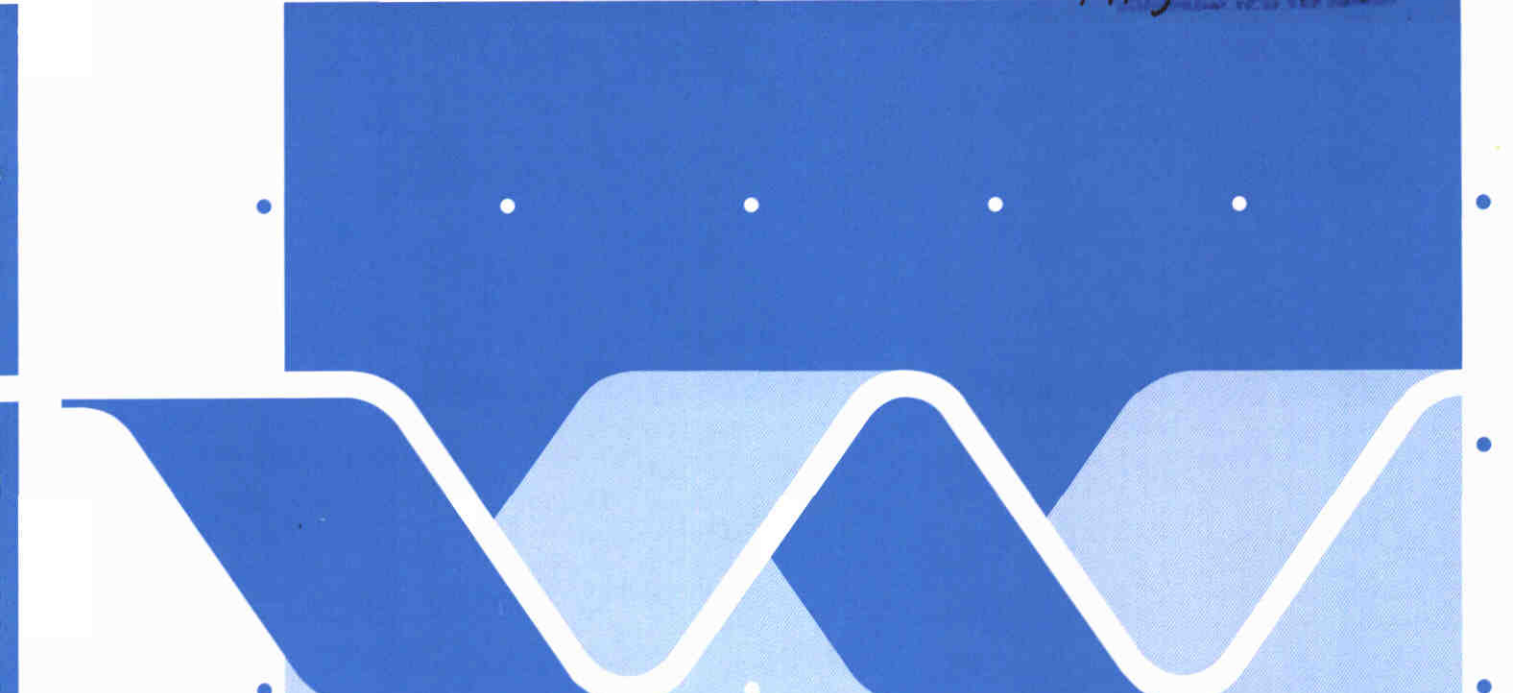




Opdrachtgever:

DG Rijkswaterstaat / RIKZ

Koppeling van ESTMORF aan een HABIMAP voorspelling

Koppeling ESTMORF, SCALWEST en HABIMAP.

november 2001

Opdrachtgever:

DG Rijkswaterstaat / RIKZ

Koppeling van ESTMORF aan een HABIMAP voorspelling

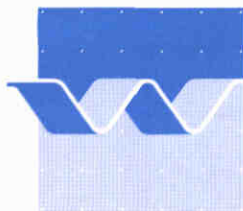
Koppeling ESTMORF, SCALWEST en HABIMAP.

M.B. de Vries

M.A.G. van Helvert

L. Verhage

november 2001



wl | delft hydraulics

OPDRACHTGEVER: DG Rijkswaterstaat / RIKZ

TITEL: Koppeling van ESTMORF aan een HABIMAP voorspelling.

SAMENVATTING:

In dit project is koppelingsprogrammatuur ontwikkeld die ESTMORF, SCALWEST (Delft3D-versie) en HABIMAP met elkaar verbinden. ESTMORF is een 1D lange termijn morfologisch model. SCALWEST is een kromlijinig 2D-hydrodynamisch model. HABIMAP is een programma in GIS die het mogelijk maakt om ecologische kaarten te maken. Met de koppeling kunnen in de toekomst de volgende parameters met HABIMAP worden gepresenteerd:

- zoutgehalte;
- droogvalduur;
- hoogteligging;
- diepte gemiddelde snelheid;
- orbitaal snelheid.

De drie koppelingsprogramma's worden aangeduid met P1, P2 en P3.

- P1 zorgt voor de conversie van een 1D naar een 2D bodem/bathymetrie. ESTMORF rekent bijvoorbeeld over een periode van 30 jaar een nieuwe bodem uit. Aan de hand van deze nieuwe bodem kan in SCALWEST een hydrodynamische berekening worden gemaakt. P1 zorgt ervoor dat de 1D-bodem uit ESTMORF geschikt wordt gemaakt om in SCALWEST te kunnen worden gebruikt.
- P2 zorgt dat de ecologische parameters in een afzonderlijk ASCII bestand worden geplaatst. Deze worden gegenereerd uit de DelftD uitvoer bestanden, nadat met SCALWEST met de nieuwe bodem uit P1 een nieuwe berekening is gemaakt.
- P3 zorgt ervoor dat de bestanden die zijn gemaakt in P2 worden omgezet naar shape-bestanden die dan vervolgens voordat ze in HABIMAP kunnen worden gebruikt, eerst nog naar een coverage moet worden omgezet.

REFERENTIES:

VER.	AUTEUR		DATUM		OPMERK.	REVIEW		GOEDKEURING	
1	van Helvert/de Vries	my	24 oktober 2001			Z.B. Wang		T. Schilperoort	
1	Helvert/Vries/Verhage	hvd	12 november 2001			Z.B. Wang		T. Schilperoort	1.
PROJECTNUMMER:			Z3179						
TREFWOORDEN:			Westerschelde, morfologie, ecologie, GIS						
INHOUD:	TEKST	17	TABELLEN	3	FIGUREN	2	APPENDICES	3	
STATUS:		☐ VOORLOPIG ☐ CONCEPT ☒ DEFINITIEF							

Inhoud

1	Inleiding	1-1
2	Opzet en modellen.....	2-2
2.1	Opzet.....	2-2
2.2	Toepassing bestaande modellen	2-3
3	Beschrijving bestaande programmatuur.....	3-4
3.1	ESTMORF/SOBEK	3-4
3.2	SCALWEST	3-5
3.3	HABIMAP	3-5
4	Werking koppelingsprogrammatuur.....	4-1
4.1	P1(1D2D conversie)	4-1
4.2	P2 (kwantificering ecologische parameters)	4-1
4.3	P3 (conversie naar coverages/shape-bestanden).....	4-3
5	User manual.....	5-1
5.1	Directory structuur.....	5-1
5.2	Werkwijze.....	5-2
6	Testfase	6-1
6.1	Uitgevoerde tests	6-1
6.1.1	P1	6-1
6.1.2	P2	6-1
6.1.3	P3	6-1
7	Aanbeveling	7-1

Bijlagen

A	Technisch ontwerp P3	A-2
B	The batch-bestanden.....	B-1
C	ascii2shape.pcf.....	C-1

I Inleiding

Het RIKZ voert in opdracht van directie Zeeland het project ZEEKENNIS uit, dat als doel heeft om op de (middel-)lange termijn bij te dragen aan de kennisontwikkeling en advisering over Integraal Waterbeheer in het Schelde-estuarium. Als onderdeel van het project ZEEKENNIS is het deelproject "Bodemfauna & Oevervegetatie" gedefinieerd, dat ingaat op de koppeling tussen de veranderingen in de fysische omgeving en het voorkomen van macrobenthos en vegetatie in met name het intergetijdengebied.

Een belangrijke openliggende vraag hierbij is de doorvertaling van de ingrepen en maatregelen in het fysische systeem (nodig voor de handhaving en verruiming van de vaargeul) naar de effecten op het ecologische systeem. Naast de directe doorvertaling moet ook rekening gehouden worden met terugkoppelings-mechanismen. Vanuit eerder inventarisaties is verondersteld dat het ecologisch functioneren onder druk staat (Vroon et.al, 1998). De EU-richtlijnen verplichten tot compensatie van verloren gegaan natuurgebied als gevolg van menselijke ingrepen (bijv. verdieping).

