



Sveriges
lantbruksuniversitet

Tillämpning av en markprofilmodell för hydrologiska beräkningar i avrinningsområdesskala

Application of a soil profile model for
hydrological estimations in catchment scale

Stefan Hellgren

REFERAT

Tillämpning av en markprofilmodell för hydrologiska beräkningar i avrinningsområdesskala

Stefan Hellgren

Det finns ett stort behov av att minska läckage av näringsämnen från åkermark till hav och sjöar. Genom att använda flera olika typer av modeller har det tidigare varit möjligt att beskriva läckaget i ett avrinningsområde med en minsta beräkningsenhet på delavrinningsområdesnivå. I nuläget är det istället önskvärt att modellera ett mindre avrinningsområde med möjlighet att återkoppla resultaten till respektive fält på det avrinningsområdet. Sådana modeller existerar redan men dessa är inte fullt ut kapabla att korrekt beskriva svenska förhållanden och egenskaper som markområden i vår region har.

Med ansatsen att skapa en sådan modell har SLU utformat ett projekt vilket detta arbete är första stadiet av. Modellen som förväntas skapas går under arbetsnamnet SWE-model vilket står för Soil Water Environment och är tänkt att i detta första stadium skapa förutsättningar att tillämpa modellen SOIL i avrinningsområdesskala. Under förfarandet att beskriva första steget i processen att utveckla en sådan modell anpassad för svenska förhållanden som jobbar i avrinningsområdesskala med en area kring 1-30 km² tillägnas fokus åt att beräkna transporten av vatten som rinner från de olika enheterna. Oavsett vilka processer som sker i marken efter att vatten tillförts antas det att allt vatten som rinner från varje simulerad enhet dräneras.

I första steget identifierades hydrologiska beräkningsenheter utifrån markanvändnings- och jordartskartor över studieområdet. Därefter valdes de beräkningsenheter som skulle simuleras ut med hjälp av funktioner som hämtar och omvandlar data i ett skript som skapats för detta ändamål. Slutligen aggregeras och summeras modellresultaten för varje unik enhet och för varje delavrinningsområde och även för hela avrinningsområdet.

Från resultaten är det möjligt att se en viss likhet i flödesdynamik mellan modellerad och uppmätt vattenföring. Effektivitetskoefficienten för hela simuleringsperioden har beräknats till att motsvara medelvärdet för de uppmätta värdena. Volymfelet ger en indikation på överskattning från modellens sida. Med en automatiserad kalibreringsprocess bör modellen ha möjlighet att prestera bättre.

Nyckelord: modell, hydrologisk beräkningsenhet, avrinningsområdesskala, delavrinningsområde, näringsämne, läckage

*Institutionen för vatten och miljö, Sveriges lantbruksuniversitet
Vallvägen 3, SE-756 51 Uppsala
ISSN 1401-5765*

ABSTRACT

Application of a soil profile model for hydrological estimations in catchment scale

Stefan Hellgren

There is a great need to reduce nutrient leaching from arable land into lakes and oceans. By using several different types of models it has previously been possible to describe nutrient losses in a catchment area with a minimum unit of sub-catchment level. At present, it is instead desirable to model a smaller catchment with an opportunity to re-connect the results to the corresponding fields in the catchment. Such models already exist but they are not fully able to properly describe Swedish conditions and land characteristics in our region.

With the approach of creating such a model, SLU has developed a project with this work as its first stage. The model is expected to be created under the working name SWE-model which stands for Soil Water Environment and is in this first stage supposed to apply the SOIL model in catchment scale. During the procedure to describe the first step in the process of developing such a model adapted to Swedish conditions and which works in the catchment scale with an area of about 10-30 km², focus has been set on calculating the transport of water flow from different hydrological response units. Regardless of the processes occurring in the soil after the water has been added, it is assumed that all the water which flows from each simulated unit is drained.

In the first step the hydrologic response units were identified based on land use and soil type in the study area. With the help of a script with functions that retrieve and transform data, certain units were chosen for simulation. The script was also created in this project. Finally, the model results were aggregated and summarized for each unique unit, for each sub-catchment, and also for the whole catchment.

From the results it is possible to see similarities in the flow dynamics between modeled and measured data. The efficiency coefficient has been calculated to correspond to the mean of the measured values for the whole simulation period. With an automated calibration process the model should be able to perform better. The volume error gives an indication of overestimation from the model.

Keywords: model, hydrological response unit, catchment scale, nutrient leakage

Department of Aquatic Sciences and Assessment, Swedish University of Agricultural Sciences

Vallvägen 3, SE-756 51 Uppsala

ISSN 1401-5765

FÖRORD

Detta examensarbete omfattar 30 hp och har utförts på Institutionen för vatten och miljö, Sveriges lantbruksuniversitet (SLU). Det är den avslutande delen på civilingenjörsprogrammet miljö- och vattenteknik på Uppsala Universitet. Handledare för examensarbetet var Faruk Djodjic, Institutionen för vatten och miljö, Sveriges lantbruksuniversitet, examinerator var Allan Rodhe, Institutionen för geovetenskaper, Uppsala Universitet och ämnesgranskare var Mats Wallin, Institutionen för vatten och miljö, Sveriges lantbruksuniversitet.

Främst vill jag tacka Faruk Djodjic som hjälpt mig att överblicka projektet när jag tenderat att fastna i detaljer samt Hampus Markensten som hjälpt till med hanteringen av modellen och som genom diskussioner påskyndat arbetet avsevärt. Jag vill även tacka Peter Larsson för diskussioner kring den problematik som ibland uppstod, Mats Wallin för läsning av rapporten samt Jakob Nisell och Jakob Lagerstedt för ämnesspecifika råd. Också tack till alla som varit inblandade i att förhöja stämningen mellan arbetstimmarna.

Uppsala, November 2010

Stefan Hellgren

Copyright © Stefan Hellgren och Institutionen för vatten och miljö, Sveriges lantbruksuniversitet

UPTEC W 10 034, ISSN 1401-5765

Tryckt hos Institutionen för geovetenskaper, Geotryckeriet, Uppsala universitet, Uppsala, 2010

POPULÄRVETENSKAPLIG SAMMANFATTNING

Tillämpning av en markprofilmodell för hydrologiska beräkningar i avrinningsområdesskala

Stefan Hellgren

Läckage av näringsämnen är ett stort problem såväl globalt som i Sverige och det finns ett stort behov av att minska läckaget från åkermark till hav och sjöar. Näringsämnen används i jordbruk för att stimulera tillväxt av grödor och enligt de senaste beräkningarna är jordbrukets läckage en av de viktigaste källorna till övergödningen och algbloomingar i Östersjön. Näringsämnens transport sker med vattenflöden och för att uppskatta koncentrationen av näringsämnen från avrinningsområden är det viktigt att ha goda kunskaper om transporten av vatten på det området. De processer som sker i marken med avseende på näringsämnen och vatten är många och komplexa och för att uppskatta dessa finns ett stort behov av att tillämpa modeller.

SLU vattenNAV är en ny enhet för olika institutioner på SLU som för närvarande har lagt fokus på tre beräkningsmodeller som i olika skalor beskriver läckage och transport av näringsämnen och vatten. Simuleringarna går till så att modellerna SOILNDB (kväve) eller ICECREAM (fosfor) först beräknar läckagekoefficienter för jordbruksmark. Dessa modeller jobbar i markprofilskala och fältskala och genom att använda dessa läckagekoefficienter i källfördelningsmodellen FyrisNP beskrivs transport och förluster av näringsämnen i avrinningsområdesskala. Den minsta beräkningsenheten som simuleras är genom denna procedur på delavrinningsområdesnivå. Detta tillvägagångssätt har dock sina begränsningar. I nuläget är det istället önskvärt att modellera ett mindre avrinningsområde med möjlighet att återkoppla resultaten till respektive fält inom avrinningsområdet. På det sättet är det möjligt att optimera användningen av näringsämnen för de olika fälten på åkermarken och genomföra lämpliga åtgärder där det finns behov av detta. Sådana modeller finns redan men dessa är inte kapabla att korrekt beskriva svenska förhållanden och egenskaper som markområden i vår region har.

För att skapa en sådan modell har SLU utformat ett projekt med flera aktiviteter av vilka detta examensarbete är den första och som fokuserar på utveckling och programmering av en första prototyp. Modellen som förväntas skapas går under arbetsnamnet SWE-model vilket står för Soil Water Environment och är tänkt att i ett första skede skapa förutsättningar att tillämpa modellen SOIL i avrinningsområdesskala. De viktigaste faktorerna som styr vattenflöden i modellen är kopplade till vegetationen på marken och markens fysiska egenskaper som till exempel konduktivitet och vattenhållande kapacitet. Modellen SOIL beräknar i dagsläget vatten- och energiflöden för en specifik kombination av vegetation och jordart. I ett avrinningsområde finns ett stort antal platser med unika sådana kombinationer. Med hjälp av lämpliga kartverktyg kan dessa platser identifieras och

sedan beskrivas med hjälp av parametrar som hämtas från en databas. Minsta enheter med unika egenskaper med avseende på jordart och vegetation benämns även som hydrologiska beräkningsenheter. Därför skapas inom varje delavrinningsområde hydrologiska beräkningsenheter med avseende på markanvändning och jordartstyp vilka sedan aggregeras och summeras både på delavrinningsnivå och för hela avrinningsområdet.

Under förfarandet att beskriva första steget i processen att utveckla en modell anpassad för svenska förhållanden och som jobbar i avrinningsområdesskala med en area kring 1-30 km², har fokus lagts på att beräkna transporten av vatten som rinner från de olika beräkningsenheterna. Oavsett vilka processer som sker i marken efter att vatten tillförts antas det att allt vatten som rinner från varje simulerad enhet dräneras och når utloppet av avrinningsområdet.

För att hämta egenskaperna för de hydrologiska beräkningsenheterna från kartor och omvandla dessa till indata har det genom programmering skapats ett skript för att automatisera processer. Skriptet har som funktion att utifrån varje beräkningsenhetens egenskaper hämta värden från databaser och ange dessa i indata till SOIL. För varje beräkningsenhet sker en simulering i SOIL. Därefter aggregeras och summeras utdata från varje simulering för att beskriva hela avrinningsområdets avrinning.

Från resultaten är det möjligt att se en viss följsamhet mellan de ritade diagrammen för den modellerade respektive uppmätta vattenföringen. Effektivitetskoefficienten för de modellerade värdena har beräknats till att motsvara medelvärdet för de uppmätta värdena för hela simuleringsperioden. Modellen tenderar att överskatta flödets volym vilket antagligen beror av förenklingar i beskrivningar av olika processer. Genom att tillämpa en automatiserad kalibrering borde modellen ha möjlighet att prestera bättre.

INNEHÅLLSFÖRTECKNING

REFERAT	I
ABSTRACT	II
FÖRORD.....	III
POPULÄRVETENSKAPLIG SAMMANFATTNING	IV
TERMINOLOGI OCH FÖRKLARINGAR.....	VIII
1 INLEDNING.....	1
1.1 BAKGRUND	1
1.2 SYFTE.....	3
1.3 SPECIFIKA MÅL.....	4
1.4 ARBETSGÅNG	4
2 STUDIEOMRÅDE.....	6
2.1 BESKRIVNING AV OMRÅDET	6
2.2 KARTOR	7
2.3 IDENTIFIERING AV BERÄKNINGSENHETER.....	8
3 INDATA OCH PARAMETRISERING	11
3.1 SOIL	11
3.2 MODELLUPPSÄTTNING.....	12
3.2.1 Inställningsfil	12
3.2.2 Jordartsinformation.....	12
3.2.3 Grödinformation	12
3.3 INDATA.....	12
3.3.1 Klimatbeskrivning	12
3.3.2 Inställningsfiler	13
3.3.3 Jordartsbeskrivning.....	13
3.3.4 Grödbeskrivning	13
3.4 AUTOMATISK GENERERING AV INDATAFILER.....	13
3.5 VAL AV BERÄKNINGSENHETER.....	15
3.6 INSTRUKTIONSFILER.....	15
4 SIMULERINGSPROCESSEN	17

4.1	AGGREGERING	17
4.2	UTVÄRDERING AV MODELLRESULTAT	18
4.2.1	Hantering av utdata	18
4.2.2	Metod för kalibrering och parameterinställningar	18
4.2.3	Nash-Sutcliffe's effektivitetskoefficient	18
4.2.4	Volymfel	18
5	RESULTAT	19
5.1	GENERERADE BERÄKNINGSENHETER	19
5.2	AVRINNINGEN	20
5.2.1	Jämförelse med uppmätta värden.....	20
5.2.2	Jämförelse mellan beräkningsenheter	22
6	DISKUSSION	25
7	SLUTSATS OCH VIDAREUTVECKLING	27
	REFERENSER	29
	BILAGA 1 - SKRIPT	A
	BILAGA 2 - INSTÄLLNINGSFILN.....	N
	BILAGA 3 - JORDARTSINFORMATION.....	O
	BILAGA 4 - GRÖDINFORMATION	P
	BILAGA 5 - INSTRUKTIONSFILN ID.....	R

TERMINOLOGI OCH FÖRKLARINGAR

Hydrologisk beräkningsenhet - En del av det aktuella området som antas besitta unika och homogena egenskaper och som appliceras på modelleringar av hydrologi.

Polygon - Geometrisk plan figur som begränsas av slutna räta linjer. (MathWorld, 2010)

Block - Ett avgränsat område med åkermark eller betesmark. Ett block avgränsas av fasta gränser som till exempel diken och vägar. Ett block är endast tillåtet att brukas av en brukare. (Jordbruksverket, 2009)

Skifte - Skiften är belägna innanför blockets gränser. En jordbrukare odlar en specifik gröda på respektive skifte inom blocket. (Jordbruksverket, 2009)

ArcInfo - Ett program med applikationer som tillåter användaren att lösa avancerade problem relaterade till geografiska informationssystem genom editering av data, geografiska analyser av kartor och visualisering av dessa (esri, 2010).

Python - Ett objektorienterat programmeringsspråk som är gratis att använda och har öppen källkod. Det är ett operativsystemsberoende språk som gör att det går att tillämpa på de flesta plattformar som finns tillgängliga idag. (Python, 2010)

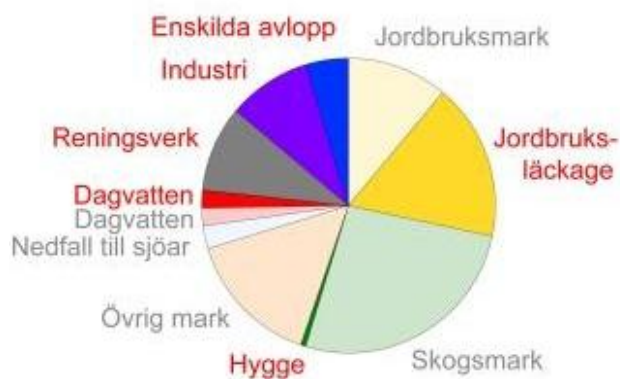
1 INLEDNING

1.1 BAKGRUND

Det finns ett stort behov av att minska läckaget av näringsämnen från åkermark till hav och sjöar. En förhöjd transport av kväve och fosfor från avrinningsområden på åkermark bidrar till eutrofiering av vattenrecipienter, det vill säga en ökad biologisk produktion med tillhörande negativa effekter i dessa akvatiska ekosystem (Sharpley et al., 1987). Enligt de senaste beräkningarna är jordbrukets läckage en av de viktigaste källorna, med avseende på Sveriges bidrag, till övergödningen och algbloomningar i Östersjön (figur 1 och figur 2). Näringsämnen används i jordbruk för att stimulera tillväxt av grödor. En viss del av näringsämnesläckaget från åkermark kommer alltid att förekomma eftersom det alltid finns ett bakgrundsläckage men en del faktorer går att påverka och därmed reducera läckaget (Jordbruksverket, 2009). Näringsämnens transport sker med vattenflöden och för att uppskatta koncentrationen av näringsämnen från avrinningsområden är det viktigt att ha goda kunskaper om transporten av vatten från området. De processer som sker i marken med avseende på näringsämnen och vatten är många och komplexa och för att prediktera dessa finns ett stort behov av modelltillämpning. Om modellerna är kapabla att beräkna förändringar av olika näringsämnens koncentrationer går det dessutom att uppskatta effekter av eventuella åtgärder i området (Johnes & Heathwaite, 1997).



Figur 1. Källor till kväveutsläpp från Sverige. Rödmarkerade källor är en följd av mänsklig påverkan. Jordbruksmark representerar utsläpp innan gödsling. (Jordbruksverket, 2009)



Figur 2. Källor till fosforutsläpp från Sverige. Rödmarkerade källor är en följd av mänsklig påverkan. Jordbruksmark representerar utsläpp innan gödsling. (Jordbruksverket, 2009)

Vatten som befinner sig i ett avrinningsområde kan transporteras till utloppspunkten på olika sätt. När vattnet ankommer till avrinningsområdet i form av nederbörd infiltrerar det marken om jorden i marken inte är mättad eller om nederbördsintensiteten inte är större än infiltrationskapaciteten. I marken kan vattnet tillfälligt lagras som markvatten eller söka sig nedåt till grundvattnet. Om infiltrationskapaciteten i marken är låg kan det bildas ytavrinning. Dock är det oftast på grund av att grundvattenytan stigit till marknivå som regnvattnet hindras från att infiltrera. Dränering kan ske både av grundvattnet och ytvattnet och detta kan ske naturligt eller artificiellt genom dräneringsrör i marken eller genom dikning (Grip & Rodhe, 2003).

