aangepast. Door het array `bb` te vervangen door `bbi` kan het allemaal nog een klein beetje efficiënter worden gemaakt.*

- *Werkarrays maken, doorgeven: 1 uur*
- *Werkarrays vullen: 2 uur*
- *Loop 1000 voor iteratie: 2 uur*
- *Loop 1000 in iteratie: 1 uur*
- *Testen of hij het nog doet: 2 uur*

Samen 8 uur.

- (c) Instelbaar maken van het stopcriterium voor de impulsvergelijking in TRIWAQ.

De waarden `epsit` en `maxit` worden in `trscue` hard ingesteld op `epsit=1e-4` en `maxit=100`.

Voor `wasuxc` worden de waarden `epsvel=reagda(10,2)=rconta(6)` en `iter2 = intgda(153,2) = icontb(3)` gebruikt. Deze waarden kunnen natuurlijk ook worden doorgegeven naar `trscue`, maar worden ze wel gevuld met de juiste waarden?

De waarde `iter2` wordt gezet in `wap072`. De default waarde hiervoor is 8. Deze waarde moet denk ik op 100 gezet worden voor TRIWAQ. Als een andere waarde dan 100 is opgegeven, moet denk ik een warning worden gegeven omdat de invoer tot nu werd genegeerd maar nu niet meer.

De waarde `epsvel` wordt gezet in `wap073`. Deze waarde wordt altijd gezet. Ook hier moet een warning komen als niet `1e-4` wordt gekozen, omdat de invoer tot nu toe werd genegeerd en nu niet meer.

Ik denk dat ik alles, na deze nazoeking, in 2 uur kan doen.

- (d) Vergelijken van het aantal iteraties voor de impulsvergelijking in WAQUA en TRIWAQ met 1 laag voor meerdere testproblemen. Als de methode van TRIWAQ snel genoeg is en als de resultaten bij gebruik van `ITERMOM=2` niet veel verschillen dan kan de methode van WAQUA komen te vervallen.

Ik denk dat het het eenvoudigste is om eerst beide methoden aan te bieden, daarna in een aparte activiteit te gaan testen of beide nodig zijn, en daarna eventueel een methode op te heffen. Een dergelijke test kan denk ik wel veel tijd vergen, enkele dagen, denk ik.

- (e) Introduceren van een aparte subroutine voor de iteratiemethode voor de impulsvergelijking in WAQUA.

*Het gaat hier om het verhuizen van een gedeelte van **wasuxc** naar een nieuwe routine. Dat hoeft niet veel tijd te kosten: 2 uur.*

- (f) Stap voor stap harmoniseren van de discretisaties voor de impulsvergelijking in WAQUA en TRIWAQ.

*We kunnen denk ik beginnen met een aanroep naar **trsjam** in te voeren (2 solvers beschikbaar). Hiermee test je gelijk of het data-model van **wasuxc** en **trscue** overeen komen. Daarna kun je de aanroep naar de WAQUA-solver toevoegen aan **trscue**.*

Hiervoor is denk ik 4 uur nodig.

Vervolgens de calls zodanig herorganiseren dat de parameterlijsten bijna het zelfde zijn. Ontbrekende arrays toevoegen (dat moet inmiddels makkelijk zijn, vanwege de eerdere activiteiten).

Ook hiervoor is denk ik 4 uur nodig.

Daarna de berekeningen gelijktrekken. De verschillende formules voor advectie zouden al makkelijk toe te voegen moeten zijn, door de activiteiten met betrekking tot niet-hydrostatisch rekenen.

Hiervoor is denk ik 12 uur nodig.

*Na verloop van tijd is de routine **wasuxc** gelijk aan **trscue** voor **kmax=1**. Dan kan de call naar **wasuxc** worden vervangen door die naar **trscue**.*

Dat hoort echter bij de volgende activiteit.

- (g) Mergen van subroutines **wasuxc** en **trscue**.