In het Schelde estuarium is de bodemfauna en de (oever)vegetatie een essentieel onderdeel van het voedselweb. Met name demersale vissen en vogels zijn voor hun voedsel in grote mate afhankelijk van het leven in en op de bodem. Fysische factoren zoals overstromingsfrequentie, morfodynamiek door waterbeweging en golfwerking, de bodemsamenstelling (zand-slib), de bodemkwaliteit (verontreiniging), alsook het zoutgehalte en de kwaliteit van het bovenliggende water zijn bepalend voor de ontwikkeling van de bodemflora en -fauna.

Binnen het deelproject "Bodemfauna & Oevervegetatie" is er voor gekozen om een voorspelling van effecten van ingrepen op ecotopen op basis van de bestaande morfologische modellen te maken. Concreet betekent dit dat er een koppeling gemaakt moet worden tussen ESTMORF en de door RIKZ gebruikte HABIMAP-applicatie. HABIMAP produceert ecotopenkaarten waarop de eerder genoemde fysische factoren worden weergegeven. Per fysische factor kan een parameter gedefinieerd worden. Deze zijn als volgt:

1. Een default kaart of veld met daarin het zoutgehalte op basis van een bestaande SCALWEST run.
2. Een default kaart of veld met daarin de orbitale snelheid. De opdrachtgever mag zelf bepalen hoe de orbitale snelheid wordt berekend.
3. Diepte gemiddelde stroomsnelheid (rechtstreeks uit het model).
4. Hoogteligging;
5. Droogvalduur.

De opdracht is alleen gericht op het maken van de programmatuur, die de bestaande modellen met elkaar koppelt en uiteindelijk invoer produceert voor verwerking in HABIMAP. Calibreren van een bepaalde periode behoort niet tot de opdracht.

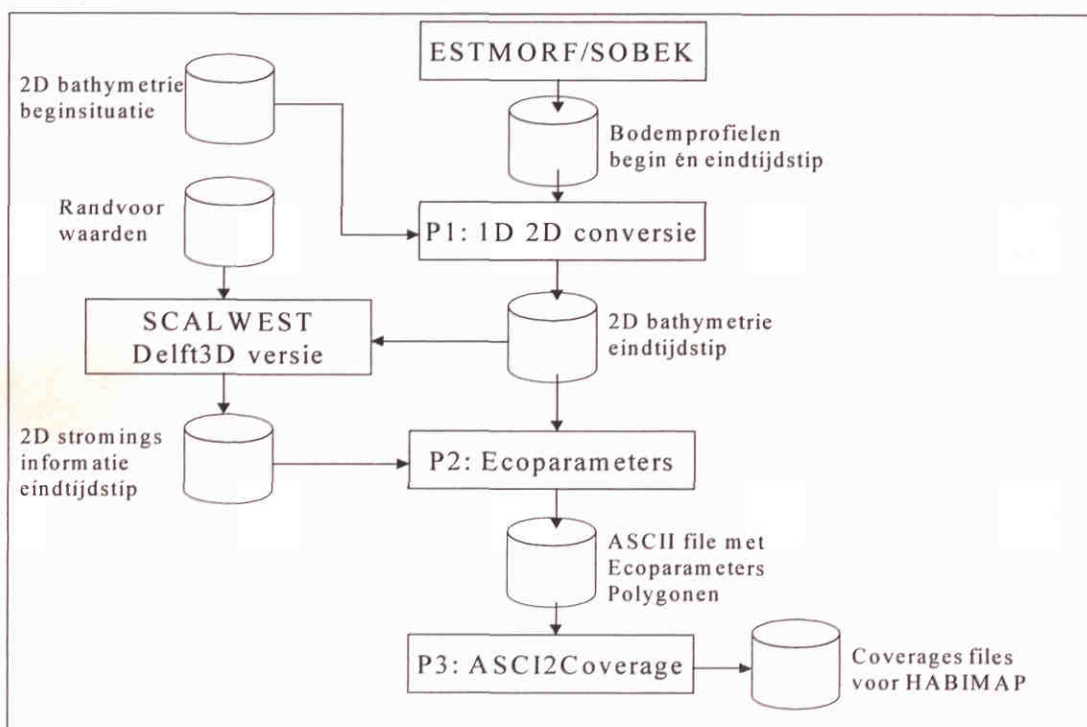
2 Opzet en modellen

2.1 Opzet

De koppeling tussen ESTMORF en HABIMAP zal worden verzord door koppeling- of conversieprogrammatuur. Omdat de te presenteren parameters typische 2D-parameters zijn, moet een 2D-hydrodynamisch model worden gebruikt/toegevoegd. Gekozen is voor SCALWEST met een kromlijniggrid).

Het hele proces (het starten van de modellen en de conversieprogrammatuur en het sturen van de datastroom) zal offline worden uitgevoerd. Dit betekent dat het proces niet door een overkoepelend programma wordt aangestuurd. Alle applicaties zullen apart moeten worden gestart. De datastroom wordt in principe handmatig gestuurd en klaargezet. Uiteindelijk levert dit een 5-tal coverages op die als invoer dienen in HABIMAP. Iedere coverage bevat een parameter.

Het product is een 'offline-product', bestaande uit de conversieprogramma's P1, P2 en P3. Deze drie programma's verbinden de bestaande modellen ESTMORF, SCALWEST en HABIMAP met elkaar. Wat ze doen is uitvoer van het ene model geschikt maken als invoer voor het vervolg model. In figuur 2.1 is een stroomschema afgebeeld van de programmatuur en de data.



Figuur 2.1 Stroomschema van de ESTMORF - HABIMAP koppeling.

De werkmethode kan puntsgewijs kort als volgt worden omschreven:

1. Met SCALWEST een beginsituatie doorrekenen.
2. Een morfologische berekening met ESTMORF maken, bijvoorbeeld 30 jaar.
3. De nieuwe berekende bathymetrie door ESTMORF wordt door P1 omgezet van een 1D naar 2D bathymetrie en geschikt gemaakt voor SCALWEST.
4. Met de nieuwe bathymetrie wordt een nieuwe berekening gemaakt in SCALWEST.
5. P2 zorgt voor de conversie naar ecoparameters.
6. P3 maakt de ecoparameters geschikt om in HABIMAP te worden verwerkt.

Nadere beschrijving van de invoer en uitvoer voor de conversieprogramma's worden later in dit rapport beschreven .

2.2 Toepassing bestaande modellen

De bestaande modellen die worden toegepast in de hiervoor beschreven opzet zijn ESTMORF, SCALWEST en HABIMAP. De status van deze modellen zijn als volgt:

- Voor ESTMORF wordt de meest recente versie gebruikt. Onlangs (augustus 2001, project Z3105) is versie 3.0 opgeleverd waarin ESTMORF is voorzien van de monding van de Westerschelde. De waterbeweging wordt uitgevoerd door het 1D waterbewegingsmodel SOBEK.
- De Delft3D versie van SCALWEST zal worden gebruikt. SCALWEST bevat een kromlijniggrid of curvilinear rooster. Op dit rooster zullen de parameters uit de vijf coverages in HABIMAP worden gepresenteerd.
- HABIMAP is een GIS-applicatie. Uitgangspunt is dat de ARCINFO versies bij WL en RIKZ geen problemen veroorzaken voor het respectievelijk genereren en verwerken van de coverages.

HABIMAP zal niet bij het WL worden geïnstalleerd voor gebruik.

Het is de bedoeling dat het genereren van de coverages zowel bij WL als bij het RIKZ kan worden uitgevoerd.

Later in rapport zullen de gebruikte modellen uitvoeriger worden beschreven.

3 Beschrijving bestaande programmatuur

3.1 ESTMORF/SOBEK

Het ESTMORF/SOBEK model voor de Westerschelde kunnen morfologische berekeningen voor de lange termijn (decennia) worden uitgevoerd. Het is een 1D model die bestaat uit een dynamisch deel en een empirisch deel. Het dynamisch deel bestaat uit een advectie-diffusie vergelijking, waarmee op basis van concentratie(verschillen) het transport wordt berekend. Het empirische deel bestaat uit evenwichtsrelaties die de morfologie berekenen.

ESTMORF maakt gebruik van het 1D-waterbewegingsmodel SOBEK. De Westerschelde is geschematiseerd in vakken. Ieder vak is weer verdeeld in drie morfologische eenheden. Dit zijn de geul (onder laag water), de lage plaat (tussen laag water en gemiddeld water) en de hoge plaat (tussen gemiddeld water en hoog water). Voor alle drie eenheden worden empirische relaties gebruikt. Voor de geul is het getijvolume de belangrijkste fysische parameter. Voor de lage en de hoge platen is dit de getijslag Δh , wat het verschil is tussen hoog en laag water.

De randvoorwaarden bestaan uit een zeewaartse rand in het westen en de afvoer van de Schelde in het oosten. Aan de zeewaartse rand is een cyclisch getij gedefinieerd wat een matig springtij is van 11 mei 1970 van 08.40 t/m 21.00 uur is. In het oosten is dit een rivierafvoer van 29 m³/s.