SLU vattenNAV är en ny enhet för olika institutioner på SLU som samverkar för att modellera och analysera vattenkvalitet i avrinningsområden. Enheten inriktar sig på att förvalta, utveckla och tillämpa modeller för beräkningar av transport, läckage, retention och källfördelningar av kväve och fosfor (SLU vattenNAV, 2010). Fokus har legat på tre beräkningsmodeller som i olika skalor beskriver läckage och transport av näringsämnen och vatten. Simuleringsproceduren är för tillfället att först beräkna läckagekoefficienter för jordbruksmark med hjälp av modellerna SOILNDB (för kväve) eller ICECREAM (för fosfor). Dessa modeller jobbar i markprofilskala och fältskala och genom att implementera dessa läckagekoefficienter i källfördelningsmodellen FyrisNP beskrivs transport och förluster av näringsämnen i avrinningsområdesskala. Den minsta beräkningsenhet som simuleras vid dessa beräkningar är på delavrinningsområdesnivå. Denna procedur har visat sig särskilt användbar för avrinningsområden med en area större än 30 km² vilket till stor del beror på befintlig indata och tillgång till relevant miljöövervakningsdata för både vattenföring och vattenkvalitet. Detta tillvägagångssätt har dock sina begränsningar i de fall där det finns bättre indata för mindre områden med en area kring 1-30 km². I en studie beskrev Sheshukov et al. (2009) hur en högre upplösning på jord- och vegetationsdata medför en mer noggrann beskrivning av området men samtidigt hur antalet

beräkningsenheter ökar och därmed bidrar till en mer komplex simuleringsprocess. I nuläget är det istället önskvärt att modellera ett mindre avrinningsområde med möjlighet att återkoppla resultaten till respektive fält inom avrinningsområdet. På det sättet är det möjligt att optimera användningen av näringsämnen för de olika fälten belägna på åkermarken och genomföra lämpliga åtgärder där det finns behov av detta. Sådana modeller existerar redan och SWAT model är till exempel en modell som definierar hydrologiska beräkningsenheter i ungefär samma skala men som även tar även hänsyn till lutningen i området. Den beskriver dock processer med parametervärden som är dåligt verifierade i Sverige vilket begränsar möjligheten att beskriva svenska förhållanden och egenskaper som markområden i vår region har (Anlauf, 2008).

Med ansatsen att skapa en sådan modell har SLU utformat ett projekt som förväntas pågå i tre år med flera aktiviteter av vilka detta examensarbete är den första med fokus på utveckling och programmering av en första prototyp. Modellen som förväntas skapas går under arbetsnamnet SWE-model vilket står för Soil Water Environment och är tänkt att i detta första stadium skapa förutsättningar att tillämpa modellen SOIL i avrinningsområdesskala. De viktigaste faktorerna som styr vatten- och energiflöden i modellen är kopplade till vegetationstäckor som till exempel jordbruksgröda och skog samt markens fysiska egenskaper som till exempel hydraulisk konduktivitet och vattenhållande kapacitet. I dess nuvarande form beräknar SOIL vatten- och energiflöden för en specifik kombination av vegetationstäckor och jordart. I ett avrinningsområde finns ett stort antal sådana kombinationer som med hjälp av lämpliga kartverktyg kan identifieras och sedan beskrivas med hjälp av specifika parametrar. Minsta enheter med unika egenskaper med avseende på jordart och vegetation kommer i detta arbete att benämnas som hydrologiska beräkningsenheter. Således skapas inom varje delavrinningsområde hydrologiska beräkningsenheter med avseende på markanvändning och jordartstyp vilka sedan aggregeras och summeras på både delavrinningsnivå och även för hela avrinningsområdet.

Simuleringarna har genomförts i ett avrinningsområde med dokumenterade data över uppmätta flöden och därefter har en utvärdering av resultatet genomförts och därmed en bedömning av modelleffektiviteten. Simuleringarna har genomförts för en tidsperiod på sex år.

1.2 SYFTE

- Utveckla och tillämpa en prototyp för SWE-model för att möjliggöra tillämpning av en fältmodell för modellering av hydrologi och näringsämnestransporter i avrinningsområdesskala.
- Undersöka om det går att uppskatta den totala avrinningen från ett avrinningsområde genom att summera den totala vattenföringen från varje enhet som är unik med avseende på markanvändning och jordartstyp inom avrinningsområdet.

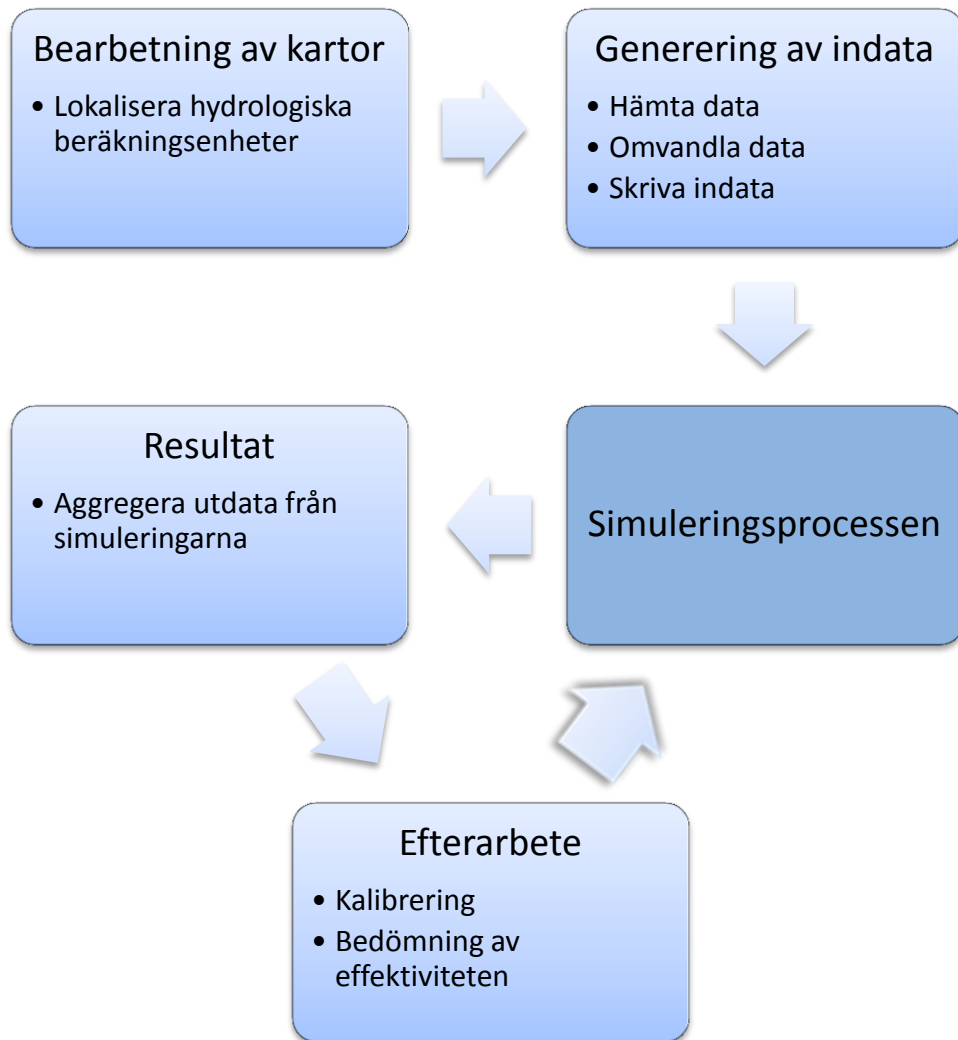
1.3 SPECIFIKA MÅL

- Automatisera genereringen av hydrologiska beräkningsenheter.
- Summera resultaten och därefter genomföra en utvärdering av resultaten och därmed en uppskattning av modellens effektivitet.
- Diskutera lösningar till eventuella problem som kan uppstå med modellen och därmed svårigheter med ansatsen till projektet.

1.4 ARBETSGÅNG

Under förfarandet att beskriva första steget i processen att utveckla en modell anpassad för svenska förhållanden som jobbar i avrinningsområdesskala med en area kring 1-30 km² har fokus lagts på att beräkna transporten av vatten som rinner från de olika beräkningsenheterna. Oavsett vilka processer som sker i marken efter att vatten tillförts antas det att allt vatten som rinner från varje simulerad enhet dräneras och når utloppet av avrinningsområdet. Indata till beräkningsmodellen SOIL beskrivs i ett flertal filer med värden som hämtas från en på förhand skapad parameterdatabas. För att kunna specificera innehållet i dessa filer behövs information om under vilka förhållanden simuleringarna ska genomföras samt vilka parametervärden som ska anges. Parametervärdena varierar med avrinningsområdets mark- och vegetationsegenskaper. Dessa egenskaper har tillhandahållits i form av polygonattribut i ett flertal kartor.

Till att börja med behöver de olika hydrologiska beräkningsenheterna lokaliseras vilket genomförs genom att bearbeta kartorna med hjälp av lämpligt kartprogram. När beräkningsenheterna har lokaliserats behöver egenskaperna för varje beräkningsenhet extraheras och omvandlas till indata för SOIL. För varje beräkningsenhet genomförs en simulering i SOIL och därmed skapas separat utdata för varje enhet. Slutligen måste därför resultaten från alla simuleringarna aggregeras för att beskriva hela avrinningsområdet. Efter sammanfogningen av resultaten genomförs en effektivitetsberäkning och kalibrering av modellen i syfte att förbättra resultaten (figur 3).



Figur 3. Arbetsflöde för de olika delarna i modellutvecklingen. Kalibreringsprocessen är mycket begränsad i denna studie.

2 STUDIEOMRÅDE

2.1 BESKRIVNING AV OMRÅDET

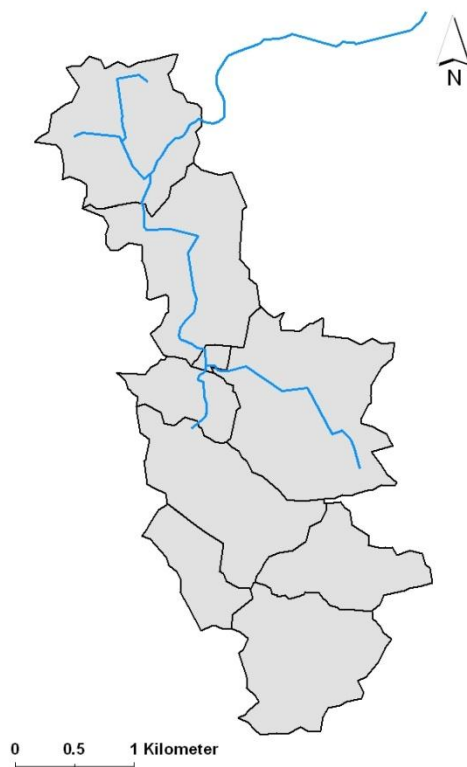
Studieområdet (figur 4) är ett litet avrinningsområde på ungefär 8 km² och ingår i det nationella miljöövervakningsprogrammet ”Typområden på jordbruksmark”. För detta område finns en långtidsserie med vattenföring uppmätt med ett triangulärt överfall från en mätstation (SMHI) i utloppspunkten. Uppmätta data över klimatförhållanden hämtades från en klimatstation ungefär 20 km från området. Avrinningsområdet är indelat i nio delavrinningsområden utifrån höjddata och med hänsyn tagen till befintliga provpunkter för analys av vattenkvalitet.



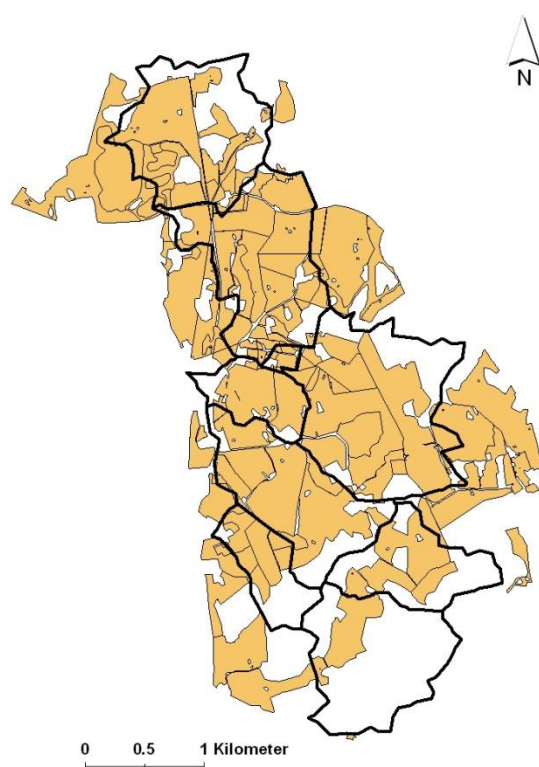
Figur 4. Markeringen E23 visar studieområdets läge i Sverige

2.2 KARTOR

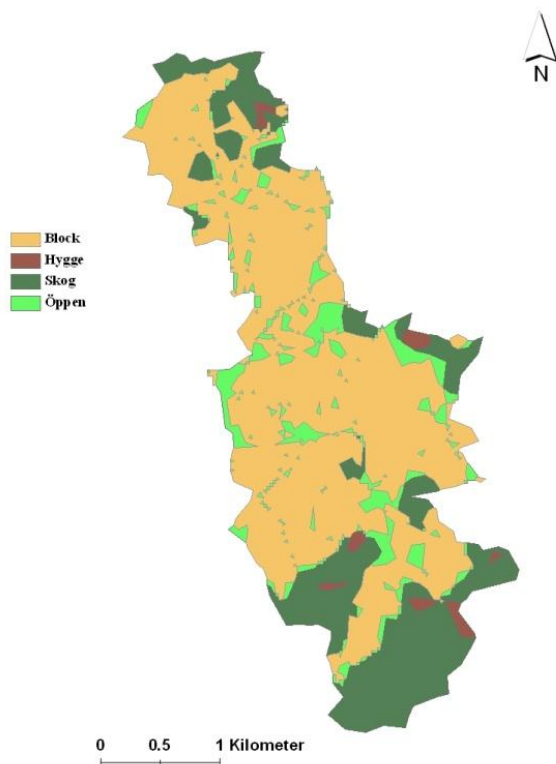
Fyra kartor över studieområdet har använts i detta projekt (figur 5-8). Dessa kartor visar området indelat i polygoner separerade efter deras egenskaper. Jordbruksblockskartan täcker endast delvis avrinningsområdet eftersom det finns andra typer av markanvändning i området. Markanvändningsdata finns för hela avrinningsområdet och inkluderar även åkermark. Jordbruksblockskartan behövs dock också eftersom den specificerar de olika blockens positioner och deras respektive id-nummer. Kartan över jordarter täcker hela avrinningsområdet.



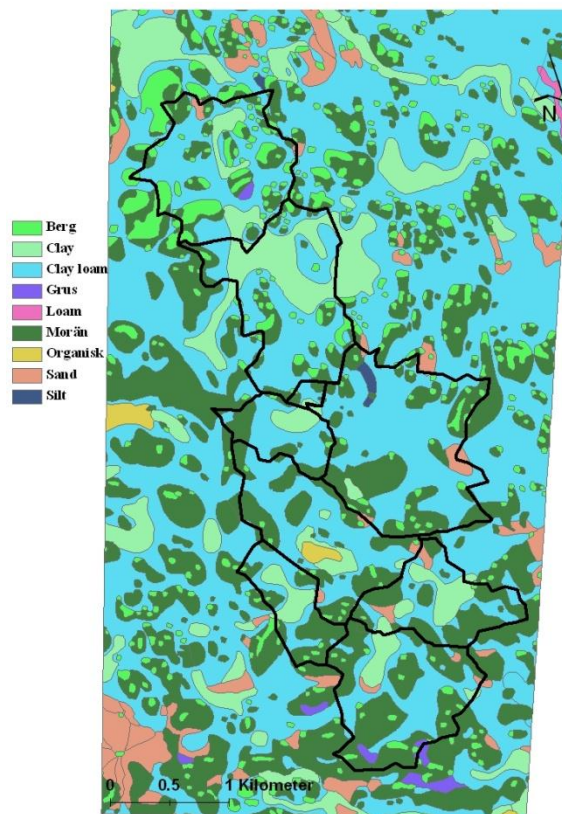
Figur 5. Delavrinningsområden i avrinningsområdet



Figur 6. Jordbruksblocken i avrinningsområdet (blockdata från Jordbruksverket, 2005)



Figur 7. Markanvändning inom avrinningsområdet (Lantmäteriet, röda kartan)



Figur 8. Jordarter inom avrinningsområdet (kombination av SGU's jordartskarta och SLU's punktdata)

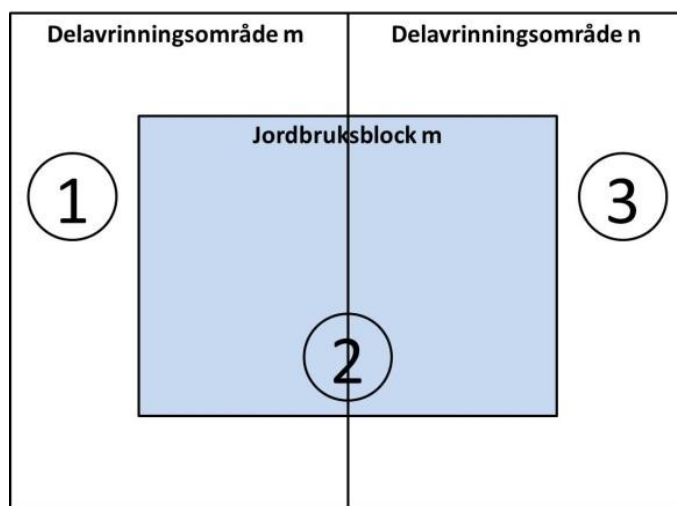
2.3 IDENTIFIERING AV BERÄKNINGSENHETER

För varje hydrologisk beräkningsenhet inom ett delavrinningsområde utförs en simulering. För att möjliggöra identifiering av beräkningsenheter måste kartorna bearbetas på ett sådant sätt att det går att extrahera information om vilket delavrinningsområde de tillhör samt vilken markanvändning, gröda och jordart som existerar på varje unik plats inom avrinningsområdet. För att genomföra detta har det i det här arbetet valts att sammanfoga kartorna till endast en karta med den för modellen viktiga informationen. För detta syfte har programmet ArcInfo använts.