*Nu de calls aan elkaar gelijk zijn, kan de call naar **wasuxc** worden vervangen door die naar **trscue**. Deze verandering kost haast geen tijd. Er staan daarna wel twee calls naar **trscue**; een voor WAQUA en een voor TRIWAQ. Dat laten we even zo, omdat het mooi maken van de koproutines later toch nog komt.*

*Het mergen van **trscue** en **wasuxc** zal dus 24 uur gaan kosten.*

Merk op dat op elk moment getest kan worden of hij het nog doet.

Ik heb hierbij niet geteld het uitzoeken of meerdere oplos-strategiën nodig zijn, wat denk ik wel 3 dagen kan kosten.

3. Continuïteitsvergelijking en impulsvergelijking, subroutines **wassuc** en **trssuw** en onderliggende routines:

- (a) Verwijderen van de “compatibility mode” uit subroutine **wasdry** voor droogvalcontroles.

*De routine **wasdry** bevat compatibility modes om oude manieren van droogvalcontrole terug te kunnen krijgen. Deze compatibility modes worden verwijderd. De compatibility mode kan niet door de gebruiker worden geselecteerd dus hoeft ergeen warning te komen.*

Ik kan dat in 1 uur doen.

- (b) Invoeren van het stopcriterium per rij in TRIWAQ.

Hiervoor moeten de arrays `ianthq`, `iaqngb`, `inext`, `inextg` en `idryv` worden aangemaakt.

In het iteratieproces, loop 1000 vanaf loop 2220, moeten de `nmloop`-loops worden vervangen door `irogeo`-loops, omdat alleen zo de niet-actieve rijen kunnen worden vermeden.

De routines `trsrv1`, `trsrv2` en `trsdry` moeten worden verrijkt met `ianthq`. De routine `trsdry` moet ook de bewerkingen met `idryv` overnemen uit `wasdry`.

De call naar `trstrd` moet worden vervangen door `wastrd`, want deze heeft de `ianthq`-functionaliteit al.

De routine `trslav` kan niet zo makkelijk worden aangepast. Deze laten we maar eerst zo. Kan later worden veranderd.

Ik denk dat ik dit kan in 8 uur. Daarna zou ik nog eens 8 uur willen besteden aan testen.

- (c) Mergen van subroutines `wastrd` en `trstrd` voor tridiagonale stelsels per rij.

In de vorige activiteit is `trstrd` reeds opgeheven.

- (d) Mergen van subroutines `wasdry` en `trsdry` voor droogvalcontroles.

De routine `trsdry` bevat nu nog een klein aantal extra dingen, die niet voorkomen in `wasdry`. Het gaat om `1:kmax`-loopjes. Deze moeten worden ingepast in de routine `wasdry`, zodat deze routines gelijk worden.

Ik denk dat ik dat in 2 uur kan.

- (e) Verplaatsen van de discretisaties van de impulsvergelijking van `wassuc` naar een nieuwe routine `wasumo`.

Een groot gedeelte van de routine `wassuc` wordt verplaatst naar de nieuwe routine `wasumo`. Deze routine gaat alleen maar bestaan tijdens dit project: aan het einde van deze melding is deze routine vervangen door `trsumo`.

Ik denk dat ik dat in 2 uur kan

- (f) Stap voor stap harmoniseren van kleine verschillen in implementatie en discretisaties tussen `wasumo` en `trsumo`.

Eerst worden de routines zodanig aangepast dat de parameterlijsten gelijk zijn.

Ik denk dat ik dat in 3 uur kan.

Vervolgens de berekeningen. Hiervoor moet ruim de tijd genomen worden. Als het goed is, kan de advection al gemakkelijk worden aangepast, omdat in de activiteit voor niet-hydrostatisch rekenen al een generieke implementatie is gemaakt voor advectiones.

Ik denk dat ik hiervoor 8 uur nodig zal hebben.

Aan het einde zijn `wasumo` en `trsumo` aan elkaar gelijk: de routine `wasumo` wordt opgeheven.