De morfologische calibratie voor ESTMORF/SOBEK is uitgevoerd over een periode van 1968 t/m 1996. De bodem in het model is samengesteldt uit lodingen die dateren van eind jaren zestig. Vanaf die tijd zijn er regelmatig lodingen uitgevoerd in de Westerschelde. Door deze lodingen is een goed beeld ontstaan van de morfologische ontwikkeling van de Westerschelde wat gunstig was voor de morfologische calibratie van het model.

Met ESTMORF is het mogelijk om het effect van baggeren en storten op de morfologie op de lange termijn te berekenen. De werkmethode bepaalt namelijk de invoer voor toekomstige berekeningen. Als invoer is nodig:

- Een bodem die is weggeschreven aan het eind van de calibratie berekening. Dit is de bodem van de Westerschelde die door ESTMORF is berekend. Dit is eind 1996;
- Bagger- en stortscenario's voor de toekomst. Dit kan op twee manieren worden uitgevoerd:
 1. een opgelegd bagger- en stortscenario, d.w.z. de hoeveelheden zijn min of meer bekend en kunnen in een ingrepen bestand worden ingevoerd of,
 2. een minimaal vereiste baggerdiepte en -breedte van de baggerplaatsen en een stortscenario met stortcapaciteiten per stortplaats.
- Overige invoer zoals zeespiegelstijging, bodemdaling en hoeveel jaar gerekend moet worden.

De uitvoer bestaat uit een 1D bodem schematisatie van de Westerschelde. Dit bestand (estmorf.map genoemd) moet worden geconverteerd naar een 2D bathymetrie die geschikt is voor SCALWEST. Verder levert ESTMORF hydrodynamische uitvoer zoals waterstanden en debieten. De plaatsen waar deze moeten worden weggeschreven moeten wel overeen komen met de plaatsen in het SCALWEST model.

3.2 SCALWEST

In het proces kan de Delft3D versie van SCALWEST worden toegepast. SCALWEST berekend voor de hele Westerschelde de 2D-hydrodynamische informatie voor tijdstippen in de getijcyclus, voor een door de gebruiker te bepalen periode. Dit model is door het RIKZ ontwikkeld en gebouwd. Ten tijde van de uitvoer van dit project was er geen versienummer van SCALWEST beschikbaar. De bodem die in het model wordt gebruikt dateert uit 1996.

Benodigde invoer:

- 2D-bathymetrie (eindsituatie). Dit bestand staat in een Delft3D formaat;
- Overige randvoorwaarden (onder andere de wijze van getij-simulatie en de duur van de simulatie)

De uitvoer bestaat uit 2D stromingsinformatie over een aantal getijcycli in Delft3D formaat (eindsituatie).

3.3 HABIMAP

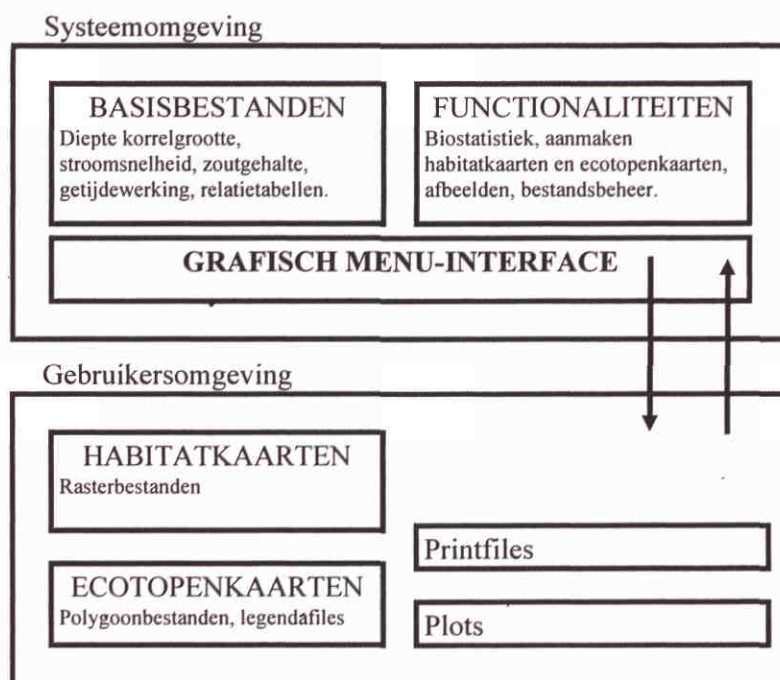
HABIMAP is een unix-ARC/INFO applicatie met aml en formmenu geprogrammeerd die in staat is habitat- en ecotopenkaarten te maken van een gebied. De applicatie HABIMAP bestaat uit drie hoofdonderdelen:

1. Een database met gegevens.
2. Een aantal functionaliteiten.
3. Een menugestuurd gebruikersinterface.

De functionaliteiten (punt twee) is opgedeeld in zeven modules, te weten:

1. Biostatistiek;
2. Habitatkaarten (tabellen);
3. Habitatkaarten (formules);
4. Habitatkaarten (complexe);
5. Ecotopenkaarten;
6. Droogvalduurkaarten;
7. Uitsnede aanmaken en afbeelden.

De structuur van dit alles is weergegeven in de figuren 3.1.



Figuur 3.1: Structuur van de applicatie HABIMAP.

Deze module is bedoeld om via biotische en abiotische kaarten uit hetzelfde gebied iets te zeggen over de relatie tussen de gekarteerde soorten en de betreffende abiotische parameters.

Voor dit project kunnen van het hydrodynamische model SCALWEST polygoenen gepresenteerd worden. Ze kunnen respectievelijk verdeeld worden in twee typen:

1. Een Habitatkaart beschrijft het potentieel leefgebied van een soort. In de applicatie zijn ze altijd een grid: ze bestaat uit een (groot) aantal cellen met een waarde in elke cel. De eenheid waarmee binnen de habitatkaart wordt gewerkt is 'leefmogelijkheden in percentages'. Een waarde van 0% in de habitatkaart betekent dat op basis van de gebruikte abiotische data en definieerde relaties het betreffende soort hier niet kan leven. Een waarde van 100% betekent dat, op basis van de gebruikte abiotische data en definieerbare relaties, de betreffende soort op deze locatie optimaal kan voorkomen. Alle tussenliggende waarden vertegenwoordigen meer of minder sterk afgenomen leefmogelijkheden voor de soort.
2. Ecotopenkaarten geeft het potentiële leefgebied van een of meer levensgemeenschappen weer. De wijze waarop binnen HABIMAP een dergelijke kaart wordt aangemaakt gaat op soortgelijke wijze als bij de habitatkaarten. Een ecotopenkaart kan een grid zijn maar wordt dan altijd omgezet naar een coverage (polygoenenkaart).

4 Werking koppelingsprogrammatuur

4.1 P1 (1D2D conversie)

Programma P1 wordt toegepast om de 1D modelresultaten van ESTMORF om te zetten naar een 2D modelgrid. Uitgegaan wordt van een resultaat op het grid zoals wordt gebruikt door SCALWEST. Opgemerkt moet worden dat SCALWEST onder Delft3D moet kunnen draaien.

Benodigde invoer:

- Een Delft3D .mdf bestand. Dit is een invoer bestand om een flow berekening te maken in Delft3D. De naam kan bijvoorbeeld zijn scalw.mdf (SCALWEST). In dit bestand wordt verwezen naar een grid bestand (scalw.grd) en een bathymetrie bestand (scalw.dep).
- Een mapping.rel bestand. Dit is een bestand waar per 1D vak van ESTMORF alle 2D punten zijn gedefinieerd.
- ESTMORF uitvoer bestand, estmorf.map. Dit is een 1D-bathymetrie aan het eind van een morfologische berekening van bijvoorbeeld 30 jaar.

De executable die voor verwerking zorgt heet Bot2bot.exe. Uit het estmorf.map bestand worden voor alle profielen/doorsneden het verschil berekend tussen de eerste en de laatste tijdstap. De veranderingen worden doorgegeven aan de oorspronkelijke bathymetrie uit het SCALWEST Delft3D-model. De relatie tussen de bodempunten en de ESTMORF vakken wordt uit de mapping.rel bestand gehaald.

Uitvoer:

- 2D-bathymetrie in Delft3D formaat voor gebruik in SCALWEST inclusief alle morfologische veranderingen berekend door ESTMORF.

Na afloop is er een nieuwe Delft3D-bodem beschikbaar. Deze heet altijd output.dep en kan gewoon hernoemd worden naar eigen keuze. De nieuwe bodem kan worden gebruikt voor een nieuwe Delft3D run.