Blockdata från Jordbruksverket anges i fältdatabasen och används för att mer i detalj definiera beräkningsenheter på jordbruksmark eftersom fokus i arbetet ligger på just jordbruksmark. Ett jordbruksblock är ett sammanhängande markområde som har relativt beständig indelning från år till år. Eftersom blocken är utformade så att de begränsas av till exempel vägar, vattendrag, diken och skog så har det i detta projekt antagits att avgränsningarna för blocken även utgör avgränsningar för hydrologiska enheter. Däremot

består oftast blocken av flera skiften med olika grödor och varje unik kombination av gröda och jordart inom blockets gränser har tilldelats egen beräkningsenhet. För att till viss del begränsa antalet beräkningsenheter har det antagits att endast en jordart förekommer inom varje jordbruksblock. Den definieras efter den jordart som upptar störst area inom respektive jordbruksblock.

För att säkerställa polygonernas positioner definieras först av allt koordinatsystemet som använts för att skapa kartorna. I ArcInfo finns det en funktion som heter "intersection" och som utför en skärning mellan två kartor. Då bevaras endast den yta som överlappas av båda kartorna och på den ytan sparas egenskaperna från båda ursprungskartorna. Genom att genomföra geometriska skärningar mellan kartorna skapas automatiskt nya polygoner utifrån de polygoner som kartorna ursprungligen innehöll. Varje skärning utförs mellan två kartor åt gången för att möjliggöra areaberäkningar av polygonerna på kartorna mellan skärningarna. När till exempel skärningen mellan delavrinningsområdeskartan och jordbruksblockskartan genomförs kommer det därför att skapas flera polygoner för varje jordbruksblock (figur 9). Eftersom respektive jordbruksblock som delas upp består av samma mark- och vegetationsegenskaper definieras alla polygoner som tillhör samma jordbruksblock som en hydrologisk beräkningsenhet. Denna beräkningsenhet kan dock komma att delas in i flera beräkningsenheter senare beroende på hur många skiften som är belägna på jordbruksblocket, vilket kontrolleras i fältdatabasen. Det enda som skiljer polygonerna med jordbruksblock är alltså vilket delavrinningsområde de tillhör. Eftersom det är nödvändigt att veta hur stor area varje polygon upptar på respektive delavrinningsområde beräknas detta i ArcInfo efter varje skärning.



Figur 9. Exempel på hur en jordbruksblockspolygon ser ut efter skärning mellan delavrinningsområdeskartan och jordbruksblockskartan. 1-3 definierar olika beräkningsenheter

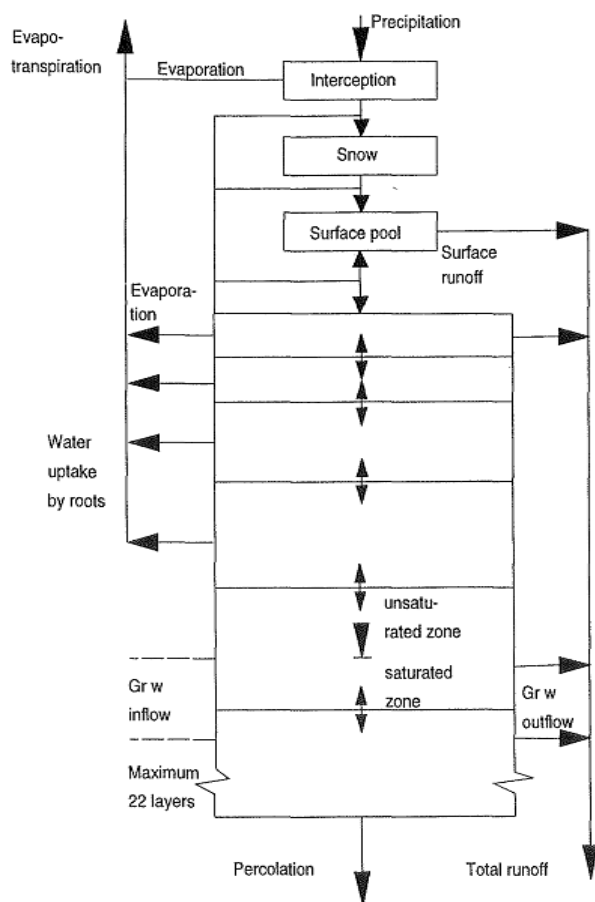
När det kommer till behandlingen av jordbruksblockskartan behöver en särskild typ av skärning utföras som i ArcInfo heter "Identity". Denna funktion sparar arean för en av kartorna oavsett om den överlappar den andra kartan eller inte. Detta genomförs på grund av att kartan med jordbruksblock inte upptar hela arean av avrinningsområdet. Resultatet från skärningen mellan delavrinningsområdeskartan och jordbruksblockskartan resulterar i en ny karta bestående av egenskaper från de båda ursprungskartorna. På motsvarande sätt skapas ytterligare nya kartor när nästa skärning utförs mellan den nya kartan och markanvändningskartan respektive jordartskartan. Slutligen har egenskaperna för dessa fyra kartor samlats i en enda karta uppdelad i polygoner med unika egenskaper med avseende på gröda, markanvändning och jordart. Proceduren avslutas genom att areorna för alla polygoner beräknas.

3 INDATA OCH PARAMETRISERING

Efter att den redigerade kartan har skapats ska informationen som den innehåller förflyttas till SOIL i form av indata. Detta har varit möjligt att genomföra genom att kontrollera på vilken form som SOIL läser in indatafiler och därefter automatisera processen som hämtar informationen från kartan och efteråt skriva ut informationen på den formen.

3.1 SOIL

SOIL är en processbaserad modell som främst används för att beräkna transport av vatten och energi i markprofilen. Den utvecklades initialt för att simulera förhållanden i markområden med skog och anpassades senare för att i stället kunna simulera alla typer av vegetationer. Modellens struktur är utformad som en djupprofil i marken och den behandlar utbytet mellan mark och atmosfär genom beräkningar av snösmältning, interception, nederbörd och evapotranspiration (Figur 10). Den centrala delen av modellen utgörs av två differentialekvationer: massbalans samt Darcys lag eller Fouriers lag. Användaren kan själv definiera vilken typ av utdata som ska skrivas ut. SOIL delar upp vattentransporten i ytavrinning, dränering och perkolation till grundvattnet (Jansson, 1991).



Figur 10. Struktur över vattentransporten i SOIL. Här framgår även de olika lagren som SOIL simulerar mellan (Jansson, 1991)

3.2 MODELLUPPSÄTTNING

SOIL läser under simuleringen in ett antal filer som definierar egenskaperna för den aktuella beräkningsenheten inom avrinningsområdet som simuleringen genomförs på. Tre av dessa filer innehåller information som påverkar vattentransporten till olika delar.

3.2.1 Inställningsfil

Inställningsfilen är en typ av huvudfil som SOIL läser in först och som definierar vilka andra filer som också ska läsas in under körningen (se nedan). Den innehåller information över bland annat parametervärden vilka till exempel beskriver vilka initiala förhållanden som ska föreligga på den unika platsen, antalet markskikt och deras respektive tjocklek som SOIL ska simulera samt korrektionskoefficienter för fysikaliska data. Korrektionskoefficienter används för att korrigera inbyggda fel i till exempel uppmätta klimatdata. För att specificera vilka parametrar som ska läsas in innehåller inställningsfilen även ett antal parameterväljare som på olika sätt kombineras för att beskriva den hydrologiska beräkningsenheten på ett så bra sätt som möjligt. I inställningsfilen definieras även under vilken tidsperiod simuleringen ska köras, information om var den aktuella beräkningsenheten befinner sig och vilken typ av utdata som ska skrivas ut.

3.2.2 Jordartsinformation

SOIL läser under simuleringen in en fil innehållandes jordartsegenskaper för den aktuella platsen. Jordartsegenskaperna specificeras för varje lager som angetts i inställningsfilen. I denna fil anges bland annat jordartens porositet, vattenhalt vid vissningsgränsen och mättad konduktivitet.

3.2.3 Grödinformation

För varje gröda som existerar på avrinningsområdet läser SOIL in en fil med gröddata under simuleringarna. Denna fil innehåller specifik information om den aktuella grödans egenskaper och definieras i en tidsserie med ett antal värden för varje år. Exempel på grödegenskaper som anges är rotdjup, ytmotstånd och bladyteindex. Dessa parametrar definieras med ett antal värden per år vilka SOIL interpolerar mellan under körningarna.

3.3 INDATA

Indata definieras efter vilka egenskaper beräkningsenheten har samt under vilka förhållanden som föreligger i området. SOIL har genomfört simuleringarna med dygn som tidssteg.

3.3.1 Klimatbeskrivning

För varje simulering som ska utföras måste klimatet i simuleringsområdet specificeras. All klimatdata har hämtats från närliggande klimatstationer från SMHI. Klimatfilen består av dygnsdata över vindhastighet, solstrålning, medeltemperatur, nederbörd och luftfuktighet över hela tidsserien. Dessa värden anges så noggrant som möjligt för samma tidsperiod som

den som angetts i inställningsfilen. Om värden saknas kan SOIL hantera detta på olika sätt. Parametrar som angetts som kontinuerliga ersätter saknade värden i klimatfilen genom att interpolera mellan närliggande värden. Om parametrar istället angetts vara i pulsform sätter SOIL de parametervärden som saknas i klimatfilen automatiskt till noll.

3.3.2 Inställningsfiler

I inställningsfilerna definieras de grödspecifika parametrarnas initiala värden. Dessa anges likadant som de i grödbeskrivningen för att inte erhålla för stora skillnader mellan initialskedet i simuleringen och fortsättningen på simuleringen. I bilaga 2, figur 21 visas ett utdrag på de 205 parametervärden som är möjliga att specificera och däribland även de grödspecifika parametervärdena. I bilaga 2, figur 22 redovisas hur de olika parameterväljarna angetts.

3.3.3 Jordartsbeskrivning

Parametervärden för jordarter hämtas från parameterdatabasen. För simuleringarna i detta arbete har marken delats upp i fyra skikt med olika tjocklek. För att förenkla simuleringsprocessen har parametrarna definierats likadant för var och ett av dessa skikt (bilaga 3, figur 23).

3.3.4 Grödbeskrivning

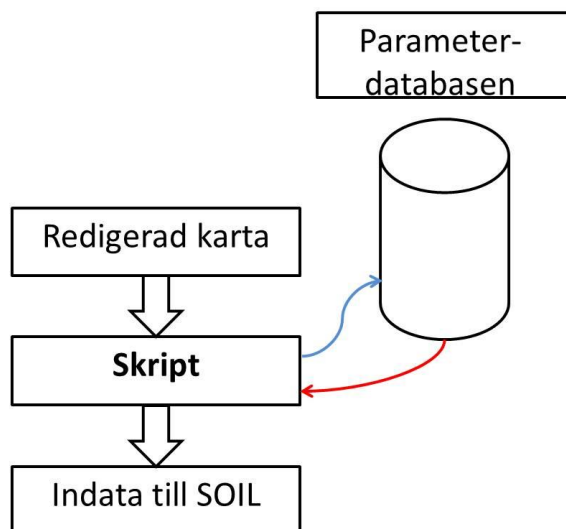
De parametrar som specificeras i grödfilen är rotdjup, bladtyeindex, ytmotstånd, förflyttningshöjd och råhetslängd. I denna fil anges även en parameter som kallas ”dagnummer”. Under denna parameter anges för vilka dagar som de grödspecifika parametervärdena ska sättas under ett år. Parametervärden för grödegenskaper är definierade i parameterdatabasen. För områden med annan markanvändning än jordbruk skapas grödfiler med den typ av vegetation som kan antas finnas på dessa områden. Se bilaga 4 för utdrag ur parameterdatabasen över de aktuella gröd- och markanvändningarna och ett exempel på en indatafil med grödinformation.

3.4 AUTOMATISK GENERERING AV INDATAFILER

Med hjälp av programmeringsspråket Python har det skapats ett skript vars primära uppgift är att hämta information från alla beräkningsenheter i den redigerade kartan, välja ut vilka som ska simuleras och omvandla dessa till indata i en form som är läsbar för SOIL. Skriptet består av ett flertal funktioner som på olika sätt hämtar, omvandlar och skriver information. I bilaga 1 bifogas koden som använts för att genomföra detta.

I det här arbetet har programmeringsspråket Python 2.5 använts för att generera inställningsfiler och filer innehållandes jordartsegenskaper för den aktuella platsen som SOIL läser in under körningarna. Filer med gröddata och klimatdata har genererats manuellt.

Parametervärden som hämtas och läses in i inställningsfilen och filen med jordartsegenskaper finns dokumenterade i en databas (figur 11). Denna innehåller ett flertal databasobjekt med parametervärden för olika markanvändningar och jordarter.



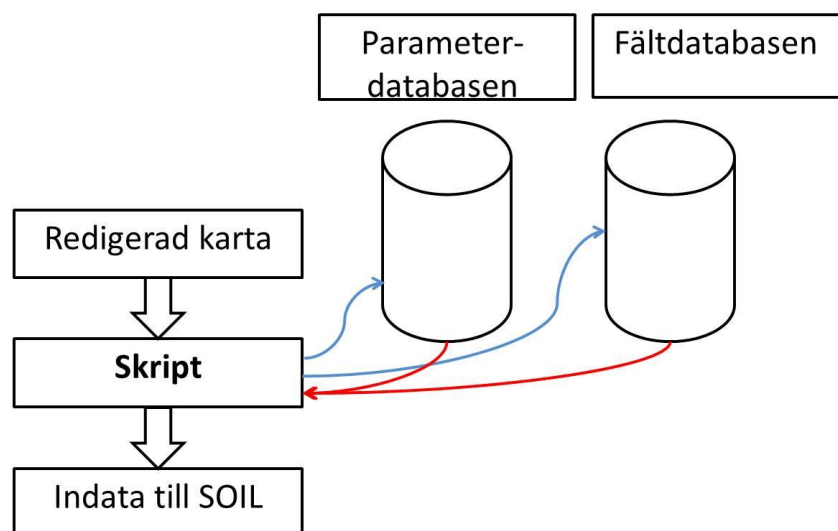
Figur 11. Illustrering av hur skriptet hämtar information från parameterdatabasen

Den första funktionen i skriptet läser in attributen från den redigerade kartan för varje separat beräkningsenhet och lagrar informationen i en temporär lista. Nästa funktion i skriptet ändrar därefter alla grödnamn och jordartsnamn i listan till id-nummer eftersom skriptet i ett senare skede använder sig av id-nummer för identifiering.

Alla beräkningsenheter inom ett specifikt jordbruksblock skiljer sig på olika sätt. Det kan vara skillnad i delavrinningsområdestillhörighet, jordart eller markanvändning. Eftersom det är önskvärt att begränsa antalet körningar i beräkningsmodellen SOIL vilket påverkas av hur många unika beräkningsenheter som finns har det gjorts vissa antaganden. Ett sådant antagande är att låta ett jordbruksblock innehålla endast en typ av jordart eftersom precisionen i jordartskartan är osäker. Detta kommer att bestämmas av vilken jordart som upptar störst area inom respektive jordbruksblock. Skriptet har tilldelats en funktion som summerar arean av de beräkningsenheter inom ett jordbruksblock med samma jordart och som sedan jämför denna area med andra summerade beräkningsenheters area för att därefter ersätta de andras jordarter med den som upptar störst area. Genom att bearbeta informationen på det här sättet reduceras antalet beräkningsenheter och antalet körningar i SOIL kommer därmed också att reduceras.

Ett jordbruksblock kan bestå av flera olika fält och varje fält definieras av en specifik gröda. I kartan med jordbruksblock identifieras inte fälten utan endast en gröda är definierad per jordbruksblock på grund av begränsningar i ArcInfo. Istället definieras de olika grödor som odlas på respektive jordbruksblock i fältdatabasen. Denna innehåller ett id-nummer för varje jordbruksblock, de motsvarande grödor som odlas på detta block samt

hur stor area varje gröda upptar på jordbruksblocket. En funktion i skriptet har till uppgift att läsa in denna tabell och spara den i en temporär lista (figur 12). En annan funktion jämför därefter id-numren från beräkningsenheten med id-numren från fältdatabasen och sparar fälten som är kopplade till respektive jordbruksblock. I fältdatabasen definieras inte det geografiska läge som respektive fält har utan det är endast möjligt att hämta information om hur stor andel fältet upptar på respektive jordbruksblock. Skriptet sparar respektive fält samt dettas delarea som en ny beräkningsenhet i den temporära listan men i övrigt med samma egenskaper som den ursprungliga beräkningsenheten. Den ursprungliga beräkningsenheten raderas därefter från den temporära listan.



Figur 12. Illustrering av hur skriptet hämtar information från parameterdatabasen och fältdatabasen

3.5 VAL AV BERÄKNINGSENHETER

Ett jordbruksblock delas in i flera beräkningsenheter om det är beläget på flera delavrinningsområden. Eftersom delavrinningsområdestillhörighet inte betraktas som en unik egenskap med avseende på indata till SOIL behöver endast en beräkningsenhet för varje gröda simuleras. På grund av detta har det byggts in en funktion som hanterar urvalsprocessen. Funktionen väljer ut en beräkningsenhet för varje gröda inom det blocket och ignorerar de beräkningsenheter som endast är unika med avseende på delavrinningsområdestillhörighet. På det sättet reduceras antalet simuleringar som ska genomföras. Dock behövs informationen om vilka delavrinningsområden de andra beräkningsenheterna är belägna på och hur stor area de upptar. Denna information lagras istället i instruktionsfiler.