- (g) Harmoniseren en mergen van subroutines `wasrv1/wasrv2` en `trsrv1/trsrv2` voor de randvoorwaarden.

De routines `trsrv2` en `wasrv2` lijken al enorm op elkaar, en wanneer deze activiteit gaat beginnen nog meer (vanwege het gebruik van de onderste laag van `zk` in plaats van `dp*` in WAQUA en vanwege de invoering van `ianthq` in TRIWAQ).*

Daarom denk ik dat ik deze twee routines kan mergen in 1 uur.

De routines `trsrv1` en `wasrv1` lijken nog helemaal niet op elkaar. Daar zal dus meer tijd voor nodig zijn: Daarom denk ik dat ik deze twee routines kan mergen in 4 uur.

- (h) Harmoniseren van de berekening van `tetau/v` voor de upwindmethode voor de natte doorstroomoppervlakken.

De parameter `tetau` wordt berekend in `wasshu` (WAQUA) en in `trstet` (TRIWAQ).

In de routine `wasshu` komt de aanpassing voor barriers extra voor. Deze moet worden toegevoegd aan `trstet`.

De routine `wasshu` gebruikt nu nog het array `dpu` om de doorstroomhoogte te bepalen. Dat is voor de aanvang van deze berekening al vervangen door de onderste laag van `zku`.

De routine `wasshu` geeft bovendien ook de variabele `hu` terug. Dat kan in TRIWAQ eigenlijk ook geen kwaad: ik denk dat `wasshu` in TRIWAQ in gebruik kan worden genomen (zonder aanpassingen).

Ik denk dus dat de routine `wasshu` de routine `trstet` op dit moment kan vervangen. Hiervoor is denk ik 4 uur nodig.

- (i) Aanpassen van de controle op stabiliteit in TRIWAQ naar voorbeeld van de implementatie die in WAQUA wordt gebruikt.

De stabiliteitscontrole in `trssuw` en `wassuc` wordt uitgevoerd met behulp van de variabele `instbl`. Deze variabele wordt een heel klein beetje anders verwerkt in WAQUA dan in TRIWAQ: in WAQUA wordt ook net buiten het eigen domein gecontroleerd (loop index set `own-wlp-xdir` in plaats van `own-wlp`). In TRIWAQ gaat `instbl` mee met `waschk`, terwijl deze in WAQUA apart gecocigld wordt. Ook maakt WAQUA zelf een foutmelding in geval van instabiliteit. Ik denk dat TRIWAQ dat over laat aan `waschk`.

Het gaat om heel kleine dingetjes: ik denk dat dat in 1 uur kan.

- (j) Geschikt maken van subroutine `trssuw` van TRIWAQ voor WAQUA-berekeningen. Met name ten aanzien van barriers.

Hiervoor moet ook weer voldoende tijd worden genomen. De call naar `wassbc` wordt in elke iteratie herhaald. Deze moet dus worden toegevoegd aan `trssuw`.

De call naar `trsbar` staat in `trsumo`, dus die is al verwerkt in de aanpak van `wasumo` en `trsumo`.

Verder moet de verticale integratie van de impulsvergelijking zo worden gemaakt dat deze bij 2D berekeningen geen kwaad kan. Hierbij is het denk ik handig om te zorgen dat `aa` een alias is van `aak` bij 2D berekeningen.

Ik denk dat ik dit in 6 uur kan. Daarna is er denk ik ook 6 uur nodig om te zorgen dat alles weer/nog goed is.

de integratie van wassuc en trssuw kan denk ik in 46 uur.

4. Transportvergelijking, subroutines wasdfc en trsdif:

- (a) Invoeren van het stopcriterium per rij in subroutine trsjac van TRIWAQ (verbeteren performance).

Deze activiteit lijkt op de eerste activiteit van de integratie wassuc-trssuw.