4.2 P2 (kwantificering ecologische parameters)

Nadat met de nieuwe bodem afkomstig uit P1 een flow-berekening is uitgevoerd kan het programma P2 voor de kwantificering van de noodzakelijke ecologische parameters verzorgen. Met P2 worden de volgende parameters berekend:

- Bodemhoogte in meters t.o.v. een door de gebruiker gehanteerde referentie niveau;
- Waterstand in meters t.o.v. een door de gebruiker gehanteerde referentie niveau;
- Saliniteit in mg/l;
- Temperatuur in graden Celsius;
- Bodemschuifspanning in N/m^2 ;

- Stroomsnelheid in m/s
- Inundatieduur in %;

Deze lijst parameters voldoen aan de geeiste parameters genoemd in de inleiding. Als extra is de temperatuur en de bodemschuifspanning mogelijk, mits deze in de SCALWEST berekening worden aangezet. Omdat SCALWEST een 2D-model is, zijn alleen de diepte gemiddelde stroomsnelheden berekent.

De invoer voor P2 is 2D-stromingsinformatie aan de hand van de nieuwe bathymetrie afkomstig uit P1. Dit zijn twee bestanden die in een Delft3D format staan opgeslagen, namelijk een trim-xxx.def en een trim-xxx.dat bestand.

De executable die voor verwerking zorgt heet habimap.exe.

Als uitvoer wordt per parameter een ASCII-bestand gemaakt met daarin polygonen en per polygoon de desbetreffende parameterwaarden. Programma P2 maakt in dit geval zeven ASCII-bestanden. De namen met de bijbehorende parameter van deze ASCII-bestanden zijn als volgt:

- param-00.pol voor de bodemhoogte;
- param-01.pol voor de waterstand;
- param-02.pol voor de saliniteit;
- param-03.pol voor de temperatuur;
- param-04.pol voor de bodemschuifspanning;
- param-05.pol voor de stroomsnelheid;
- param-06.pol voor de inundatie duur.

In de param-01.pol t/m bestand param-05.pol wordt per parameter drie waarden gegeven, namelijk de minimale, gemiddelde (avg) en maximale waarde. In de param.pol bestanden voor de bodemhoogte en de inundatie duur staat maar één waarde. Het gevolg is dat voor iedere parameterwaarde een apart shape-bestand zal worden gemaakt. Dit wordt in P3 nader beschreven.

Om toch een shape-bestand te verkrijgen voor de orbitaalsnelheid is een willekeurige param.pol bestand met drie parameterwaarden gekopieerd. Deze krijgt de naam param-07.pol. In deze file zijn de maximale, gemiddelde en de minimale waarden vervangen door default waarden 0,1 (Allemaal 'handmatig' uitgevoerd). In HABIMAP kunnen deze waarden worden bewerkt. Er zijn dus acht param.pol bestanden. De achtste is:

- param-07.pol voor de orbitaal snelheid.

De header van iedere param.pol bestand bestaat uit een aantal regels. Deze zijn als volgt:

1. omschrijving;
2. curvilinear;
3. rund-id xxx van de Delft3D trim-xxx.def en trim-xxx.dat bestanden;
4. getal die het aantal parameterwaarden aangeeft. Voor param-00.pol en param-06.pol is dit 1 en voor param-01.pol t/m param05.pol is dit 3;
5. parameterwaarde 1;
6. parameterwaarde 2;

7. parameterwaarde 3;
8. DATA-BLOCK.

De regel 'DATA BLOCK' geeft aan dat er een blok volgt met daarin de data. Hierover later meer.

Wanneer een 3D-hydrodynamisch model is gebruikt, is het mogelijk om P2 voor de parameters saliniteit, temperatuur en stroomsnelheid uit verschillende lagen de desbetreffende parameterwaarden te genereren. Bij het aanroepen van de executable moet dit dan worden opgegeven. Later in dit rapport wordt hierop nog teruggekomen. Bij een 2D-model zijn deze waarden diepte gemiddeld.

4.3 P3 (conversie naar coverages/shape-bestanden)

Programma P3 converteert de param.pol bestanden naar shape-bestanden die kunnen worden gebruikt in ArcInfo. Arcinfo converteert vervolgens deze shape-bestanden naar coverages die door HABIMAP, onder UNIX, weer kunnen worden ingelezen. Echter is de laatste conversie buiten dit project gehouden.

Voor P3 zijn de zeven (of acht) param.pol bestanden nodig en een invoer file voor de executable in P3 namelijk de ascii2shape.pcf bestand. Dit bestand is belangrijk omdat deze o.a. informatie verschaft hoe de polygonen in de param.pol bestanden gelezen moeten worden. Dit wordt aangeduid met CLOCKWISE. Verder staat in dit bestand info die niet hoeft worden gewijzigd en dus hetzelfde kan blijven.

De executable die voor verwerking zorgt heet ASCII2SHAPE.EXE.

De executable genereert voor iedere parameterwaarde een shape-bestand. Bijvoorbeeld voor de minimale waterstand, de gemiddelde waterstand en de maximale waterstand ieder een shape-bestand, totaal dus drie shape-bestanden. De naam van het shape-bestand bestaat uit maximaal dertien karakters. De eerste acht zijn gereserveerd voor de parameternaam met de waarde. De overige karakters zijn gereserveerd voor de run-id die voorkomt in de trim-xxx.def en trim-xxx.dat Delft3D bestanden. Bijvoorbeeld voor de minimale waterstand wordt dan 'WATERMIN_XXX.shp'. Als de totale naam voor een shape-bestand uit meer dan dertien karakters bestaat dan wordt de naam van de parameter afgekapt.

Door deze benaming van de shape-bestanden worden voor de parameters de volgende termen gebruikt:

- voor de bodemhoogte BOTTOMLV_XXX.shp.
- voor de minimale, gemiddelde en maximale waterstand respectievelijk WATERMIN_XXX.shp, WATERAVG_XXX.shp en WATERMAX_XXX.shp.
- voor de minimale, gemiddelde en maximale saliniteit respectievelijk SALTMIN_XXX.shp, SALTAVG_XXX.shp en SALTMAX_XXX.shp.
- voor de minimale, gemiddelde en maximale temperatuur respectievelijk TEMPMIN_XXX.shp, TEMPAVG_XXX.shp en TEMPMAX_XXX.shp.
- voor de minimale, gemiddelde en maximale bodemschuifspanning respectievelijk FRICMIN_XXX.shp, FRICAVG_XXX.shp en FRICMAX_XXX.shp.

- voor de minimale gemiddelde en maximale stroomsnelheid respectievelijk CURRMIN_XXX.shp, CURRAVG_XXX.shp en CURRMAX_XXX.shp.
- voor de inundatieduur FLOODING_XXX.shp.
- voor de minimale gemiddelde en maximale orbitaalsnelheid respectievelijk ORBIMIN_XXX.shp, ORBIAVG_XXX.shp en ORBIMAX_XXX.shp.

Als een parameter geen waarde heeft dan wordt de polygoon ook niet aangemaakt in de shape voor die parameter.

Als bijkomende bestanden die nodig zijn voor Arcinfo, worden met dezelfde bestandsnamen ook nog bestanden met een .shx en een .dbf extensie aangemaakt. Totaal worden er dus voor de zeven parameters één-en-vijftig bestanden aangemaakt als output van P3. Met de orbitaal snelheid komt het totaal op zestig bestanden.

De format van de param.pol bestanden gegenereerd in P2 is belangrijk voor het draaien van P3. De header informatie in de param.pol bestanden (zie in paragraaf 4.2) is dus belangrijk. Geadviseerd wordt om hierin niets te wijzigigen. In het data blok worden waarden voor de in de header genoemde parameters weggeschreven. Als een parameter geen waarde heeft dan wordt die waarde ook niet weggeschreven.

Op elke regel in het blok worden de gegevens voor de polygoon met de daar bij behorende waarden van de parameters weggeschreven.

Elke regel begint met alle x, y coördinaten van de polygoon. Omdat een polygoon een gebied in moet sluiten moeten er minimaal drie paren aanwezig zijn, hier wordt op getest; het programma weet dat het laatste punt met het eerste moet worden verbonden. Na de coördinaten volgen de waarden van de parameters. Een voorbeeld van een regel:

```
x1,y1,x2,y2,.....,xi,yi,wp1,wp2,wp3
```

Deze regel bevat i coördinaten-paren met daar aan gekoppeld drie waarden. Dat betekent dat er ook drie parameter namen in de header aanwezig moeten zijn. Het aantal coördinaten paren in elke regel is vrij; het programma weet doordat het aantal parameters bekend is hoeveel paren er elke keer in een regel staan. De volgorde van de paren legt ook de polygoon vast waarbij het niet van belang is of de coördinaten met de klok mee of tegen de klok in worden weggeschreven. De laatste coördinaat wordt aan de eerste gekoppeld om de polygoon te sluiten. Oppassen dat de volgorde van de paren niet zodanig is dat lijnen elkaar snijden want de BUILD-operatie van ArcInfo maakt er dan meer polygonen van.

De komma's worden als scheiding gebruikt om in staat te zijn aan te geven of een parameter een waarde heeft. Als in de voorbeeld regel parameter twee geen waarde heeft dan ziet de regel er als volgt uit:

```
x1,y1,x2,y2,.....,xi,yi,wp1,,wp3
```

Als de laatste parameter geen waarde heeft dan is het laatste karakter van de regel dus een komma (,).