3.6 INSTRUKTIONSFILER

För att ta hänsyn till arean av de beräkningsenheter som inte simuleras skapas under genereringen av indata till modellen även instruktionsfiler i textformat för att hantera dessa.

Den första filen som skapas heter DownStreamID och innehåller en lista med information om hur delavrinningsområdena rinner till varandra, det vill säga hur de är relaterade till varandra avrinningsmässigt. Denna används inte i detta skede men kommer att behövas vid det tillfälle det önskas ta hänsyn till det. RunList är en fil som listar de utvalda beräkningsenheterna från urvalsprocessen. Denna är till för att SOIL ska veta vilka indatafiler den ska basera dess simuleringar på. Den sista filen heter ID (bilaga 5, figur 27) och den hanterar alla skapade beräkningsenheter oberoende av urvalsprocessen och beskriver arean som dessa upptar samt inom vilket delavrinningsområde. Utifrån denna lista är det möjligt att härleda informationen om hur areorna ska summeras för att ta hänsyn till hela avrinningsområdet. För att beräkna varje fältarea ($A_{\text{fält}}$) har det skapats en funktion som hämtar information om delarean (A_{del}) och totalarean (A_{tot}) (som hämtats från fältdatabasen) samt polygonarean (A_{polygon}) (som beräknats under bearbetningen av kartorna).

$$(A_{\text{fält}}) = A_{\text{polygon}} * \left(\frac{A_{\text{del}}}{A_{\text{tot}}} \right) \quad [\text{m}^2] \quad (1)$$

4 SIMULERINGSPROCESSEN

SOIL är skapat för att ta emot indata för en hydrologisk beräkningsenhet i taget. Detta innebär att processen kring att ta emot indata har behövt modifierats. SOIL kan ta emot en inställningsfil åt gången, läsa in den, avgöra vilka hjälpfiler den ska läsa in tillsammans med inställningsfilen och därefter fullfölja simuleringen. Modifieringen innebär att när SOIL sedan är klar med simuleringen för den första beräkningsenheten ska den automatiskt läsa in nästa inställningsfil.

4.1 AGGREGERING

För varje genomförd simulering skapas en utdatafil. Denna fil innehåller modellerad information för varje dag för hela tidserien. Vilken typ av information som eftertraktas i utdata specificeras i inställningsfilen eller i automatiseringsprocessen. I detta fall då fokus valts att läggas på transporter av vatten är de specificerade utdataparametrarna:

- Total avrinning [mm/dygn]
- Dränering [mm/dygn]
- Perkolation [mm/dygn]
- Djupperkolation [mm/dygn]
- Ytavrinning [mm/dygn]

Efter körningarna i SOIL beräknas den aggregerade avrinningen genom instruktionsfilen ID som innehåller den nödvändiga informationen som behövs för att aggregera beräkningsenheterna felfritt. I instruktionsfilen definieras alla beräkningsenheter och inte bara de som har simulerats.

För varje beräkningsenhet som simulerats erhålls utdataparametrarna i en lista med specifika värden för varje dag under simuleringsperioden. Genom att i instruktionsfilen därefter läsa av vilka beräkningsenheter som tillhör respektive delavrinningsområde beräknas det totala bidraget från varje delavrinningsområde.

$$q_{del} = \frac{1}{A_{del}} * \sum_{n=1}^N (q_{ber} * A_{ber}) \quad [\text{mm/dygn}] \quad (2)$$

där q_{del} är delavrinningsområdets avrinning, q_{ber} beräkningsenhetens avrinning, A_{del} delavrinningsområdets area, A_{ber} beräkningsenhetens area och n beräkningsenhet.

Slutligen summeras alla delavrinningsområdets vattenföring för att erhålla avrinningsområdets totala avrinning.

$$q_{tot} = \frac{1}{A_{avr}} * \sum_{m=1}^M (q_{del} * A_{del}) \quad [\text{mm/dygn}] \quad (3)$$

där q_{tot} är hela avrinningsområdets avrinning, A_{avr} hela avrinningsområdets area och m delavrinningsområde.

4.2 UTVÄRDERING AV MODELLRESULTAT

4.2.1 Hantering av utdata

Efter simuleringar i SOIL av den bearbetade indatan erhålls resultaten i tabellform över vattenkvantiteten som transporteras ut ur respektive beräkningsenhet. Genom att summera vattenmängden för alla beräkningsenheter inom ett delavrinningsområde beräknas vilken vattenmängd som kommer att rinna till ett annat delavrinningsområde. Skriptet har en separat funktion som skapar en lista över hur de olika delavrinningsområdena rinner till varandra. Med hjälp av denna lista går det att generera beräkningar till de aktuella delavrinningsområdena.

4.2.2 Metod för kalibrering och parameterinställningar

När resultaten från körningarna har genererats och det är möjligt att jämföra dessa med uppmätta data bör modellen kalibreras genom ändring av parametervärden i inställningsfilen och filen med jordartsegenskaper. Eftersom gröddatafilerna inte genereras automatiskt kommer dessa inte att kalibreras inom detta projekt.

4.2.3 Nash-Sutcliffe's effektivitetskoefficient

Nash-Sutcliffes effektivitetskoefficient (NSE) används för att utvärdera hydrologiska modellers prestation. Termen i täljaren visar variansen av resultaten från simuleringarna, det vill säga ett mått på hur mycket de modellerade värdena skiljer sig från de uppmätta värdena. Termen i nämnaren visar den residuala variansen, det vill säga ett mått på hur mycket medelvärdet av de uppmätta värdena skiljer sig från de modellerade värdena. En effektivitetskoefficient med ett större värde än noll påvisar att de modellerade värdena ger en bättre uppskattning av avrinningen än medelvärdet av de uppmätta värdena. En effektivitetskoefficient med ett mindre värde än noll påvisar att medelvärdet av de uppmätta värdena ger en bättre uppskattning av avrinningen än de modellerade värdena (Moriasi et al., 2007). Det är dock inte möjligt att upptäcka flödesdynamik med denna metod.

$$NSE = 1 - \frac{\sum_{t=1}^T (Q_o^t - Q_m^t)^2}{\sum_{t=1}^T (Q_o^t - \bar{Q}_o)^2} \quad [-] \quad (4)$$

där Q_o är uppmätt avrinning, Q_m modellerad avrinning, $\bar{Q}_o$ medelvärdet av uppmätta avrinningen och t dygn.

4.2.4 Volymfel

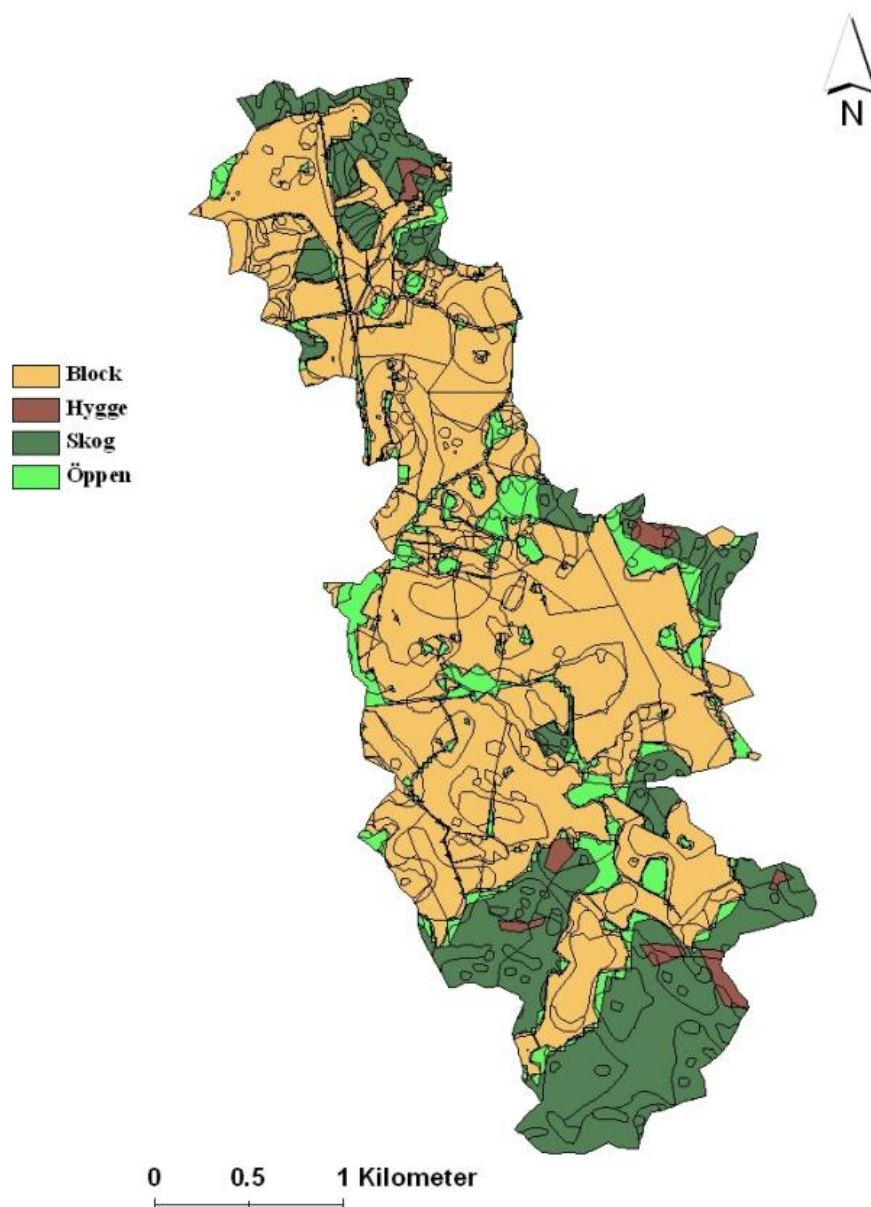
För att inte endast beräkna storleken på hur mycket de modellerade värdena skiljer sig från medelvärdet av de uppmätta värdena beräknas även volymfelet (V_E) för de modellerade värdena. Volymfelet ger en uppfattning om hur mängden avrinning från de modellerade värdena skiljer sig från mängden avrinning från de uppmätta värdena.

$$V_E = 100 * \frac{\sum_{t=1}^T (Q_o^t - Q_m^t)}{\sum_{t=1}^T Q_o^t} \quad [\%] \quad (5)$$

5 RESULTAT

5.1 GENERERADE BERÄKNINGSENHETER

Innan urvalet av vilka beräkningseenheter som ska simuleras genomförts generades 1406 beräkningseenheter i avrinningsområdet (figur 13).



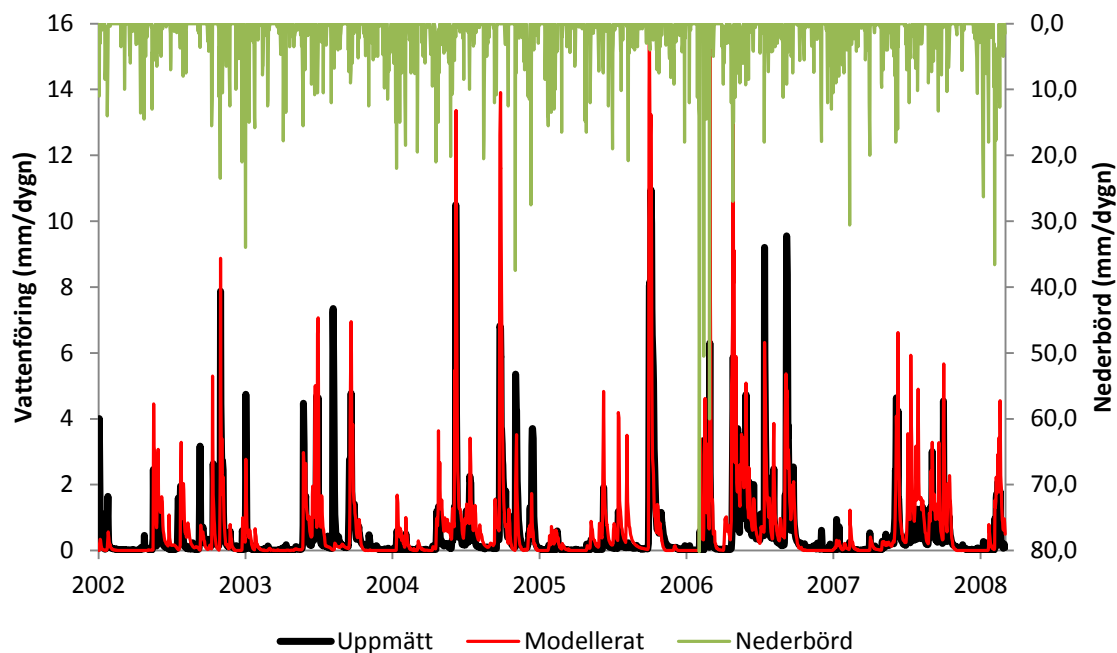
Figur 13. Visualisering av markanvändningen i den redigerade kartan

Efter att låtit skriptet läsa in den redigerade kartan valdes 647 beräkningsenheter ut för simulering och därmed skapades lika många inställningsfiler och filer med jordartsegenskaper. Medelareal för beräkningsenheter i studieområdet beräknades till 0,12 km² (12 ha).

5.2 AVRINNINGEN

Efter att ha kört SOIL ett antal gånger med varierad indata för att uppnå en så jämn serie av den totala vattenföringen som möjligt och undvika de största avvikelserna har följande resultat genererats (figur 14).

5.2.1 Jämförelse med uppmätta värden



Figur 14. Det modellerade och uppmätta flödet för hela tidsserien

$$NSE = -0,0072$$

$$V_E = 86,3 \%$$

I tabell 1 visas antalet dagar som överskattar respektive underskattar avrinningen samt hur stort volymfelet var för dessa dagar.

Tabell 1. Antalet dagar under simuleringsperioden som överskattar respektive underskattar avrinningen

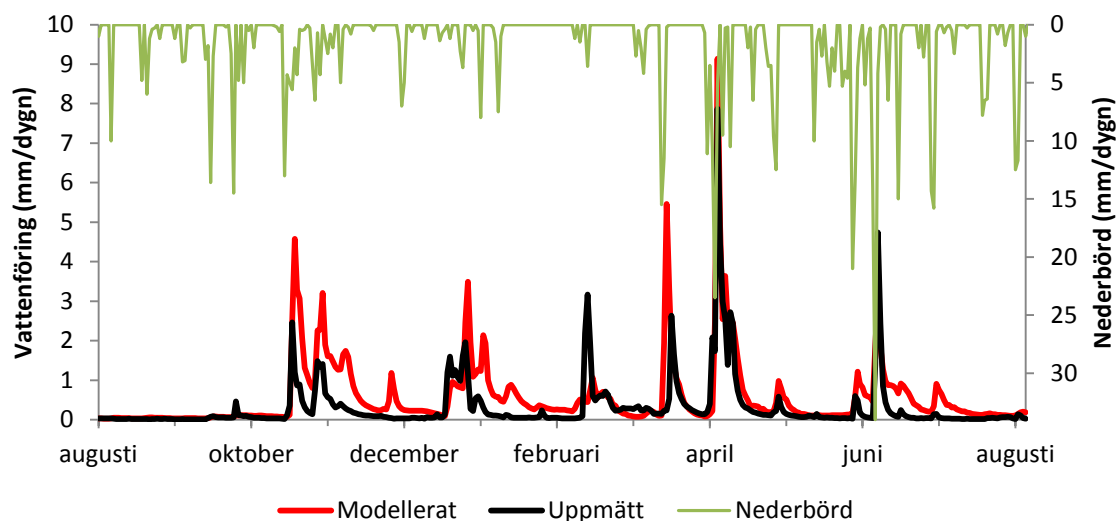
	Överskattning	Underskattning
Antal dagar (-)	1841	412
V_E (%)	160,4	37,6

För att bättre se hur modellen presterade beräknades resultaten även årsvis. Eftersom den sista dagen i simuleringsperioden var den 31 augusti har detta datum valts för att separera åren från varandra. I tabell 2 presenteras effektivitetskoefficienten för varje år. För perioden 2002-2003 erhöles den bästa effektivitetskoefficienten och för perioden 2007-2008 den sämsta.

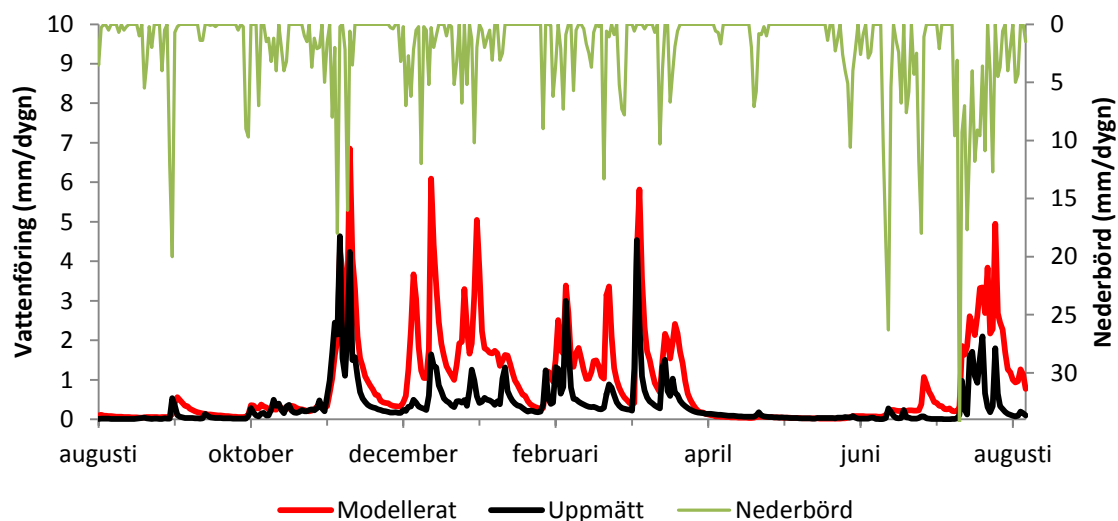
Tabell 2. Effektivitetskoefficienten beräknad per år under simuleringsperioden

Period	NSE
2002-2003	0,22
2003-2004	-0,06
2004-2005	0,06
2005-2006	0,08
2006-2007	0,09
2007-2008	-1,43

I figur 15 och figur 16 har avrinningen plottats för det bästa respektive sämsta året.