Hij is alleen wel wat eenvoudiger, omdat de structuur van trsdif eenvoudiger is: de convergentie per rij heeft alleen effect in trsjac en daardoor in trsflx.

Omdat het eenvoudiger is, denk ik dat ik de aanpassing kanmaken in 4 uur, en 4 uur om te testen.

- (b) Stap voor stap afstemmen van de implementatie van de discretisaties in wasdfc op die in trsdif.

Deze activiteit lijkt op het afstemmen van wasumo en trsumo.

Weer twee stappen: eerst de headers het zelfde maken (3 uur) en daarna de berekeningen het zelfde maken (6 uur). Bij het gelijk maken van de berekeningen heb ik het mergen van de advectionoperatoren al meegeteld.

- (c) Inbouwen van een switch voor de horizontale advectione termen in trsdif.

Dit heb ik eigenlijk bij de vorige stap opgenomen.

- (d) Toevoegen van een aparte solver voor modellen met 1 laag en tweede orde upwind benadering in trsdif.

Het gaat hier om het overnemen van delen van waspnd in trsdif. De structuur is redelijk eenvoudig. Ik denk dat ik hier 6 uur voor nodig heb, en daarna nog 4 uur om te testen.

De integratie van wasdfc en trsdif zal in zijn geheel in 27 uur kunnen worden gedaan.

Zodra het samenvoegen van rekenroutines van WAQUA en TRIWAQ is uitgevoerd kunnen ook de koproutines voor de eerste en tweede halve tijdstap worden samengevoegd.

1. Samenvoegen van koproutines wascvu en wascvv (Chezy- en overlatenberekening).

2 uur

2. Samenvoegen van koproutines wasxru en wasxrv ($k - \epsilon$ turbulentiemodel).

3 uur

3. Samenvoegen van koproutines wastru en wastrv (transportvergelijking).

6 uur

4. Samenvoegen van koproutines wasspu en wasspv (hydrodynamica).

6 uur

Alle activiteiten uit dit hoofdstuk komen samen op 146 uur.

6.4 Globale tijdschatting

Het is moeilijk om een gedegen tijdschatting voor deze fase te maken. Niet alleen is het uit te voeren werk afhankelijk van de eerdere fases en daarom nog niet goed bekend, daarnaast is ook lastig in te schatten hoe snel de hier beschreven stappen kunnen worden uitgevoerd. Een globale indicatie van de inspanning per stap is als volgt:

Stap	Omschrijving	Inspanning
1.	samenvoegen bodemwrijving en overlaten	3 dagen
2.	samenvoegen routines impulsvergelijking	6 dagen
3.	samenvoegen routines continuïteitsvergelijking	12 dagen
4.	samenvoegen routines stoftransport	4 dagen
5.	samenvoegen koproutines x/y -richtingen	3 dagen
6.	eindtesten en rapportage	5 dagen
	onvoorzien 20%	7 dagen
TOTAAL		40 dagen

Referenties

- [1] E.A.H. Vollebregt. Parallellisatie van het horizontaal $k - \epsilon$ turbulentie model. Technical Report TR01-10, VORtech, Postbus 260, 2600 AG Delft, April 2001. In opdracht van RIKZ.
- [2] E.A.H. Vollebregt, M.R.T. Roest, and B. van 't Hof. Detailontwerp domein decompositie met horizontale verfijning. Technical Report TR01-06 (versie 2.2), VORtech, Postbus 260, 2600 AG Delft, November 2004. In opdracht van RIKZ.
- [3] E.A.H. Vollebregt, B. van 't Hof, and J.A.Th.M. van Kester. Detail ontwerp verbetering droogvallen en onderlopen in WAQUA/TRIWAQ. Technical Report TR04-02, VORtech, Postbus 260, 2600 AG Delft, 2004.
- [4] M. Zijlema and G.S. Stelling. Further experiences with computing non-hydrostatic free-surface flows involving water waves. *Int.J.Num.Meth.Fluids*, 48:169–197, 2004.