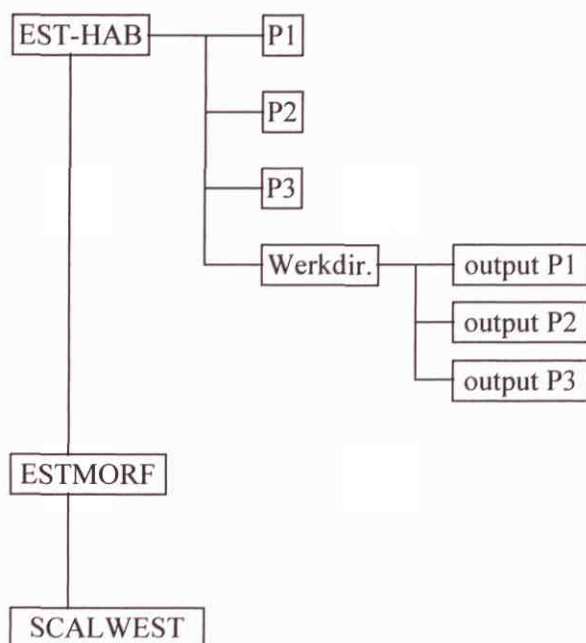
Het programma P3 leest alle regels tot aan het eind van het bestand (EOF) zodat in de header niet het aantal regels met polygonen behoeft te staan.

In bijlage A wordt het technisch ontwerp beschreven van P3.

5 User manual

5.1 Directory structuur

De ontwikkelde programmatuur P1, P2 en P3 zijn losse executables die apart aangeroepen moeten worden. De benodigde invoer bestanden moeten worden klaar gezet en worden verplaatst tussen de verschillende directories. Dit wordt een off-line werkmethode genoemd. De beste werkwijze is een directory structuur te ontwerpen waarin onderscheidt wordt gemaakt tussen vaste invoer bestanden en variabele invoer bestanden. De vaste invoer bestanden zijn de bestanden die nodig zijn om de koppelingsprogrammatuur te laten werken. De variabele bestanden zijn bijvoorbeeld de bestanden die afhankelijk zijn per reken- of onderzoekscenario. Het is dan het beste om deze variabele bestanden te plaatsen in een werkdiractory die de naam krijgt van het desbetreffende rekenscenario. De vaste bestanden blijven in principe in de directory waar ze thuis horen en de kans dat hiermee iets ongewenst gebeurt is klein. Zo is de volgende directory structuur mogelijk:



Figuur 4.1: Directory structuur werkwijze ESTMORF-HABIMAP.

Wanneer er voor een nieuw scenario HABIMAP bestanden gemaakt moeten worden, kan de werkdiractory worden gekopieerd. Met een batch-bestand kunnen de koppelingprogramma's worden aangeroepen en de bewerking uitgevoerd. Het voordeel van een batch-bestand is dat de benodigde bestanden van en naar de werkdiractory worden verplaatst. In de werkdiractory staan drie batch-bestanden. Deze zijn:

- runP1.bat;
- runP2.bat;
- runP3.bat.

In bijlage B wordt de inhoud van de batch-bestanden gegeven.

5.2 Werkwijze

Vanuit de werkdirectory worden de koppelingsprogramma's aangestuurd. In deze directory staan alle scenario-afhankelijke bestanden zoals de ESTMORF uitvoer (estmorf.map) en de uitvoer van het hydrodynamisch model (trim-xxx.dat en trim-xxx.def). Verder staan er de eerder genoemde batch-bestanden die de koppelingsprogramma's aanroepen.

Met de batch-bestanden wordt de benodigde (vaste) invoer plus de executable naar de werkdirectory gekopieerd en na uitvoering de uitvoer naar de desbetreffende output directory onder de werkdirectory gekopieerd. Al het overbodige wordt tot slot uit de werkdirectory verwijderd.

P1 heeft als vaste invoer het Delft3D-mdf bestand om flow berekeningen te maken. In deze xxx.mdf bestand wordt verwezen naar het grid van het model met bijbehorende bodem van het model. Dit zijn de xxx.grd en xxx.dep bestanden. Voor SCALWEST moet in deze directory de bijbehorende bestanden worden geplaatst. Bijvoorbeeld scalw.mdf met scalw.grd en scalw.dep. Verder mag als vaste invoer de mapping.scw bestand niet ontbreken.

Wanneer het batch-bestand wordt aangeroepen worden de vaste bestanden en de executable bot2bot.exe naar de werkdirectory gekopieerd. Het mapping.scw bestand moet worden gekopieerd naar de vaste bestandsnaam voor P1 mapping.rel. Wanneer er met een ander Delft3D model wordt gewerkt als met SCALWEST, dan moet dit door de gebruiker in het batch-bestand worden aangepast/aangegeven.

Nadat de executable zijn werk heeft uitgevoerd worden alle uitvoer gekopieerd naar de outputP1 directory. Deze staat, net zoals outputP2 en outputP3, onder de werkdirectory zodat alle belangrijke uitvoer bij het bijbehorende scenario worden bewaard. In de outputP1 directory staat de nieuwe bodem die altijd de naam output.dep krijgt. Deze kan worden hernoemd en weer worden gebruikt voor een nieuwe flow berekening in Delft3D. De overige bestanden zijn report bestanden die kunnen worden geraadpleegd als er iets mis gaat.

De vaste invoer voor P2 is alleen de executable habimap.exe. De variabele invoer bestanden staan in de werkdirectory en zijn de trim-xxx.dat en trim-xxx.def bestanden. In de trim-bestanden staan Delft3D-flow uitvoer die zijn gegenereerd met bijvoorbeeld de nieuwe bodem uit P1. Deze uitvoer zal uiteindelijk in habimap gepresenteerd moeten worden.

Wanneer het batch-bestand runP2 wordt aangeroepen wordt de executable vanuit de P2 directory naar de werkdirectory gekopieerd en vervolgens gestart. Voor iedere parameter wordt een param.pol bestand gemaakt. In de outputP2 directory worden de zeven bestanden geplaatst. Deze zijn:

- param-00.pol voor de bodemhoogte;
- param-01.pol voor waterstand;
- param-02.pol voor saliniteit;

- param-03.pol voor de temperatuur;
- param-04.pol voor de bodemschuifspanning;
- param-05.pol voor de diepte gemiddelde stroomsnelheid;
- param-06.pol voor de inundatie duur.

Het batch-bestand voor P2 zorgt dat alles weer wordt opgeruimd en zo de werkdirectory alleen de benodigde bestanden heeft voor verdere verwerking.

In het batch-bestand moet de begintijd en eindtijd worden opgegeven waarover de data is weggeschreven (zie bijlage B). Daaronder staat een blok met daarin de nummers die overeenkomen met de nummers van de param.pol bestanden (0 t/m 6). Voor de parameters saliniteit, temperatuur en stroomsnelheid wordt opgegeven uit welke laag de waarden moeten worden gegenereerd. In het batch-bestand in bijlage B staat achter de ECHO 0. Dit betekent dat de diepte gemiddelde waarde wordt gegenereerd. Wanneer achter ECHO 1 staat, wordt voor de desbetreffende parameters de waarde uit de bovenste laag gegenereerd. Dit is toepasbaar wanneer een 3D-model is toegepast.

Wanneer de begintijd en eindtijd waarover de data is weggeschreven onbekend zijn, wordt door het aanroepen van alleen habimap.exe de tijden in het DOS-scherf gegeven.

Bij aanroepen van de executable wordt eerst de naam van de trim bestanden zonder extensie gevraagd (dus trim-xxx). Nadat deze zijn ingevoerd en een return is gegeven verschijnen de begintijd en de eindtijd in het DOS-scherf. Deze moeten vervolgens weer worden ingevoerd omdat de executable ernaar vraagt (dit is te zien doordat er een vraagteken in het DOS-scherf zichtbaar wordt).

Daarna geeft de executable een lijst van alle parameters met het daarbij behorende nummer. Door het nummer in te voeren wordt van de desbetreffende parameter een param.pol bestand gemaakt. Voor de parameters saliniteit, temperatuur en stroomsnelheid moet ook nog eens worden opgegeven of het uit de bovenste laag moet komen (een 1) of diepte gemiddeld moet zijn (een 0).

Op deze manier kan handmatig de lijst worden afgewerkt. Wanneer alles is uitgevoerd kan met ctrl+c de executable (evt. eerder) worden afgebroken. In de trim-xxx.rpt bestand kan worden bekeken wat voor informatie in de trim bestanden staan opgeslagen.

De .pol bestanden zijn de invoer bestanden voor P3 die de shape-bestanden genereren. Er zijn twee bestanden voor P3. Dit zijn:

- ASCII2Shape.exe;
- ascii2shape.pcf.

De eerste is de executable. Het tweede, pcf-bestand, is het script die aangeeft wat o.a. de algemene namen zijn van de .pol bestanden (param.pol) en in welke richting de polygonen moeten worden gelezen. In bijlage C staat de inhoud van ascii2shape.pcf gegeven.