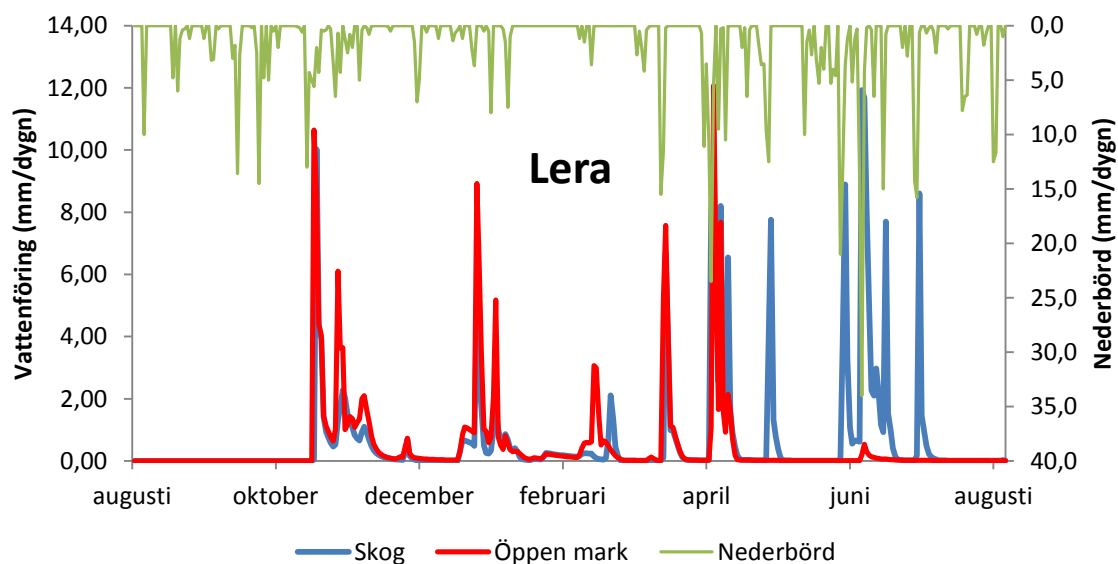
Figur 15. Den modellerade och uppmätta vattenföringen för året med bäst effektivitetskoefficient (period 2002 - 2003)



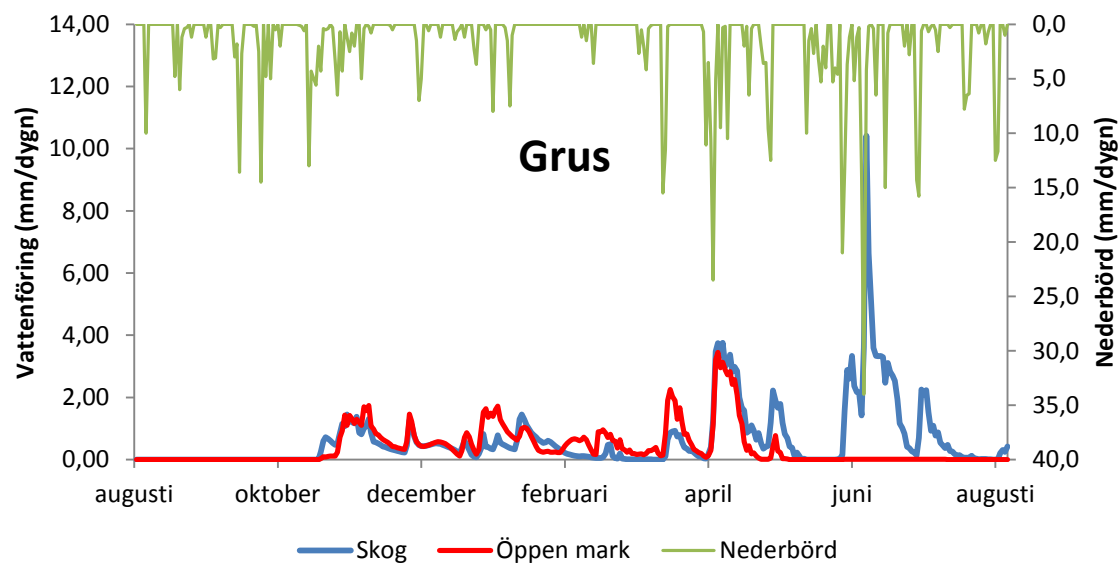
Figur 16. Den modellerade och uppmätta vattenföringen för året med sämst effektivitetskoefficient (period 2007 - 2008)

5.2.2 Jämförelse mellan beräkningsenheter

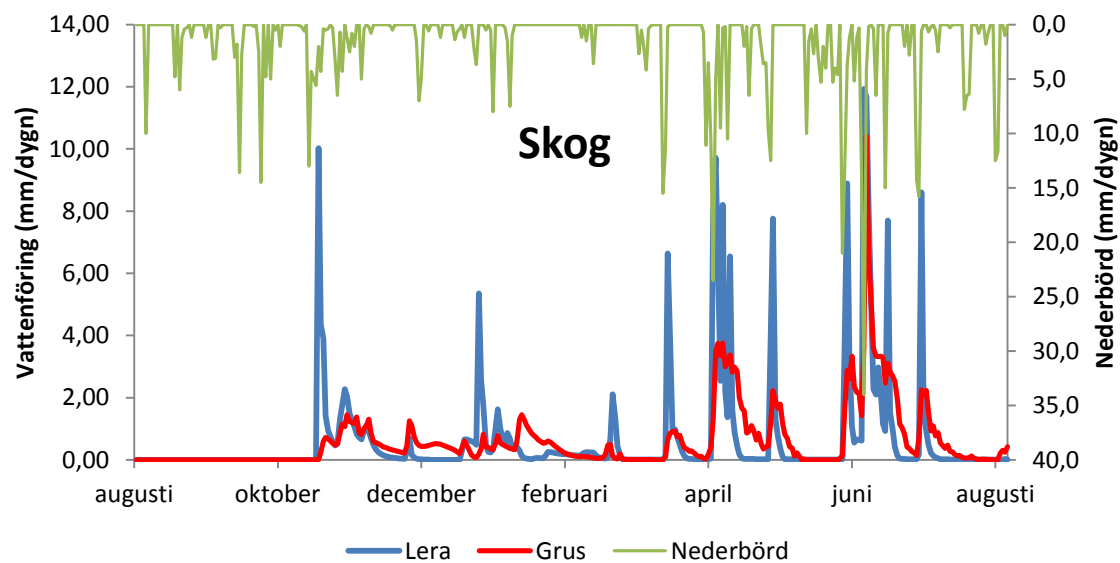
Här illustreras avrinningen från några av alla de beräkningsenheter som har simulerats i studien (figur 17-20).



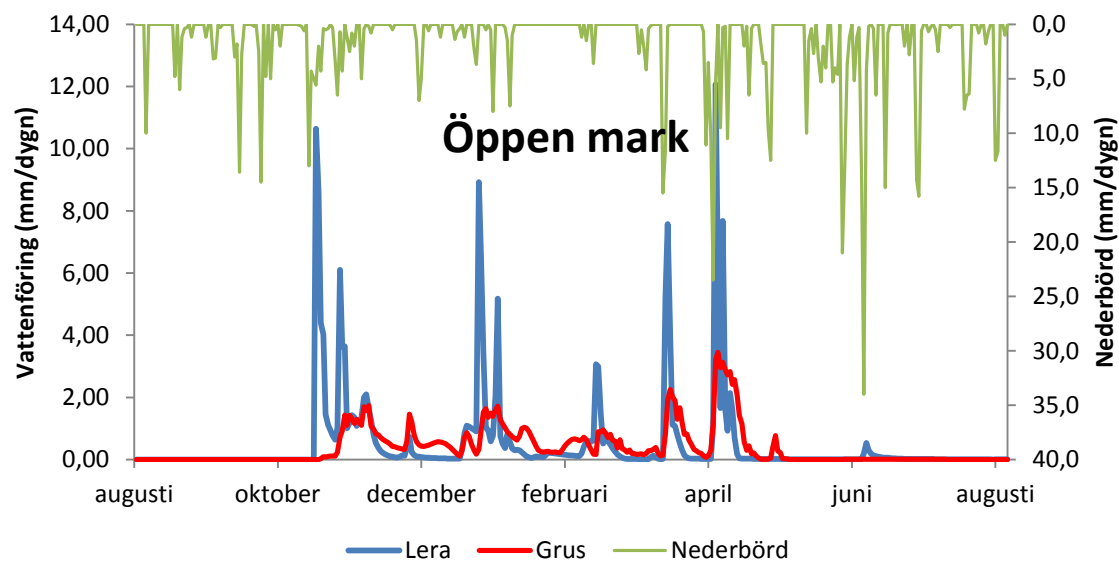
Figur 17. Jämförelse mellan markanvändningarna skog och öppen mark på lera



Figur 18. Jämförelse mellan markanvändningarna skog och öppen mark på grus



Figur 19. Jämförelse mellan jordarterna lera och grus i skog



Figur 20. Jämförelse mellan jordarterna lera och grus på öppen mark

I tabell 3 åskådliggörs mängden avrinning samt återstoden av vattnet från de plottade beräkningsenheterna.

Tabell 3. Total nederbörd (P) och avrinning (R) för de olika beräkningsenheterna för hela simuleringsperioden. Återstoden av vattnet kan vara lagring eller avdunstning.

Jordart	Markanvändning	P (mm/år)	R (mm/år)	P - R (mm/år)
Lera	Skog	666	329	337
Lera	Öppen Mark	666	302	364
Grus	Skog	666	324	342
Grus	Öppen Mark	666	238	428

6 DISKUSSION

I denna studie har avrinningen beräknats genom användning av parametervärden från litteraturdata. Kalibreringen har varit mycket begränsad och endast utförts på en parameter, med undantag för ett fåtal parametrar som beskriver initialförhållanden för simuleringen. Parametern som kalibrerades beskriver ytavrinningen från beräkningsenheterna.

Efter att ha observerat resultaten från simuleringarna och jämförelser mellan uppmätta och modellerade värden går det först och främst att dra slutsatsen att modellen på ett tillfredställande sätt kan beskriva flödesdynamik. Som det framgått har ingen detaljerad kalibrering av modellen hunnit genomföras och dessa första resultat är mycket lovande inför framtida utvecklingsinsatser. Om en automatiserad kalibrering av modellen utvecklas för att kunna beräkna parametrarnas känslighet och osäkerhet borde resultaten kunna förbättras en hel del.

Eftersom att det i denna studie har antagits att avrinningen från olika beräkningsenheter omedelbart rinner till utloppet och att resultaten tyder på en viss likhet mellan modellerad och uppmätt avrinning, skulle det kunna tyda på att processer som sker mellan hydrologiska beräkningsenheter inte är fullt så betydelsefulla i denna skala.

Från beräkningarna av volymfelet går det även att konstatera att de modellerade flödena oftast är högre än de uppmätta vilket det kan finnas en rad förklaringar till. För det första måste alla eventuella felaktigheter i programmet och orimliga resultat utvärderas och åtgärdas. Därefter bör en detaljerad kalibrering av modellen genomföras för att hitta optimala parametervärden. Slutligen kan det vara lämpligt att undersöka den konceptuella modellen närmare om stora avvikelser mellan modellerade resultat och mätningarna kvarstår.

Resultaten från jämförelsen av beräkningsenheter har inte ansetts vara betydelsefulla i detta arbete eftersom en hel del kalibrering av modellen återstår. Istället presenteras dessa för att illustrera hur olika egenskaper hos beräkningsenheter påverkar vattenföringen. Detta kan vara intressant att bedöma efter parameterkalibrering för att öka förståelsen kring avrinning. Genom att beräkna återstoden av mängden vatten (tabell 3) i beräkningsenheterna är det möjligt att bedöma ungefär hur stor del avdunstning som sker från varje kombination av markanvändning och jordart. De värdena kan ge en uppfattning om hur sanningsenliga resultaten är och vara till hjälp inför kalibreringen av modellen.

Angående genereringen av indata till SOIL bör det noteras att ett flertal parametrar inte är beroende av beräkningsenheternas egenskaper utan är i detta skede oföränderliga oberoende av enhetens egenskaper. Detta beror helt enkelt på att mängden parametrar som behövt definierats är för omfattande för denna studie. De parametrar som behandlats och alltså varierar med indata har antagits ha störst inflytande på körningarna med avseende på transport av vatten. Det finns dock parametrar som helt enkelt definierats efter

standardvärden som har tillhandahållits tillsammans med modellen. I den fortsatta utvecklingen av modellen ingår dock känslighets- och osäkerhetsanalys för att utvärdera parametrarnas betydelse för modellresultat.

Faktorer som kan ha orsakat sämre överensstämmanden i resultaten kan vara att SOIL inte tar någon hänsyn till lutningen i området eftersom SOIL endast arbetar i en dimension. Lutningen på delavrinningsområden påverkar antagligen flödet till relativt stor del och kan behöva implementeras i ett senare skede. En annan faktor kan vara att utflödet ur de hydrologiska beräkningsenheterna antagits rinna till vattendragen direkt utan att detta behöver vara fallet. Ytvattnet som rinner från en beräkningsenhet kan mycket väl infiltrera ett annat område längre ner i avrinningsområdet innan det rinner till utloppspunkten. Ogynnsam placering av mätstationen för klimatdata skulle också kunna påverka resultaten till viss del. Lokala väderförhållanden kan då leda till skillnader i klimatet mellan mätplatsen och studieområdet.

7 SLUTSATS OCH VIDAREUTVECKLING

Det finns fortfarande mycket arbete som behöver utföras i utvecklingen av SWE-model och metoder som använts i detta arbete kan komma att förändras för att till exempel förbättra prestandan, men denna studie har givit en överblick över vilka problem som eventuellt kan uppkomma med ansatsen till en ny modell. Detta inkluderar vilka olika moment som behöver genomföras och hur de skulle kunna modifieras.

Denna prototyp har gett möjligheten till att relatera varje enskild beräkningsenhets bidrag till den totala avrinningen i avrinningsområdet. Från resultaten är det möjligt att se en viss likhet i flödesdynamik mellan modellerad och uppmätt vattenföring. Effektivitetskoefficienten för hela simuleringsperioden har beräknats till att motsvara medelvärdet för de uppmätta värdena. Med en automatiserad kalibreringsprocess borde modellen ha möjlighet till att prestera bättre. Volymfelet ger en indikation på överskattning från modellens sida.

För närvarande är det bara simuleringsfiler belägna på åkermark som till antalet stämmer. Simuleringsfiler med annan markanvändning har hittills genererats för varje polygon på kartan utanför jordbruksblock. Om polygonerna är geografiskt åtskilda trots att de besitter samma egenskaper har det för dessa polygoner skapats separata simuleringsfiler. För att spara processorkraft kan dessa polygoner slås samman. I ett senare skede i utvecklingen av modellen, då en automatisk parameterkalibrering kommer att ske, kommer man att vilja begränsa simuleringstiden så mycket som möjligt och här kommer denna bearbetning att bli absolut nödvändig. Ytterligare metoder för att minska simuleringstiden är att skapa en funktion med areaselektion som filtrerar ut de till ytan minsta beräkningsenheterna, vilkas bidrag kan antas vara försumbara, för att sedan addera denna area till simuleringsfiler med samma egenskaper. För att minska antalet simuleringsfiler går det även att tilldela alla beräkningsenheter utanför åkermarken, men inom ett särskilt delavrinningsområde, en gemensam jordart. För dessa beräkningsenheter skulle det då innebära att det endast är markanvändningen som kommer att variera och därmed generera nya simuleringsfiler. Denna problematik beror till stor del av kvaliteten och precisionen i de befintliga jordartskartorna. Här hittas oftast en svag länk där jordartsindelningen är mycket osäker.

I framtiden kan det finnas ett intresse i att omvandla de binära filer som för närvarande används som indata till textfiler. Att lagra innehållet på det sättet skulle kunna möjliggöra en förenklad hantering av informationen i dem. Binära filer upptar dock inte lika mycket utrymme som textfiler eftersom innehållet i dessa är mer komprimerat och det finns frågetecken kring hur hastigheten på inläsningen förändras.

Det är högst intressant att automatisera återkopplingen till varje enskild beräkningsenhet. I ett senare skede bör det vara möjligt att genom återkoppling lokalisera högriskområden med avseende på näringsämnen.

För att erhålla en bättre överensstämmelse i det initiala skedet av simuleringen är det möjligt att förlänga tidsserien genom att addera värden i början av indatafilerna, antagligen med data från år 1, för att i utdata sedan ignorera dessa värden med förhoppningen att övriga värden för den ursprungliga tidsserien ställt in sig.