In het batch-bestand runP3 wordt de executable en het script bestand gekopieerd naar de werkdirectory. Uit de outputP2 directory wordt de eerste param-01.pol gekopieerd en hernoemd naar de algemene naam param.pol. De executable wordt gestart en uitgevoerd. Het resultaat zijn per param.pol voor de minimum, gemiddelde en maximale waarde ieder drie shape-bestanden plus de twee bijbehorende bestanden (.dbf en .shx). Deze worden

gekopieerd naar outputP3. Vervolgens wordt hetzelfde uitgevoerd voor de overige param.pol bestanden.

Deze werkwijze is natuurlijk geheel vrij. De batch-bestanden en de directory structuur kunnen naar wens van de gebruiker worden aangepast.

6 Testfase

6.1 Uitgevoerde tests

6.1.1 P1

Test	Resultaat
Wordt de 1D informatie (verandering diepte) goed op het 2D grid afgebeeld?	Ja, in quick-in wordt de nieuwe bodem goed zichtbaar.
Wordt de verandering goed opgeteld bij de originele bathymetrie?	De nieuwe bodem is een totaal nieuwe bathymetrie. De morfologische veranderingen worden niet verwerkt in de initiele bathymetrie.
Kan de uitvoer bestand worden ingelezen door Delft3D?	Ja, in Delft3D moet de nieuwe bodem als attribute worden ingevoerd.

6.1.2 P2

Test	Resultaat
Passen bathymetrie en flow grids bij elkaar	Ja, zie test P1
Checken of parameterwaarden goed zijn uitgerekend uit flow data. Bijv. door 1 polygoon te kiezen en met de hand na te rekenen	In GPP voor bijv. de waterstand een bestand exporteren. Dit bestand bevat de xy coördinaten met daarin waarden voor de waterstand. Dit zijn waarden uit de trim bestand. Voor een willekeurig coördinaat met bijbehorende waarden opzoeken in het param.pol bestand en de waarden vergelijken. Klopt.
Worden de {x,y} paren goed uitgevoerd.	Ja
Worden de parameterwaarden aan de goede polygoon gekoppeld	Ja, in de param.pol bestanden staat per polygoon drie waarden de minimum, gemiddelde en de maximum.
Klopt de header informatie in de ascii bestand	Ja.

6.1.3 P3

Test	Resultaat
Wordt de p2 bestand ingelezen.	Ja.
Worden de polygonen goed aangemaakt volgens de coördinaten van het oorspronkelijke modelgrid.	In de shape-bestanden kan niet worden afgeleid of dit is gebeurt. Dit kan pas echt in HABIMAP worden beoordeeld.
Wordt per "shape" de parameterwaarde aan de juiste polygoon gekoppeld.	Idem zie boven.
Worden de goede bestand namen gegenereerd.	Ja.
Kan ArcInfo de coverage inlezen.	Testen bij RIKZ
Kan HABIMAP de coverages werkelijk gebruiken.	Testen bij RIKZ.

7 Aanbeveling

De ontwikkelde conversie programmatuur zijn in principe voor alle hydrodynamische modellen bruikbaar, mits het hydrodynamische grid overeenkomt met het netwerk van het in ESTMORF gebruikt 1D-model. Ook voor 3D-modellen kunnen de conversie programmatuur worden toegepast. Echter zijn er een paar aanbevelingen te geven:

- In dit project is voor de orbitaalsnelheid handmatig een shape-bestand gemaakt door een `param.pol` te kopiëren. Voor de orbitaalsnelheid zou ook een golfmodel gebruikt kunnen worden. In Delft3D is de koppeling met een hydrodynamische model goed toepasbaar. Op deze manier kan het effect van morfologische ontwikkelingen op de lange termijn op golven en de orbitaalsnelheid bepaald worden.
- De procedure na het genereren van de shape-bestanden richting HABIMAP zou verder geautomatiseerd kunnen worden.
- Mogelijkheid om met statistische waarden te werken.
- Koppeling met 3D-modellen verder uitwerken.

A Technisch ontwerp P3

Inleiding

In het functioneel ontwerp van de genoemde koppeling wordt het programma **P3** beschreven waarmee het mogelijk is een ASCII bestand om te zetten naar een ArcInfo coverage.

Het format van de bestand is in die beschrijving vastgelegd. Het komt er in principe op neer dat een header beschrijft welke parameters aan polygonen zijn gekoppeld, en per regel wordt een polygoon gegeven met daarbij de waarden van de parameters.

Het programma HABIMAP gebruikt coverages met daarin 1 parameter per coverage. De naam van de coverages reflecteert de naam van de parameter. De ASCII bestand wordt aangemaakt op PC, het programma HABIMAP wordt onder UNIX gebruikt.

Er zijn twee mogelijkheden om deze conversie uit te voeren:

- a. het ASCII bestand eerst omzetten naar Shape-bestanden, en deze converteren naar een coverage met de SHAPEARC opdracht.
- b. een AML maken die een ASCII bestand aangemaakt op PC omzet naar de coverages voor HABIMAP.

Optie a. heeft de voorkeur omdat dan direct op PC een controle kan worden uitgevoerd op het resultaat zonder direct gebonden te zijn aan ArcInfo omdat deze niet algemeen beschikbaar behoeft te zijn. In het vervolg wordt deze optie als basis gebruikt.

ASCII2Shape opzet

Het programma wordt ontwikkeld met de Visual Fortran Development Studio van Compaq. Omdat de opzet van het programma recht toe recht aan is wordt de globale opzet in pseudo code beschreven.

Lees de bestanden met opdrachten

1. Lees naam ASCII bestand
2. Lees type klok-richting
3. Lees type opslag: één parameter/alle parameters
4. Lees behandeling lege parameter velden: "no-value"/"no-polygon"
5. Lees "no values" voor de verschillende data typen

Lees en verwerk de header van de ASCII bestand met polygonen en parameters

1. Bepaal aantal parameters, bepaal type parameter, bepaal "no value" waarde
2. Lees Jaartal

Voor elke regel in het data blok

1. Lees coördinaten en parameterwaarden
2. Controleer de coördinaten op aantal en of er geen snijdingen zijn
3. Sorteer de coördinaten kloks-of tegenkloksgevijs

Bepaal de manier waarop de parameters in de shape-bestand(en) moeten worden opgeslagen.

1. Per parameter, bepaal de naam van de bestanden en schrijf de shape-bestanden.
2. Eén shape-bestand, maak een shape-bestand aan met een gegeven naam schrijf de shape-bestand.

Bovenstaande pseudo code kan worden omgezet in hoofd-modules en sub-modules. Deze worden in het volgende hoofdstuk beschreven.

Modules

In dit hoofdstuk wordt de pseudo code uitgewerkt in modules. Met name de hoofd modules zullen overeenkomen met de hoofdlijnen uit het vorige hoofdstuk.

In de tekst zullen de namen van de modules tussen het procent teken (%) worden genoemd.

Inlezen bestand met opdrachten %ReadCommands%

Het principe is dat elk commando begint met een identificatie en dat de informatie uit het vervolg wordt gehaald. De volgorde van de commando regels in de script bestand is in principe vrij. Sommige opdrachten zijn verplicht; voor andere geldt dat de defaults worden gebruikt als het commando er niet is.

In de beschrijving worden **IDENTIFICATIES** door het programma herkend en moeten ook als zodanig worden gebruikt. Als ergens een waarde moet worden ingevuld dan wordt een *cursieve* naam gebruikt als variabele. Default waarden worden tussen [] haken gezet. Zijn er meerdere mogelijkheden aanwezig dan worden deze door het | teken gescheiden en tussen { } haken geplaatst.

Voor de rest worden aanwijzingen gegeven gedurende de beschrijvingen. De beschrijving van commando begint met ↪ **Description** *omschrijving* op de volgende regel gevolgd door het commando, en eindigen met ↪.

↪ **Description** *Naam ASCII bestand*

ESTMORF/SCALWES-BESTAND, *filedescriptor*

Dit commando is verplicht en het programma stopt als deze niet aanwezig is. Met *filedescriptor* wordt de aanduiding van de te gebruiken bestand bedoeld, en wel het pad + de bestandsnaam. Het programma eist geen speciale extensie; dat betekent dat elke naam, met of zonder extensie als zodanig wordt gebruikt.

↪

↪ **Description** *klok-richting*

COORDINATE-ORDER, {[AS-IS] | **CLOCKWISE** | **ANTICLOCKWISE**}

Het kan voordelen hebben de coördinaten op een bepaalde manier te ordenen om het gebruik in ArcView of ArcInfo te versnellen. De default waarde is nu gezet op as-is, d.w.z. geen verdere aanpassing. Dat kan veranderen als duidelijk is wat de beste oplossing is voor gebruik in de conversie naar ArcView en ArcInfo.



→ *Description* Lees type opslag: één parameter/alle parameters

STORAGE-TYPE, { ONE-FILE | [PER-PARAMETER] }

Uiteindelijk moet de informatie aan HABIMAP worden aangeboden per parameter. Het is echter mogelijk dat het voor een eerste gebruik op de PC met ArcView veel handiger is om alle parameterwaarden in dezelfde shape-bestand te hebben. Dit commando kan worden gebruikt om dat met het programma te realiseren.