Det kan vara intressant att jämföra resultaten från simuleringarna i detta arbete med simuleringar utförda i modeller som simulerar i samma skala men som inte är anpassade för svenska förhållanden för samma studieområde för att få en uppfattning om hur bra denna modell presterar. Detta med förhoppningen att utnyttjandet av att inkorporera svenska parametervärden resulterar i en ännu bättre prediktion av vattenföring och senare även vattenkvalitet.

REFERENSER

Anlauf, R. (2008). Watershed Modeling with ArcSWAT: Calibration and Validation for the Prediction of Flow. *International Meeting on Soil Fertility Land Management and Agroclimatology*, (ss. 279-288).

esri. (2010). *ArcInfo Overview*. Hämtat från esri:
<http://www.esri.com/software/arcgis/arcinfo/index.html> den 3 November 2010

Grip, H., & Rodhe, A. (2003). *Vattnets väg från regn till bäck*. Uppsala: Hallgren & Fallgren studieförlag AB.

Jansson, P.-E. (1991). Description of the SOIL model. Uppsala, Sverige: Institutionen för Markvetenskap.

Jansson, P.-E. (1993). SOIL model User's Manual. Uppsala, Sverige: Sveriges Lantbruksuniversitet.

Johnes, P. J., & Heathwaite, A. L. (1997). Modelling the impact of land use change on water. *Hydrological Processes*, 11, 269-286.

Jordbruksverket. (Mars 2009). Instruktion för fältinventering av brukarblock. *Instruktion för fältinventering av brukarblock version 4*. Sverige: Jordbruksverket.

Jordbruksverket. (den 10 September 2009). *Jordbruket och övergödningen*. Hämtat från Jordbruksverket:
<http://www.jordbruksverket.se/amnesomraden/miljoocklimat/ingenovergodning/jordbruke-tochovergodningen.4.4b00b7db11efe58e66b80001608.html> den 28 Oktober 2010

MathWorld. (den 19 November 2010). *Wolfram MathWorld Polygon*. Hämtat från Wolfram MathWorld: <http://mathworld.wolfram.com/Polygon.html> den 20 November 2010

Moriasi, D. N., Arnold, J. G., Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations. *Transactions of the ASABE*, 50(3): 885-900.

Python. (2010). *About Python*. Hämtat från Python Programming Language – Official Website: <http://www.python.org/about/> den 3 November 2010

Sharpley, A. N., Smith, S. J., & Naney, J. W. (1987). Environmental Impact of Agricultural Nitrogen and Phosphorus Use. *Agric. Food Chem.*(35), 812-817.

Sheshukov, A., Daggupati, P., Lee, M.-C., & Douglas-Mankin, K. (den 7 Augusti 2009). ArcMap Tool for Pre-processing SSURGO Soil Database. *Proceedings of the 5th International SWAT Conference*.

SLU vattenNAV. (2010). *SLU vattenNAV*. Hämtat från Sveriges Lantbruksuniversitet:
<http://vattennav.slu.se/> den 17 November 2010

BILAGA 1 - SKRIPT

```
## Skript för generering av indatafiler till SOIL
##
## av Stefan Hellgren
```

```
SWATSOIL = {'Sand':'1',
            'Loam':'4',
            'Clay loam':'7',
            'Clay':'10',
            'Morän':'11',
            'Organisk':'12',
            'Berg':'13',
            'Silt':'14',
            'Grus':'15',
            }
```

```
CROPSWAT = {'Höstkorn':'1',
            'Vårkorn':'2',
            'Spring barley':'2',
            'Havre':'3',
            'Oats':'3',
            'Höstvete':'4',
            'Winter Wheat':'4',
            'AGRC':'4',
            'Vårvete':'5',
            'Corn':'9',
            'Vårraps':'21',
            'Spring rape':'21',
            'Spring rape 2':'21',
            'Ley':'50',
            'Ley 2':'50',
            'Pasture':'50',
            }
```

```
from xlrd import open_workbook
import pyodbc, arcgisscripting, copy
```

```
fileName = 'C:\~\tyomr__korr_Identity_I1.shp'
destDir = 'C:\~\Utdata/'
DBfile = 'C:\~\parameters 32.mdb'
cropDB = 'C:\~\_utdrag_100525(2).xls'
```

```
def readGis():
    gp=arcgisscripting.create(9.3)

    fcdescribe=gp.Describe(fileName)
    flds=fcdescribe.Fields

    header=[]
    for fld in flds:
        value=fld.Name
        if type(value) == unicode:
```

```
        value = value.encode('8859')
        header.append(value)
```

```
cursor=gp.SearchCursor(fileName)
row=cursor.Next()
```

```
gisList = []
while row:
    line=[]
    for fld in flds:
        value=row.GetValue(fld.Name)
        if type(value) == unicode:
            value = value.encode('8859')
        line.append(value)
    gisList.append(line)
    del line
    row=cursor.Next()
```

```
del cursor
return header, gisList
```

```
def setCropId(header, gisList):
```

```
    for y in gisList:
        if y[header.index('CROPSWAT')] in
            CROPSWAT:
                y[header.index('CROPSWAT')] =
                    CROPSWAT[y[header.index('CROPSW
                    AT')]]
    return gisList
```

```
def setSoilId(header, gisList):
```

```
    for y in gisList:
        if y[header.index('SWATSOIL')] in
            SWATSOIL:
                y[header.index('SWATSOIL')] =
                    SWATSOIL[y[header.index('SWATSOI
                    L')]]
    return gisList
```

```
def setLandUseId(header, gisList):
```

```
    for y in gisList:
        if y[header.index('HEL_ID')] == 0:
            if y[header.index('MARKANVNAM')] ==
                'Skog':
                y[header.index('CROPSWAT')] = 80
            if y[header.index('MARKANVNAM')] ==
                'Öppen':
                y[header.index('CROPSWAT')] = 50
```



```

        if y[header.index('MARKANVNAM')] ==
            'Hygge':
                y[header.index('CROPSWAT')] = 81
        if y[header.index('MARKANVNAM')] ==
            'Block':
                y[header.index('CROPSWAT')] = 50
        if y[header.index('MARKANVNAM')] == '
            ':
                y[header.index('CROPSWAT')] = 50
    return gisList

def setSoilType(header, soilIDList):

    ## ----- Hittar polygoner för respektive
        jordbruksblock -----
        -----
        -----

    sameSoilList = []
    blockIdList = []
    for plygn in soilIDList:
        ## Går igenom en polygon åt gången och
            sparar jordbruksblock i en lista 'jbbList'
        blockId = plygn[header.index('HEL_ID')]
        if blockId != 0:
            if blockId not in blockIdList:
                blockIdList.append(blockId)

            jbbList = []
            for compareplygn in soilIDList:
                y =
                compareplygn[header.index('HEL_ID')]
                if y == blockId:
                    jbbList.append(compareplygn)

    ## ----- Hittar polygoner med samma jordart
        för det aktuella jordbruksblocket -----
        -----
        -----

    largestArea = 0
    soilIdList = []
    for jbbPlygn in jbbList:
        ## Går igenom lista med polygoner inom
            samma jordbruksblock
        soil =
        jbbPlygn[header.index('SWATSOIL')]
        if soil not in soilIdList:
            soilIdList.append(soil)

        soilList = []
        for comparejbbPlygn in jbbList:

            compareSoil =
            comparejbbPlygn[header.index('SWATS
                OIL')]
            if compareSoil == soil:

                soilList.append(comparejbbPlygn)
                ## Lista med polygoner med samma
                jordart

    ## ----- Summerar areor och beräknar
        vilken jordart som upptar störst area inom
        det aktuella jordbruksblocket -----
        -----

        area = 0
        for sameSoil in soilList:
            ## Adderar arean av de polygoner med
                samma jordartstyp
            area = area +
            sameSoil[header.index('Area_tot29')]

            if area > largestArea:
                mostPresentSoil =
                sameSoil[header.index('SWATSOIL')]
                largestArea = copy.copy(area)

        for a in jbbList:
            a[header.index('SWATSOIL')] =
            mostPresentSoil    ## Tilldelar alla
                polygoner inom samma jordbruksblock
                den jordart som förekommer mest inom
                detta
            sameSoilList.append(a)

    else:
        sameSoilList.append(plygn)

    print '\nNumber of fields before crop
        extraction:',len(sameSoilList)
    return sameSoilList

def readXI():
    wb = open_workbook(cropDB,'rb')

    cropHeader = []
    cropInfo = []
    for s in wb.sheets():
        if s.name == 'Final':
            for row in range(s.nrows):
                if row == 0:
                    for col in range(s.ncols):
                        x = s.cell(row,col).value
                        if type(x) == unicode:
                            x = x.encode('8859')

```

```

        cropHeader.append(x)
    if row != 0:
        line = []
        for col in range(s.ncols):
            x = s.cell(row,col).value
            if type(x) == unicode:
                x = x.encode('8859')
            line.append(x)
        cropInfo.append(line)
##     else:
##         print s.name, 'is not a valued sheet\n'

    return cropHeader, cropInfo

def getCrops(nHeader, sameSoilList, cropHeader,
             cropInfo):
    print 'Preparing hydrological response units for
    available crops..\n'
    totalHRU = []
    unvalid = []
    nHeader.append('delArea')
    nHeader.append('totArea')
    for plygn in sameSoilList:
        plygn.append('0')
        plygn.append('0')
    for plygn in sameSoilList:
        x = plygn[nHeader.index('HEL_ID')]
        y = plygn[nHeader.index('ID')]
        a = [x, y]
        if x == 0:
            totalHRU.append(plygn)
        else:
            if a not in unvalid:
                unvalid.append(a[:])
                delArea = []
                totArea = []
                tempCrops = []
                for line in cropInfo:
                    if line[cropHeader.index('HEL_ID')]
                    == x:

                tempCrops.append(line[cropHeader.index
                ('GrödKod')])

                delArea.append(line[cropHeader.index('A
                rea_new')])

                totArea.append(line[cropHeader.index('T
                ot_Area_block_new')])

            copyCropList = []
            for crop in tempCrops:
                copyCropList.append(plygn[:])
            for count in range(0, len(copyCropList)):

```

```

                copyCropList[count][nHeader.index('CR
                OPSWAT')] = tempCrops[count]

                copyCropList[count][nHeader.index('del
                Area')] = delArea[count]

                copyCropList[count][nHeader.index('totA
                rea')] = totArea[count]
            for newPlygn in copyCropList:
                totalHRU.append(newPlygn)

    print 'Number of fields after crop extraction:
    ',len(totalHRU)
    return nHeader, totalHRU

def setIniMark(header, sameSoilList):
    v = 0
    unvalid = []
    header.append('iniName')

    for plygn in sameSoilList:
        x = plygn[header.index('HEL_ID')]
        z = plygn[header.index('CROPSWAT')]
        h = [x, z]
        if x != 0:
            if h not in unvalid:
                unvalid.append(h)
                v = v + 1
                b = '%03d' % v
                for plygn2 in sameSoilList:
                    y = plygn2[header.index('HEL_ID')]
                    p =
                    plygn2[header.index('CROPSWAT')]
                    g = [y, p]
                    if h == g:
                        plygn2.append(b + '.ini')
                if x == 0:
                    v = v + 1
                    b = '%03d' % v
                    plygn.append(b + '.ini')

    return header, sameSoilList

def getJbbArea(delArea, totArea, newJbbArea):
    andel = delArea/totArea
    c = andel*newJbbArea
    return c

def calcTotAreaNew(nHeader, namedList, x):
    nHeader.append('totAreaNew')
    for i in range(1, x+1): ## Går igenom alla
        delavrinningsområden
        totAreaNew = 0

```

```

for j in range(0, len(namedList)):
    if namedList[j][nHeader.index('ID')] == i:
        if
            namedList[j][nHeader.index('HEL_ID')]
            == 0:
                c =
                    namedList[j][nHeader.index('Area_tot29')]
                ]
            else:
                delArea =
                    namedList[j][nHeader.index('delArea')]
                totArea =
                    namedList[j][nHeader.index('totArea')]
                newJbbArea =
                    namedList[j][nHeader.index('AreaSplitJ')]
                ]
                c = getJbbArea(delArea, totArea,
                    newJbbArea)
                totAreaNew = totAreaNew + c
            else:
                pass
        for k in range(0, len(namedList)):
            if namedList[k][nHeader.index('ID')] == i:
                namedList[k].append(totAreaNew)

return nHeader, namedList

def calcFullArea(nHeader, namedList, x):
    nHeader.append('fullArea')
    fullArea = 0
    for j in range(0, len(namedList)):
        if namedList[j][nHeader.index('HEL_ID')] ==
            0:
                c =
                    namedList[j][nHeader.index('Area_tot29')]
                ]
            else:
                delArea =
                    namedList[j][nHeader.index('delArea')]
                totArea =
                    namedList[j][nHeader.index('totArea')]
                newJbbArea =
                    namedList[j][nHeader.index('AreaSplitJ')]
                ]
                c = getJbbArea(delArea, totArea,
                    newJbbArea)
                fullArea = fullArea + c
    for plygn in namedList:
        plygn.append(fullArea)

return nHeader, namedList

def createIdList(nHeader, namedList):

```

```

x = 0
for count in namedList: ## Beräknar antalet
    Delavrinningsområden i SameSoilList
    och placerar värdet i 'x'.
    y = count[nHeader.index('ID')]
    if y > x:
        x = y

nHeader, namedList =
    calcTotAreaNew(nHeader, namedList, x)
nHeader, namedList = calcFullArea(nHeader,
    namedList, x)

listFile = open(destDir+'ID'+'.txt', 'w')
skrivRad = 'DelID ' + 'IniName ' + 'BlockID ' +
    'SoilID ' + 'CropID ' + 'delArea ' +
    'OrgArea ' + 'TotArea ' + 'FullArea\n'
listFile.write(skrivRad)
unvalid = []

for i in range(1, x+1): ## Går igenom alla
    delavrinningsområden
    for j in range(0, len(namedList)):
        if namedList[j][nHeader.index('ID')] == i:
            a, b = i,
                namedList[j][nHeader.index('iniName')]
            d =
                namedList[j][nHeader.index('AREA_M')]
            e, f, g =
                int(namedList[j][nHeader.index('HEL_ID
                ')]),
                int(namedList[j][nHeader.index('SWATS
                OIL')]),
                namedList[j][nHeader.index('CROPSWA
                T')]
            h =
                namedList[j][nHeader.index('totAreaNew
                ')]
            m =
                namedList[j][nHeader.index('fullArea')]

            a = '%01d' % a
            e = '%03d' % e
            f = '%03d' % f

            if
                namedList[j][nHeader.index('HEL_ID')]
                == 0:

                g = '%03d' % g
                c =
                    namedList[j][nHeader.index('Area_tot29')]
                ]

```

```

        skrivRad = a +' '+ b + e.rjust(12) +
f.rjust(4) + g.rjust(4) +' '+ str(c).rjust(16)
+ str(d).rjust(11) + str(h).rjust(16) +
str(m).rjust(15)+'\n'
        listFile.write(skrivRad)

    else:

        delArea =
namedList[j][nHeader.index('delArea')]
        totArea =
namedList[j][nHeader.index('totArea')]
        newJbbArea =
namedList[j][nHeader.index('AreaSplitJ')]
    ]
        g = '%03d' % g
        c = getJbbArea(delArea, totArea,
newJbbArea)
        skrivRad = a +' '+ b + e.rjust(12) +
f.rjust(4) + g.rjust(4) +' '+ str(c).rjust(16)
+ str(d).rjust(11) + str(h).rjust(16) +
str(m).rjust(15)+'\n'
        listFile.write(skrivRad)

listFile.close()

## ----- Parametrar - Start -----
-----
-----

def setParam(nHeader, plygn, defaultP, table,
        headLine):
    cursor.execute(table)
    for row in cursor:
        if plygn[nHeader.index(headLine)] != ' ':
            if int(plygn[nHeader.index(headLine)]) ==
row.param:
                p = row.paramV
                break
            else:
                p = defaultP
        else:
            p = defaultP
    e, f = str(p).split('.')
    if len(e) == 1:
        if f == '0':
            p = '%.9f' % p
        else:
            p = '%.10f' % p
    if len(e) == 2:
        p = '%.8f' % p
    if len(e) == 3:
        p = '%.7f' % p
    return p

```

```

def setParam2(nHeader, plygn, defaultP, table,
        headLine):
    cursor.execute(table)
    for row in cursor:
        if plygn[nHeader.index(headLine)] != ' ':
            if plygn[nHeader.index(headLine)] ==
row.param:
                p = row.paramV
                break
            else:
                p = defaultP
        else:
            p = defaultP
    e, f = str(p).split('.')
    if len(e) == 1:
        if f == '0':
            p = '%.9f' % p
        else:
            p = '%.10f' % p
    if len(e) == 2:
        p = '%.8f' % p
    if len(e) == 3:
        p = '%.7f' % p
    return p

def getDB(nHeader, plygn):
    conn = pyodbc.connect('DRIVER={Microsoft
Access Driver (*.mdb,
*.accdB)};DBQ='+DBfile, autocommit =
True)
    global cursor
    cursor = conn.cursor()

    ## param12 fcond
    defaultP = 20.00000000
    table = 'SELECT [type of soil] AS param,
[fcond] AS paramV FROM [soil (SOIL)]'
    headLine = 'SWATSOIL'
    p012 = setParam2(nHeader, plygn, defaultP,
table, headLine)

    ## param71 LAIV1
    defaultP = 0.1000000015
    table = 'SELECT [GrödKod] AS param, [laiv1]
AS paramV FROM [newCrop]'
    headLine = 'CROPSWAT'
    p071 = setParam(nHeader, plygn, defaultP, table,
headLine)

    ## param72 LAIV2
    defaultP = 1.000000000
    table = 'SELECT [GrödKod] AS param, [laiv2]
AS paramV FROM [newCrop]'

```

```
headLine = 'CROPSWAT'
p072 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param73 LAIV3
defaultP = 5.000000000
table = 'SELECT [GrödKod] AS param, [laiv3]
    AS paramV FROM [newCrop]'
headLine = 'CROPSWAT'
p073 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param74 LAIV4
defaultP = 1.000000000
table = 'SELECT [GrödKod] AS param, [laiv4]
    AS paramV FROM [newCrop]'
headLine = 'CROPSWAT'
p074 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param75 LAIV5
defaultP = 0.1000000015
table = 'SELECT [GrödKod] AS param, [laiv5]
    AS paramV FROM [newCrop]'
headLine = 'CROPSWAT'
p075 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param76 DISPLV1
defaultP = 0.1000000015
table = 'SELECT [GrödKod] AS param,
    [displv1] AS paramV FROM [newCrop]'
headLine = 'CROPSWAT'
p076 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param77 DISPLV2
defaultP = 0.1000000015
table = 'SELECT [GrödKod] AS param,
    [displv2] AS paramV FROM [newCrop]'
headLine = 'CROPSWAT'
p077 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param78 DISPLV3
defaultP = 0.1000000015
table = 'SELECT [GrödKod] AS param,
    [displv3] AS paramV FROM [newCrop]'
headLine = 'CROPSWAT'
p078 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param79 DISPLV4
defaultP = 0.1000000015
```

```
table = 'SELECT [GrödKod] AS param,
    [displv4] AS paramV FROM [newCrop]'
headLine = 'CROPSWAT'
p079 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param80 DISPLV5
defaultP = 0.1000000015
table = 'SELECT [GrödKod] AS param,
    [displv5] AS paramV FROM [newCrop]'
headLine = 'CROPSWAT'
p080 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param81 DAYNUM1
defaultP = 50.000000000
table = 'SELECT [GrödKod] AS param,
    [daynum1] AS paramV FROM
    [newCrop]'
headLine = 'CROPSWAT'
p081 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param82 DAYNUM2
defaultP = 0.000000000
table = 'SELECT [GrödKod] AS param,
    [daynum2] AS paramV FROM
    [newCrop]'
headLine = 'CROPSWAT'
p082 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param83 DAYNUM3
defaultP = 0.000000000
table = 'SELECT [GrödKod] AS param,
    [daynum3] AS paramV FROM
    [newCrop]'
headLine = 'CROPSWAT'
p083 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param84 DAYNUM4
defaultP = 0.000000000
table = 'SELECT [GrödKod] AS param,
    [daynum4] AS paramV FROM
    [newCrop]'
headLine = 'CROPSWAT'
p084 = setParam(nHeader, plygn, defaultP, table,
    headLine)
```

```
## param85 DAYNUM5
defaultP = 0.000000000
```

```

table = 'SELECT [GrödKod] AS param,
            [daynum5] AS paramV FROM
            [newCrop]'
headLine = 'CROPSWAT'
p085 = setParam(nHeader, plygn, defaultP, table,
                headLine)

cursor.close()
conn.close()
return p012, p071, p072, p073, p074, p075, p076,
        p077, p078, p079, p080, p081, p082,
        p083, p084, p085

## ----- Parametrar - Slut -----
## -----

def iniSelec(nHeader, namedList):
    unvalid = []
    totalHRU = []
    iniXtra = []
    for plygn in namedList:
        h = plygn[nHeader.index('iniName')]
        if h not in unvalid:
            totalHRU.append(plygn)
            unvalid.append(h)
        else:
            iniXtra.append(plygn)
    return totalHRU, iniXtra

def setInfo(nHeader, plygn):
    delId = plygn[nHeader.index('ID')]
    blockId = int(plygn[nHeader.index('HEL_ID')])
    soilId =
        int(plygn[nHeader.index('SWATSOIL')])
    if plygn[nHeader.index('CROPSWAT')] != ' ':
        cropId =
            int(plygn[nHeader.index('CROPSWAT')])
        )
    else:
        cropId = 0
    delId = '%01d' % delId
    blockId = '%03d' % blockId
    soilId = '%03d' % soilId
    cropId = '%03d' % cropId

    return delId, blockId, soilId, cropId

def setSoilP(soil, paramValue):
    conn = pyodbc.connect('DRIVER={Microsoft
        Access Driver (*.mdb,
        *.accdb)};DBQ='+DBfile, autocommit =
        True)
    global cursor

```

```

cursor = conn.cursor()
table = 'SELECT [SOIL ID] AS param,
            [' + paramValue + '] AS paramV FROM
            [hydraulic properties standard soils]'
cursor.execute(table)
for row in cursor:
    if type(row.param) == unicode:
        row.param = row.param.encode('8859')
    if soil == row.param:
        p = row.paramV
        break
return p
cursor.close()
conn.close()

def genSoilP(soil, iniName):
    name,ext = iniName.split('.')
    soilPFile = open(destDir+name+'P'+".dat",'w')

    # Param1
    paramValue = 'XPSI'
    soilP1 = setSoilP(soil, paramValue)

    # Param2
    paramValue = 'A0t'
    soilP2 = setSoilP(soil, paramValue)
    soilP2 = '%.2f' % soilP2

    # Param3
    paramValue = 'A1t'
    soilP3 = setSoilP(soil, paramValue)
    soilP3 = '%.3f' % soilP3

    # Param4
    paramValue = 'NVAR'
    soilP4 = setSoilP(soil, paramValue)
    soilP4 = '%.1f' % soilP4

    # Param5
    paramValue = 'SATC'
    soilP5 = setSoilP(soil, paramValue)
    soilP5 = '%.3f' % soilP5

    # Param6
    paramValue = 'LAMBDA'
    soilP6 = setSoilP(soil, paramValue)
    soilP6 = '%.2f' % soilP6

    # Param7
    paramValue = 'RES'
    soilP7 = setSoilP(soil, paramValue)
    soilP7 = '%.2f' % soilP7

    # Param8

```

```

paramValue = 'PORO'
soilP8 = setSoilP(soil, paramValue)
soilP8 = '%.2f' % soilP8

# Param9
paramValue = 'PSIE'
soilP9 = setSoilP(soil, paramValue)
soilP9 = '%.2f' % soilP9

# Param9
paramValue = 'PSIE'
soilP9 = setSoilP(soil, paramValue)
soilP9 = '%.2f' % soilP9

# Param10
paramValue = 'WILT'
soilP10 = setSoilP(soil, paramValue)
soilP10 = '%.2f' % soilP10

# Param11
paramValue = 'SATCT'
soilP11 = setSoilP(soil, paramValue)
soilP11 = '%.3f' % soilP11

line1 = str(soilP1)
soilPFile.write(line1+'\n')
line2 = str(soilP2)
soilPFile.write(line2+'\n')
line3 = str(soilP3)
soilPFile.write(line3+'\n')
line4 = 'XXXX'+ '001'+ '001'
soilPFile.write(line4+'\n')
line5 = ' '+ '0'+ '25'+ '1'+
        '+str(soilP4)+str(soilP5).rjust(7)+str(soilP
6).rjust(8)+str(soilP7).rjust(8)+str(soilP8)
.rjust(8)+str(soilP9).rjust(8)+str(soilP10).
rjust(8)+str(soilP11).rjust(8)
soilPFile.write(line5+'\n')
line6 = ' '+ '25'+ '75'+ '1'+
        '+str(soilP4)+str(soilP5).rjust(7)+str(soilP
6).rjust(8)+str(soilP7).rjust(8)+str(soilP8)
.rjust(8)+str(soilP9).rjust(8)+str(soilP10).
rjust(8)+str(soilP11).rjust(8)
soilPFile.write(line6+'\n')
line7 = ' '+ '75'+ '100'+ '1'+
        '+str(soilP4)+str(soilP5).rjust(7)+str(soilP
6).rjust(8)+str(soilP7).rjust(8)+str(soilP8)
.rjust(8)+str(soilP9).rjust(8)+str(soilP10).
rjust(8)+str(soilP11).rjust(8)
soilPFile.write(line7+'\n')
line8 = ' '+ '100'+ '300'+ '1'+
        '+str(soilP4)+str(soilP5).rjust(7)+str(soilP
6).rjust(8)+str(soilP7).rjust(8)+str(soilP8)

```

```

.rjust(8)+str(soilP9).rjust(8)+str(soilP10).
rjust(8)+str(soilP11).rjust(8)
soilPFile.write(line8+'\n')

```

```
soilPFile.close()
```

```

def createRunList(nHeader, totalHRU):
    unvalid = []
    listFile = open(destDir+'RunList'+".txt",'w')
    row = 'IniName' + ' ' + 'CropID'+'\n'
    listFile.write(row)
    for plygn in totalHRU:
        namn = plygn[nHeader.index('iniName')]
        cropId =
            plygn[nHeader.index('CROPSWAT')]
        cropId = '%03d' % cropId
        if namn not in unvalid:
            unvalid.append(namn)
            row = namn + ' ' + cropId+'\n'
            listFile.write(row)
    listFile.close()

```

```

def createDownList(header, info4):
    listFile2 =
        open(destDir+'downstreamList'+".txt",'w'
        )
    skrivRad = 'DelID '+'DownStreamID'\n'
    listFile2.write(skrivRad)

```

```

y = 0
for count in range(0, len(info4)):
    if info4[count][header.index('ID')] > y:
        y = info4[count][header.index('ID')]

```

```

for i in range(0, y):
    for j in range(0, len(info4)):
        if info4[j][header.index('ID')] == i+1:
            a, b = info4[j][header.index('ID')],
            info4[j][header.index('DOWNST_ID')]
            a = '%01d' % a
            b = '%01d' % b

```

```

        skrivRad = a + ' ' + b +'\n'
        listFile2.write(skrivRad)
        break

```

```
listFile2.close()
```

```

## ----- Programmets sekvenser -----
-----
-----

```

```

def createIni():                                ##
    Main Function

    header, gisList = readGis()
        ## copying the attribute table from Gis
        into a list
    cropIDList = setCropId(header, gisList)
        ## Changes the crop name to crop ID
        according to the dictionary 'CROPSWAT'
    soilIDList = setSoilId(header, cropIDList)
        ## Changes the soil name to soil ID
        according to the dictionary 'SWATSOIL'
    forestIDList = setLandUseId(header, soilIDList)
        ## Defines which crop which is located
        on the polygon
    sameSoilList = setSoilType(header, forestIDList)
        ## Transforms all of the HRU's type of
        soil into the most frequent (largest
        coverage) soil type inside of one jbb
    cropHeader, cropInfo = readXl()
        ## copying the information from the jbb
        database into a list
    nHeader, croppedList = getCrops(header,
        sameSoilList, cropHeader, cropInfo) ##
        Derives the crops from the jbb database
        list for each HRU located on cropland
    nHeader, namedList = setIniMark(nHeader,
        croppedList)    ## Adds one field to all
        of the HRU's and names it (And also to
        the header)
    createIdList(nHeader, namedList)
        ## Creates a list which defines which
        subcatchment each file belongs to
    totalHRU, iniXtra = iniSelec(nHeader,
        namedList)

    print 'Writing ini-files...'
    for plygn in range(0, len(totalHRU)):
        iniFile =
            open(destDir+totalHRU[plygn][nHeader.i
            ndex('iniName')], 'w')

        p012, p071, p072, p073, p074, p075, p076,
        p077, p078, p079, p080, p081, p082,
        p083, p084, p085 = getDB(nHeader,
        totalHRU[plygn])    ## Retrieves the
        parameters for the current HRU
        if
            totalHRU[plygn][nHeader.index('MARK
            ANVNAM')] == 'Skog':
                p022 = 16
                p022 = '%.8f' % p022
            else:
                p022 = 2

```

```

p022 = '%.9f' % p022
delId, blockId, soilId, cropId =
    setInfo(nHeader, totalHRU[plygn])

    genSoilP(totalHRU[plygn][header.index('
    SWATSOIL')],
    totalHRU[plygn][header.index('iniName')
    ])

    content = str(" ! Model name
    1 1
    SOIL
    ! Model version
    2 1
    7.3
    ! Run number
    3 1
    8
    ! Parameter values P
    4 205
    20.00000000 -10.00000000
        0.5000000075E-01 -5.000000000
        1.000000000 1.000000000
    10.00000000 10.00000000 -1.000000000
        1.000000000 1.000000000
        "+p012+"
    20.00000000 0.5000000000 8.000000000
        -1.000000000 -2.000000000
        0.000000000
    0.000000000 0.000000000 3.000000000
        "+p022+" 100000.0000
        0.000000000
    -5.000000000 4.000000000 -10.00000000
        -10.00000000 0.000000000 -
        1.000000000
    0.2000000030 5.000000000 100.0000000
        8.000000000 0.000000000
        59.90000153
    4.000000000 1.120000005 0.1400000006
        2.000000000 -2.000000000
        50.00000000
    0.5000000000 0.4000000060 0.2000000030
        0.1000000015 0.1000000015
        0.5000000075E-01
    0.5000000075E-01 0.5000000075E-01
        0.2999999933E-01 0.1999999955E-01
        0.000000000 -0.1000000015
    -0.8000000119 -1.200000048 121.0000000
        180.0000000 250.0000000
        260.0000000
    0.9999999776E-02 0.9999999776E-02
        0.9999999776E-02 0.9999999776E-02
        0.9999999776E-02 50.00000000

```



```

50.00000000 50.00000000 50.00000000
50.00000000 ""+p071+"" ""+p072+""
""+p073+"" ""+p074+"" ""+p075+""
""+p076+"" ""+p077+"" ""+p078+""
""+p079+"" ""+p080+"" ""+p081+""
""+p082+"" ""+p083+"" ""+p084+""
""+p085+"" 1.000000000 1.000000000
1.000000000 1.000000000
2.500000000
0.1000000015 5.000000000 0.8999999762
100.0000000 0.8999999762
100.0000000
3.000000000 0.1000000015
0.1000000012E-06 3.000000000
0.7000000030E-01 0.2859999995E-05
0.5000000000 0.2500000000 0.5000000000
0.2500000000 2.0000000000
0.0000000000
0.50000000075E-01 0.50000000075E-01
0.50000000075E-01 0.50000000075E-01
0.50000000075E-01 0.50000000075E-01
0.50000000075E-01 0.50000000075E-01
0.50000000075E-01 0.50000000075E-01
0.50000000075E-01 0.50000000075E-01
0.50000000075E-01 1.000000000
1.000000000 25.00000000 50.00000000
25.00000000 200.0000000
0.000000000
1.000000000 0.8000000119 0.4000000060
500.0000000 0.2000000030
0.000000000
2.000000000 10.00000000 2.000000000
4.000000000 10.00000000
365.0000000
10.00000000 2.000000000 18.00000000
8.000000000 10.00000000
0.000000000
70.00000000 0.6999999881 0.5000000000
0.000000000 0.000000000
100.0000000
50.00000000 4.000000000 100.0000000
20.00000000 0.000000000
20.00000000
0.5000000000 1.000000000 0.6999999881
5000000.000 100.0000000
0.1999999955E-01
50.00000000 5.000000000 10.00000000
10.00000000 10.00000000
10.00000000
10.00000000 0.9999999776E-02
100.0000000 10.00000000
100.0000000 0.9999999776E-02

```

```

50.00000000 1.000000000 0.000000000
0.000000000 0.000000000
0.000000000
0.000000000 0.000000000 0.000000000
0.000000000 0.000000000
0.000000000
0.000000000 0.000000000 0.000000000
0.000000000 0.000000000
0.000000000
0.000000000 0.000000000 0.000000000
0.000000000 0.000000000
0.000000000
0.000000000
! Index in P
8 117
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 18 20 21 22
23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42
43 44 54 57 61 66 71 76 81 86 90 91 92
93 94 95 96 97 98 99
100 101 102 103 104 126 127 128 133 134 135
136 137 138 139 140 141 142 143 144
145 146 147 148 149 150 151 152 153 154 155
156 157 158 159 160 161 162 163 164
165 166 167 168 169 170 171 176 177 178 179
180 181 182 183 184 206
! Parameter names P
7 116
ALBEDO CHOEN RFRACLOW CIECO
IFLOWR CNUMD CTEMP1
DDIST
DDRAIN EGPSI EPRAT FCOND FDF
FWFRAC GEOTER GFLEV
GFLOW GWSOF GWSOL HEIGHT
HPAMP HPBAS HPCUT
HPLAY
HPMAX HPZERO HUMUS IGWLEV
INTLAI INTRIS IPOT ITEMS
ITHETA LATID NUMLAY PRECA0
PRECA1 PRLIM PSLIM RALAI
RNTLAI ROOTF ROOTDEP ROOTT
ROUGHV RSV LAIV DISPLV
DAYNUM CFORM SAGEM1 SAGEM2
SAGEZP SAGEZQ SD1OL
SD2OM
SDENS SLWL0 SMAFR SMRIS
SMTEM SRET STCON
SURDEL
THICK UNUM UPROF UTHICK VC
WUPATE WUPBTE WUPCRI
WUPF WUPFB XADIV XINFLI
XLOOP XNLEV YAINF YCH