→ *Description* Behandeling lege parameter velden

NO-PARAMETER-VALUE, { [NO-VALUE] | NO-POLYGON }

Het programma moet weten hoe er moet worden omgegaan met lege parameter velden. Er zijn twee opties:

1. **NO-VALUE**, het programma gebruikt een “no-value” waarde om als vervangende waarde weg te schrijven. Later wordt voor de verschillende data typen en bij de beschrijving van de parameter namen nader op ingegaan.
2. **NO-POLYGON**, als er geen parameterwaarde is dan wordt de polygoon ook niet aangemaakt.

Er zijn twee situaties, n.l. er wordt één shape-bestand aangemaakt met alle parameters of een shape-bestand per parameter. In beide gevallen is de optie **NO-VALUE** mogelijk zodat deze default wordt gebruikt.

In het geval dat er per parameter een shape-bestand wordt aangemaakt is het mogelijk de optie **NO-POLYGON** te gebruiken.

Er doet zich nog een derde situatie voor, n.l. dat er een shape-bestand wordt aangemaakt voor alle parameters en de optie **NO-POLYGON** is ingesteld. Deze optie kan in dat geval alleen worden gehonoreerd als er alleen polygonen zijn met waarden voor alle parameters en polygonen met geen enkele parameterwaarde. In alle andere gevallen zal de optie in deze situatie een foutmelding opleveren.



Als in het veld van een parameter geen waarde staat dan moet bekend zijn wat daar mee wordt gedaan. In het commando **NO-PARAMETER-VALUE** wordt aangegeven wat er in een dergelijke omstandigheid met de parameter moet worden gedaan. Als de optie **NO-VALUE** wordt gebruikt dan moet er een waarde worden gebruikt.

Het gebruik van een “no-value” waarde is een twee-traps raket, eerst wordt uit de commando bestand bepaald welke waarde voor een bepaald data type wordt gebruikt, daarna kan deze door informatie uit de Estmorf/Scalwes bestand worden herroepen.

Nu volgt eerst de beschrijving in het kader van de commando bestand.

→ **Description** “no values” voor de verschillende data typen

NO-VALUE,type,waarde

Met *type* wordt een data type genoemd en met de “no-value” *waarde* die voor dat type moet worden gebruikt. Achter waarde staat tussen rechte haken de waarde die door het programma wordt gebruikt voor dat type als er geen **NO-VALUE** commando is gevonden. Verder geldt dat alleen de hieronder genoemde data typen worden ondersteund. Als een ander data type in de ASCII bestand wordt aangegeven dan zal het programma stoppen. Het is altijd mogelijk een type toe te voegen.

In de commando bestand mogen meerdere commando's voorkomen.

De volgende commando's zijn mogelijk:

NO-VALUE,INTEGER*4,value[99999]
NO-VALUE,REAL*4,value[1.E10]
NO-VALUE,CHARACTER,value[empty]

Er wordt getest of de gegeven waarde ook werkelijk bij het type past; het programma stopt met een melding.

→

Verwerk de header van het ASCII bestand %ProcessHeader%

De header bevat allerlei informatie waarmee het mogelijk wordt om de data uit het ASCII bestand te lezen en om een shape-bestand aan te maken met polygonen.

Voor het lezen van de header zijn genoemd:

- Bepaal aantal parameters, bepaal type parameter, bepaal “no value” waarde
- Lees run-id

De module werkt als volgt:

- 1) -Lees eerste regel, de omschrijving.
- 2) -Lees gridtype (curvilinear of raster)
- 3) -Lees run-id xxx van trim-xxx.def en van trim-xxx.dat
- 4) -Lees het aantal parameters
- 5) -Behandel de parameter regels, per regel **%HandleParameter%**
- 6) -Lees tot en met de regel met DATA-BLOCK, deze regels worden als commentaar gezien en er wordt verder niets mee gedaan.

Wat de module doet is duidelijk, toch per actie een toelichting.

Ad. 1 Omschrijving

Dit is een commentaar regel die eventueel in allerlei uitvoer kan worden gebruikt. Wordt verder niet inhoudelijk gebruikt.

Ad. 2 Run-id

Zoals in het functioneel ontwerp aangegeven bevat deze regel karakters die een aanduiding zijn voor het jaar waarvoor de berekening is gedaan. Uiteindelijk worden er 4 karakters gebruikt.

Als het aantal < 4 dan worden de eerste 2 karakters gebruikt, in alle andere gevallen de eerste 4. Er wordt niet getest wat de karakters voorstellen.

Ad. 3 Aantal parameters

Het programma moet weten hoeveel regels er volgen met informatie over de parameters. tevens wordt dit getal gebruikt om de polygoon regels te kunnen verwerken. Er staat hard een getal waar in principe geen beperking aan is verbonden, het aantal parameters is vrij.

Ad. 4 Parameter regels

De k regels die volgen op de regel met het aantal parameters, waarin k het aantal is, worden per regel behandeld door de module %HandleParameter%. Deze wordt in de volgende paragraaf behandeld.

Ad. 5 Rest of commentaar regels in header

Het aantal regels is volledig vrij, het programma doet er verder niets mee, pas nadat het data blok is bereikt worden de regels weer belangrijk. Deze moeten met *** beginnen zodat het programma weet dat wat er achter staat geen functie heeft.

%HandleParameter%

Deze module behandelt, zoals gezegd, elke regel als parameter regel.

Aan de module wordt de string aangeboden die vervolgens wordt verwerkt. Het resultaat wordt in arrays opgeslagen.

De module gaat er van uit dat een regel als volgt is opgebouwd:

naam,data-type,"no-value"

Naam

Dit is de naam van de parameter die wordt gebruikt als identificatie en wel op twee manieren, n.l. als kolom naam in een shape-bestand met meerdere parameters, of voor de naam van de coverage per parameter. Het aantal karakters in de naam is vrij, bij het gebruik zijn er wel beperkingen.

Een kolom naam gebruikt de eerste 10 karakters van de parameter naam. De naam van een coverage is maximaal 13 karakters en gebruikt daarvoor de run-id dat met het _ teken aan de parameter naam wordt gekoppeld. De run-id bestaat uit 4 karakters dus kunnen er 8 karakters van de parameter naam worden gebruikt.

Als er meerdere parameters zijn dan wordt eerst getest of er geen identieke namen voorkomen, dit levert een fout op. Vervolgens wordt gekeken of het aantal karakters dat werkelijk wordt gebruikt voor alle parameters uniek is; ook hier volgt een melding als dat niet het geval is.

data-type

Hier wordt een type aangegeven dat bekend moet zijn in het programma. Deze typen zijn al genoemd bij het commando **NO-VALUE**. Voldoet een *data_type* niet aan een van de toegestane typen dan stopt het programma. Default wordt het type **REAL*4** gebruikt, dus als de positie in de regel leeg is.

"no-value"

Op deze positie in de regel komt een waarde te staan die de default no-value waarde of de waarde uit de commando bestand overschrijft. Er wordt getest of de waarde voldoet aan het data type. Als er geen waarde staat dan geldt een van de eerder genoemde waarden.

Verwerken polygon regels %APolygonLine%

Elke regel uit het data blok heeft dezelfde structuur, n.l. een set van coördinaten die de polygoon beschrijft, gevolgd door de waarden voor de parameters.

In de module %APolygonLine% wordt elke regel uit het data blok verwerkt volgens de in de pseudo code geschetste manier.

De stappen worden in deze paragraaf verder uitgewerkt.

Zoals zal blijken uit het vervolg kunnen er fouten optreden. Wordt een fout gevonden dan wordt dat gemeld maar het programma gaat wel door met inlezen en elke fout wordt ook gemeld. Als er fouten optreden dan stopt het programma na het lezen van alle polygoon regels.

Bepaal de coördinaten en de parameterwaarden %CoordPar%

Uit de string waarin de gehele regel staat worden alle waarden en lege velden gelezen. Omdat bekend is hoeveel parameterwaarden er zijn is ook bekend met hoeveel coördinaten de polygoon wordt weergegeven.

Er wordt wel getest of het aantal waarden dat wordt gebruikt voor de coördinaten even is en of er minstens 3 sets met coördinaten zijn.

Vervolgens wordt elke parameter getest:

- Is het veld leeg en moet er een "no-value" waarde worden ingevuld.
- Voldoet de waarde aan het data-type.

Wat betreft de parameters in de regel is nu alles bekend. Nu moeten de coördinaten verder behandeld worden. Na de twee tests die volgen wordt aan de set het dan eerste punt toegevoegd om te zorgen dat de polygoon ook werkelijk gesloten wordt.

Snijden lijnen van de polygoon elkaar %CheckCrossing%

Als lijnen van de polygoon elkaar snijden dan levert dat problemen op in de coverage. Daar moet dus op worden gecontroleerd. Deze controle wordt uitgevoerd in de aangegeven module. Treedt er een snijdende lijn op dan wordt dit gemeld en aangemerkt als een fout.