```

YFAINF AWS YPHAS YTAM
 YTAMP NSOURCE AHR
 ACLOUD
 ASCALE SOILCOVER SIFRAC PSIRS
 ISTOREMIN ISTOREL IRRIRATE
 IRRIAM
 XTGD PMAX UPMOV PADDIND
 CANDENSM CONDRIS CONDVDP
 CONDMAX
 TSUM TSUMCRIT HCANOPYV
 ROUGHLMIN PSIBOTTOM SALTINIC
 ALPHAHT SPOOLMAX
 SPCOVTOT WUPCRISAT WUPREDSAT
 SCALE
 ! State variable names X
 10 10
 WATER HEAT PLANT STREAM
 SURPOOL HSNOW WSNOW
 SALTSUR
 SALTX WATP
 ! Index in X
 11 11
 1 23 45 46 47 48 49 50 51 73 96
 ! State variable output index for PG-bin file
 19 15
 1 2 3 4 5 23 24 25 26 27 45 46 47 48
 49
 ! Flow variable output index for PG-bin file X
 51 15
 1 2 3 4 5 23 24 25 26 27 45 46 47 48
 49
 ! Flow variable names T
 12 25
 WFLOW EFLOW WUPRATE DRIVF
 INFIL EVAG DFLOW PUMP
 SPOOLA SPOOLINF SURR HEATSINK
 DEEPPERC DEEPCONV SALTPIN
 SALTSURR
 SALTPOUT SALTIN SALTFL SALTDF
 SALTDEEP WFLOWP WFLOWPN
 WFLOWNP
 INFREEZE
 ! Index in T
 13 26
 1 22 43 53 54 55 56 78 79 80 81 82
 104 105 106 107
 108 109 110 131 153 154 175 196 217 240
 ! Flow variable output index for PG-bin file
 20 26
 1 2 3 4 22 23 24 25 43 44 45 46 47 53
 54 55 79 80 81 82
 83 84 85 86 104 105
 ! Flow variable output index for PG-bin file T
 52 26

1 2 3 4 22 23 24 25 43 44 45 46 47 53
 54 55 79 80 81 82
 83 84 85 86 104 105
 ! Auxiliary variable names G
 14 52
 TEMP THQUAL THETA PSI INTCAP
 INTERC EINTPOT EACTI
 ISTORE RA ROUGH DISPL RS
 WUPPOT EACT ETR
 EVAPO VPD RNTG SENS LATENT
 SURFMOS LAI SATLEV
 PREC TOTQ PIPEQ PERC
 SPOOLCOV TQUALP DENSS
 SWATS
 SAGE SWELL FROSTBU FROSTBL
 TTSTEP DINFIL RAC VPS
 VPA RSSOIL ROOTDEPTH RICH
 EBAL ETRPSI ETRTEM PAI
 HCANOPY SALTCONC THETATOT
 BYPASS
 ! Index in G
 15 53
 1 23 45 67 89 90 91 92 93 94 95 96
 97 98 99 100
 101 102 103 104 105 106 107 108 109 110
 111 112 113 114 115 116
 117 118 119 121 123 124 125 126 127 128
 129 130 131 132 133 134
 135 136 158 180 202
 ! Auxiliary variable output index for PG-bin file
 21 51
 1 2 3 4 5 23 24 25 26 27 45 46 47 48
 49 67 68 69 70 71
 89 90 91 92 93 94 95 96 97 98 99 100 101
 102 103 107 109 113 114 115
 116 117 119 120 121 122 123 124 125 128 129
 ! Auxiliary variable output index for PG-bin file G
 53 51
 1 2 3 4 5 23 24 25 26 27 45 46 47 48
 49 67 68 69 70 71
 89 90 91 92 93 94 95 96 97 98 99 100 101
 102 103 107 109 113 114 115
 116 117 119 120 121 122 123 124 125 128 129
 ! Driving variable names D
 16 14
 EPOT PRECMM TA TD HR
 WS RNT CLOUDN
 RIS IRIG SPSOURCE WSOURCE
 SALTDEPC DRAINLEV
 ! Index in D
 17 15
 1 2 3 4 5 6 7 8 9 10 11 12 34
 35 36
 ! Driving variable output index for PG-bin file

```

22 10
1 2 3 4 5 6 7 8 9 10
! Driving variable output index for PG-bin file D
54 10
1 2 3 4 5 6 7 8 9 10
! Switches values SW
5 40
1 1 1 1 0 0 0 2 1 0
0
0 3 1 1 2 1 1 1 1 1
1
0 1 0 0 0 3 1 0 0 1
1
0 0 0 0 1 0 0
! Switches name SW
6 40
AVERAGEX AVERAGET AVERAGEG
AVERAGED OUTFORN INSTATE
OUTSTATE DRIVPG
LISALLV CHAPAR ADDSIM VALIDPG
ROOTDIST WATEREQ HEATEQ
WUPTAKE
HEATWF GWFLOW FRLOADP FRLIMINF
FRLIMUF FRINTERA FRSWELL
DDAILY
SUREBAL CRACK ATIRRIG EVAPOTR
INWATER INHEAT HEATPUMP
SNOW
INTERCEPT ROUGHNESS PLANTDEV
RSCALC VISALLOUT UNITG
SALT FRPREFL
! Reading identification string
24 1
A
! Time step
26 1
4
! Time specifications
25 3
1052687520 1055931840 1440
! Driving variable file
31 1
CLIMATE.BIN
! Parameter file
32 1
ALLOUT.PAR
! Translation file
33 1
SOIL.TRA
! Final state variable file
35 1
SOIL.FIN
! Hydraulic soil properties
38 1

```

```

SOIL.P.DAT
! Thermal soil properties
39 1
THCOEF.DAT
! Additional driving variable file no 1 temporal
development of crops
40 1
CROPDATA.BIN
! Initial temperature profile
41 1
TEMP.DAT
! Initial tension profile
42 1
MPOT.DAT
! Additional driving variable file no 2
Heatextraction
43 1
HEATEXTR.BIN
! Salt concentration file
44 1
SALTCONC.BIN
! Drainlev
45 1
DRAINLEV.BIN
! Psibott
46 1
PSIBOTT.BIN
! INFO_X
60 1
25097
! ISILENT
61 1
0
! Aro number
101 1
2
! Delaro number
102 1
""+delId+""
! Block nr
103 1
""+blockId+""
! Soil type nr
104 1
""+soilId+""
! Crop type nr
105 1
""+cropId+""
")
iniFile.write(content)
iniFile.close()

print 'Done!'
print '-----'

```

```

print 'A total of', len(totalHRU) + len(iniXtra),
      'polygons\n', 'Number of ini-files:',
      len(totalHRU)
print '-----'
## if len(totalHRU) > 400:
##     safetyMessage = "\n!!! The number of ini-
        files exceeds 400. Attempt if possible to

```

```

        reduce the number of ini-files by
        increasing the area filter. !!!"
##     print safetyMessage.upper()
        createRunList(nHeader, totalHRU)
        createDownList(nHeader, totalHRU)

createIni()

```

BILAGA 2 - INSTÄLLNINGSFILEN

```
! Model name
1 1
SOIL
! Model version
2 1
7.3
! Run number
3 1
8
! Parameter values P
4 205
20.00000000 -10.00000000 0.5000000075E-01 -5.000000000 1.000000000 1.000000000
10.00000000 50.00000000 -2.000000000 1.000000000 1.000000000 20.00000000
20.00000000 0.500000000 8.000000000 -1.000000000 -2.000000000 0.000000000
0.000000000 0.000000000 3.000000000 2.000000000 100000.0000 0.000000000
-5.000000000 4.000000000 -10.00000000 -10.00000000 0.000000000 -0.1000000015
0.2000000000 5.000000000 100.0000000 8.000000000 0.000000000 59.90000153
4.000000000 1.120000005 0.1400000006 2.000000000 -2.000000000 50.00000000
0.5000000000 0.4000000060 0.2000000030 0.1000000015 0.1000000015 0.5000000075E-01
0.5000000075E-01 0.5000000075E-01 0.2999999933E-01 0.1999999955E-01 0.000000000 -0.1000000015
-0.8000000119 -1.200000048 121.0000000 180.0000000 250.0000000 260.0000000
0.9999999776E-02 0.9999999776E-02 0.9999999776E-02 0.9999999776E-02 0.9999999776E-02 50.00000000
50.00000000 50.00000000 50.00000000 50.00000000 0.000000000 5.000000000
1.000000000 4.000000000 1.000000000 0.000000000 0.500000000 0.0700000003
0.3499999940 0.1000000015 100.0000000 164.0000000 165.0000000 225.0000000
226.0000000 1.000000000 1.000000000 1.000000000 1.000000000 2.500000000
0.1000000015 5.000000000 0.8999999762 100.0000000 0.8999999762 100.0000000
3.000000000 0.1000000015 0.1000000012E-06 3.000000000 0.7000000030E-01 0.2859999995E-05
0.5000000000 0.2500000000 0.5000000000 0.2500000000 2.000000000 0.000000000
0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 0.5000000075E-01
0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 0.5000000075E-01
0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 0.5000000075E-01 1.000000000
1.000000000 25.00000000 50.00000000 25.00000000 200.0000000 0.000000000
1.000000000 0.8000000119 0.4000000060 500.0000000 0.2000000030 0.000000000
2.000000000 10.00000000 2.000000000 4.000000000 10.00000000 365.0000000
```

Figur 2.1. Utdrag ur inställningsfilen. Här visas några av de 205 parametervärdena som är möjliga att specificera

```
- - - - -
! Switches values SW
5 40
1 1 1 1 0 0 0 2 1 0 0
0 3 1 1 2 1 1 1 1 1 1
0 1 0 1 0 3 1 0 0 1 1
0 0 0 0 1 0 0
! Switches name SW
6 40
AVERAGEG AVERAGED OUTFORN INSTATE OUTSTATE DRIVPG
LISALLV CHAPAR ADDSIM VALIDPG ROOTDIST WATEREQ HEATEQ WUPTAKE
HEATWF GWFLOW FRLOADP FRLIMINF FRLIMUF FRINTERA FRSWELL DDAILY
SUREBAL CRACK ATIRRIG EVAPOTR INWATER INHEAT HEATPUMP SNOW
INTERCEPT ROUGHNESS PLANTDEV RSCALC VISALLOUT UNITG SALT FRPREFL
```

Figur 2.2. Utdrag ur inställningsfilen. Här visas hur de olika parameterväljarna har angetts

BILAGA 3 - JORDARTSINFORMATION

1000										
0.54										
0.025										
XXXX		001	001							
	0 25	1	1.0	0.060	0.13	9.00	47.50	37.30	27.20	12.000
	25 75	1	1.0	0.060	0.13	9.00	47.50	37.30	27.20	12.000
	75100	1	1.0	0.060	0.13	9.00	47.50	37.30	27.20	12.000
	100300	1	1.0	0.060	0.13	9.00	47.50	37.30	27.20	12.000

Figur 3.1. Utdrag ur filen med jordartsegenskaper

BILAGA 4 - GRÖDINFORMATION

ID	Grödkod	crop	Gröda	Hestadspec	albedo	cform1	cform2	cform3
1	5	Spring wheat	Vårvete		20	1	1	1
2	2	Spring barley	Vårkorn	finns	20	1	1	1
3	21	Spring rape	Vårraps	finns	20	1	1	1
4	4	Winter wheat	Höstvete	finns	20	1	1	1
5	8	Winter rye	Höstråg		20	1	1	1
6	3	Oats	Havre	finns	20	1	1	1
7	20	Winter rape	Höstraps		20	1	1	1
8	45	Potatoes	Potatis		20	1	1	1
9	19	Sugar beets	Sockerbeta		20	1	1	1
10	50	Ley	Vall	finns	20	1	1	1
11	1	Winter barley	Höstkorn		20	1	1	1
12	30	Peas	Ärter		20	1	1	1
13	24	Sunflower	Solros		20	1	1	1
14	9	Maize	Majs	finns	20	1	1	1
21	60	Ley	Träda	finns	20	1	1	1
22	52	Ley	Betesmark	finns	20	1	1	1
23	77	Ley	Skyddszon 3 Lmzon	finns	20	1	1	1
24	23	Spring rape	Vårrybs	finns	20	1	1	1
25	80	Forest	Skog	finns	10	1	1	1
*(New)								

Figur 4.1. Utdrag ur parameterdatabasen innehållandes de grödor och markanvändningar som definieras i indata till SOIL

daynum1	daynum2	daynum3	daynum4	daynum5	displv1	displv2	displv3
105	120	180	227	228	0	0,600000023841858	0,600000023841858
105	120	180	227	228	0	0,600000023841858	0,600000023841858
105	120	180	227	228	0	0,600000023841858	0,600000023841858
100	205	237	238	274	0,01999999996	0,600000023841858	0,600000023841858
100	205	237	238	274	0,01999999996	0,600000023841858	0,600000023841858
105	120	180	227	228	0	0,600000023841858	0,600000023841858
100	205	237	238	274	0,01999999996	0,600000023841858	0,600000023841858
131	140	193	249	263	0	0,300000011920929	0,300000011920929
124	170	230	304	305	0	0,25	0,300000011920929
100	164	165	225	226	0	0,5	0,0700000002980232
100	205	237	238	274	0,01999999996	0,600000023841858	0,600000023841858
105	120	180	227	228	0	0,600000023841858	0,600000023841858
105	120	180	227	228	0	0,600000023841858	0,600000023841858
105	120	180	227	228	0	0,600000023841858	0,600000023841858
100	164	165	225	226	0	0,5	0,0700000002980232
100	164	165	225	226	0	0,5	0,0700000002980232
100	164	165	225	226	0	0,5	0,0700000002980232
105	120	180	227	228	0	0,600000023841858	0,600000023841858
10	120	180	227	360	11	11	11
*(New)							

Figur 4.2. Utdrag ur parameterdatabasen innehållandes de dagnummer och förflyttningshöjd som definieras i indata till SOIL

Date	RSV	LAIV	DISPLV	ROUGHV	ROOTDEP
2002-08-15 12:00	150	0	0	0,005	0
2002-08-16 12:00	150	0	0	0,005	0
2003-04-15 12:00	150	0	0	0,005	-0,2
2003-04-30 12:00	40	5	0,6	0,08	-0,7
2003-06-29 12:00	100	3	0,6	0,08	-0,7
2003-08-15 12:00	150	0	0	0,005	0
2003-08-16 12:00	150	0	0	0,005	0
2004-04-14 12:00	150	0	0	0,005	-0,2
2004-04-29 12:00	40	5	0,6	0,08	-0,7
2004-06-28 12:00	100	3	0,6	0,08	-0,7
2004-08-14 12:00	150	0	0	0,005	0
2004-08-15 12:00	150	0	0	0,005	0
2005-04-14 12:00	150	0	0	0,005	-0,2
2005-04-29 12:00	40	5	0,6	0,08	-0,7
2005-06-28 12:00	100	3	0,6	0,08	-0,7
2005-08-14 12:00	150	0	0	0,005	0
2005-08-15 12:00	150	0	0	0,005	0
2006-04-14 12:00	150	0	0	0,005	-0,2
2006-04-29 12:00	40	5	0,6	0,08	-0,7
2006-06-28 12:00	100	3	0,6	0,08	-0,7
2006-08-14 12:00	150	0	0	0,005	0
2006-08-15 12:00	150	0	0	0,005	0
2007-04-14 12:00	150	0	0	0,005	-0,2
2007-04-29 12:00	40	5	0,6	0,08	-0,7
2007-06-28 12:00	100	3	0,6	0,08	-0,7
2007-08-14 12:00	150	0	0	0,005	0

Figur 4.3. Utdrag ur en indatafil till SOIL innehållandes grödsspecifik information

BILAGA 5 - INSTRUKTIONSFILEN ID

DelID	IniName	BlockID	SoilID	CropID	delArea	OrgArea	TotArea	FullArea
1	525.ini	64845107614	007	050	26183.5	1215290.0	1164000.722	7814085.41532
1	526.ini	64845104418	011	050	6513.17008929	1215290.0	1164000.722	7814085.41532
1	527.ini	64845104836	007	004	51081.1200061	1215290.0	1164000.722	7814085.41532
1	528.ini	64855104522	007	050	985.054016113	1215290.0	1164000.722	7814085.41532
1	529.ini	64845108271	007	004	1515.58967538	1215290.0	1164000.722	7814085.41532
1	529.ini	64845108271	007	004	29604.5183257	1215290.0	1164000.722	7814085.41532
1	530.ini	64855100962	007	003	33090.3066883	1215290.0	1164000.722	7814085.41532
1	531.ini	64855100962	007	060	10027.3656631	1215290.0	1164000.722	7814085.41532
1	532.ini	64845107155	007	060	25176.5173077	1215290.0	1164000.722	7814085.41532
1	533.ini	64845105157	010	060	12588.3322998	1215290.0	1164000.722	7814085.41532
1	534.ini	64845104963	010	060	7834.29171143	1215290.0	1164000.722	7814085.41532
1	535.ini	64845103084	010	050	108.012359732	1215290.0	1164000.722	7814085.41532
1	536.ini	64845103084	010	060	849.047975272	1215290.0	1164000.722	7814085.41532
1	537.ini	64845104062	010	060	10247.87759	1215290.0	1164000.722	7814085.41532
1	538.ini	64845108710	007	050	6347.45603841	1215290.0	1164000.722	7814085.41532
1	539.ini	64845105911	013	052	10984.5996094	1215290.0	1164000.722	7814085.41532
1	540.ini	64855104249	010	060	40085.5	1215290.0	1164000.722	7814085.41532
1	541.ini	64855112612	007	060	1537.35800781	1215290.0	1164000.722	7814085.41532
1	542.ini	64855112612	007	004	6610.63943359	1215290.0	1164000.722	7814085.41532
1	543.ini	64845105923	013	052	68055.296875	1215290.0	1164000.722	7814085.41532
1	544.ini	64845103805	013	052	6325.50976563	1215290.0	1164000.722	7814085.41532
1	545.ini	64845106192	007	052	16396.6992188	1215290.0	1164000.722	7814085.41532
1	546.ini	64845107875	007	050	6494.45716712	1215290.0	1164000.722	7814085.41532
1	547.ini	64845107875	007	004	17023.3498471	1215290.0	1164000.722	7814085.41532

Figur 5.1. Utdrag ur instruktionsfilen ID. Här listas areorna av de beräkningsenheter som tillhör delavrinningsområde 1