Sorteren van de coördinaten sets %SortOut%

Met de opdracht **COORDINATE-ORDER** wordt aangegeven op welke manier de coördinaten sets moeten worden geordend. De test op het snijden is al gedaan en als die test goed is verlopen dan wordt de module %SortOut% aangeroepen.

De mogelijkheden worden hierna beschreven.

AS-IS

Er wordt verder geen andere ordening aangebracht. De volgorde wordt gebruikt zoals deze is.

Clockwise of Anticlockwise

De coördinaten moeten in een gewenste volgorde worden aangeboden voor het maken van de polygoon. Daartoe wordt eerst getest in welke volgorde ze nu staan. Wijkt deze af van de gewenste volgorde dan moet de volgorde van de sets volledig worden omgedraaid, het laatste punt wordt het eerste en het eerste wordt het laatste.

Zoals al gemeld wordt na deze behandeling een punt aan de reeks toegevoegd, n.l. het eerste punt.

Parameters in shape-bestand %WriteShape%

In het voorgaande zijn alle voorbereidende werkzaamheden uitgevoerd. Er moet nu worden nagegaan of er ook werkelijk een shape-bestand kan worden weggeschreven en hoe dat moet gebeuren.

De eerste test is of er geen fout is opgetreden in de polygoon regels. De tweede test is of de aangevraagde storage-type mogelijk is met de constatering of de parameters werkelijk een waarde krijgen toegewezen.

De test op fouten is eenvoudig: er is bijgehouden of er op enig moment een fout was en afhankelijk van de uitkomst wordt de beslissing genomen.

De tweede test, storage-type, is iets gecompliceerder en wordt in een aparte paragraaf beschreven.

Na deze test kunnen de parameters werkelijk worden opgeslagen in de shape-bestand met de mogelijkheid van één shape-bestand of een shape-bestand per parameter.

Test op storage-type %CheckStorage%

De test wordt gebaseerd op drie gegevens:

1. Storage-type, met de opties *one-file* en *per-parameter*.
2. No-parameter-value, met de opties *no-value* en *no-polygon*.
3. Zijn er polygoonen waar parameters geen waarde hebben gekregen.

Voor de gegevens soort 2 en 3 kan voor de mogelijke combinaties worden aangegeven welk storage-type mogelijk is.

waarden optie		polygoon situatie		mogelijke opslag	
no-value	no-polygon	leeg veld	geen leeg veld	one-file	per-parameter
X	-	X	-	X	X
X	-	-	X	X	X
-	X	X	-	-	X
-	X	-	X	X	X

Komt de aangevraagde storage-type niet overeen met de mogelijke opslag dan treed er een fout op en stopt het programma met een melding.

Aanmaak shape-bestanden %WriteParameters%

Alle voorwaarden zijn vervuld om de parameters weg te schrijven. Er moet nu worden georganiseerd dat de parameterwaarden en de bijbehorende polygonen worden weggeschreven naar het juiste bestand.

De manier waarop dit moet gebeuren is mede afhankelijk van de basis routines die worden gebruikt om de bestanden aan te maken en om de gegevens weg te schrijven.

Als eerste moet de naam bekend zijn van de aan te maken bestanden. Als de optie "one-file" van het storage-type wordt gebruikt dan wordt een vast naam gebruikt, n.l. **ESTMORF**. Bij de optie "per-parameter" worden de namen gecomponeerd uit de parameter namen en de jaar identificatie zoals dat al eerder is beschreven.

Er wordt door het programma niet gecontroleerd of een bestand al bestaat; bestaat een bestand dan wordt deze overschreven. De gebruiker is verantwoordelijk voor het renamen van bestanden, in het manual van **ASCII2Shape** zal worden aangegeven hoe daar mee om te gaan.

Het organiseren van het wegschrijven wordt in de volgende paragrafen beschreven.

B The batch-bestanden

runP1.bat

```
@ECHO OFF
:
CTTY nul
DEL *.rpt
DEL OUTPUTP1\*.rpt
:
:copy uit P1
COPY ..\P1\*. *
COPY mapping.scw mapping.rel
CTTY con
:
ECHO model.mdf >inp
bot2bot <inp
:
CTTY nul
COPY output.dep OUTPUTP1
DEL bot2bot.exe
DEL model.mdf
DEL scal2.grd
DEL scal2.dep
:DEL estmorf.map
DEL mapping.*
DEL output.dep
:
:DEL EstDelta.par
COPY *.rpt OUTPUTP1
DEL *.rpt
DEL inp
CTTY con
:
:End
```

runP2.bat

```
@ECHO OFF
:
CTTY nul
DEL *.rpt
:DEL *.map
DEL *.par
CTTY con
:
```

```

ECHO trim-ts8 data file >inp
ECHO 10800      start time      >>inp
ECHO 54000      stop time       >>inp
:
ECHO 0          parameter index bodemhoogte >>inp
ECHO 1          parameter index waterlevel >>inp
ECHO 2          parameter index salinity  >>inp
ECHO 0          1 = bovenste laag, 0 = diepte gemiddeld >>inp
ECHO 3          parameter index temperature >>inp
ECHO 0          1 = bovenste laag, 0 = diepte gemiddeld >>inp
ECHO 4          parameter index tau       >>inp
ECHO 5          parameter index current  >>inp
ECHO 0          1 = bovenste laag, 0 = diepte gemiddeld >>inp
ECHO 6          parameter index inundatie >>inp
:
copy ..\P2\habimap.exe
HabiMap <inp
del inp
copy para*.pol OUTPUTP2
del para*.pol
del habimap.exe

```

runP3.bat

```

@ECHO OFF
:
CTTY nul
DEL *.rpt
:DEL *.map
DEL *.par
CTTY con
:
copy ..\P3\*. *
:
copy outputP2\param-00.pol param.pol >nul
:copy param-01.pol param.pol
ASCII2Shape
copy *.dbf OUTPUTP3
copy *.shx OUTPUTP3 >nul
copy *.shp OUTPUTP3 >nul
del *.dbf
del *.shx
del *.shp
:del param-00.pol
:
copy outputP2\param-01.pol param.pol >nul
:copy param-01.pol param.pol

```

```
ASCII2Shape
copy *.dbf OUTPUTP3
copy *.shx OUTPUTP3 >nul
copy *.shp OUTPUTP3 >nul
del *.dbf
del *.shx
del *.shp
:del param-01.pol
:
copy outputP2\param-02.pol param.pol >nul
:copy param-02.pol param.pol
ASCII2Shape
copy *.dbf OUTPUTP3
copy *.shx OUTPUTP3 >nul
copy *.shp OUTPUTP3 >nul
del *.dbf
del *.shx
del *.shp
:del param-02.pol
:
copy outputP2\param-03.pol param.pol >nul
:copy param-03.pol param.pol
ASCII2Shape
copy *.dbf OUTPUTP3
copy *.shx OUTPUTP3 >nul
copy *.shp OUTPUTP3 >nul
del *.dbf
del *.shx
del *.shp
:del param-03.pol
:
copy outputP2\param-04.pol param.pol >nul
:copy param-04.pol param.pol
ASCII2Shape
copy *.dbf OUTPUTP3
copy *.shx OUTPUTP3 >nul
copy *.shp OUTPUTP3 >nul
del *.dbf
del *.shx
del *.shp
:del param-04.pol
:
copy outputP2\param-05.pol param.pol >nul
:copy param-05.pol param.pol
ASCII2Shape
copy *.dbf OUTPUTP3
copy *.shx OUTPUTP3 >nul
```

```
copy *.shp OUTPUTP3 >nul
del *.dbf
del *.shx
del *.shp
:del param-05.pol

copy outputP2\param-06.pol param.pol >nul
:copy param-06.pol param.pol
ASCII2Shape
copy *.dbf OUTPUTP3
copy *.shx OUTPUTP3 >nul
copy *.shp OUTPUTP3 >nul
del *.dbf
del *.shx
del *.shp
:del param-06.pol
:
del param.pol
:
del ASCII2Shape.exe
del ascii2shape.pcf
```

C **ascii2shape.pcf**

estmorf/scalwes-file,param.pol
coordinate-order,CLOCKWISE
storage-type,per-parameter,FirstTry

*** nieuwe opdracht
*** storage-type,grid,corner
*** optie is corner of center, de laatste is default

no-parameter-value,no-polygon
no-value,integer*4,999
no-value,real*4,-1.E10
no-value,character,elke-waarde



WL | Delft Hydraulics

Rotterdamseweg 185
postbus 177
2600 MH Delft
telefoon 015 285 85 85
telefax 015 285 85 82
e-mail info@wldelft.nl
internet www.wldelft.nl

Rotterdamseweg 185
p.o. box 177
2600 MH Delft
The Netherlands
telephone +31 15 285 85 85
telefax +31 15 285 85 82
e-mail info@wldelft.nl
internet www.wldelft.nl

