



UPPSALA  
UNIVERSITET

W13029

Examensarbete 30 hp  
September 2013

# Bestämning av vattendelare i urban miljö

Metod för avgränsning av avrinningsområden  
i ArcGIS utgående från dagvattenssystemet

---

Karl-Martin Calestam



## REFERAT

### Bestämning av vattendelare i urban miljö – Metod för avgränsning av avrinningsområden i ArcGIS utgående från dagvattensystemet

*Karl-Martin Calestam*

Växande städer leder till att naturmark omvandlas till stadsmiljöer. Det skapar mer hårdgjorda ytor och därmed ökade volymer dagvatten som behöver hanteras. Modellering av dagvattennätet används ofta för att bedöma kapaciteten och risk för översvämningar. Avrinningsområdets storlek är en avgörande parameter för att bedöma hur mycket vatten som kommer till en viss ledning. Traditionellt har automatisk bestämning av avrinningsområden gjorts enbart utifrån topografin. I områden med dagvattennät styr däremot ledningarnas sträckning i första hand hur vattnet rinner, och de följer inte nödvändigtvis terrängen. ArcHydro Tools, som är ett tillägsprogram till ArcMap, har utvecklats för att beräkna topografiska avrinningsområden. Genom att programmera en tillägsfunktion i Python har processen i ArcHydro Tools anpassats för att kunna ta hänsyn till dagvattennätet. Dagvattennätet approximerades som vattendrag. Tryckledningar och tunnlar tillåter inte något inflöde av vatten. Därför beskrevs dessa som ändpunkter i ledningsnätet, varifrån vattnet inte rann vidare. Funktionen som skapades i det här examensarbetet tillät att vattnet stannade i dessa punkter, till skillnad från i den ursprungliga metoden. Den utgick istället från att vattnet rann till kanten av det undersökta området, vilket i de här fallen skulle ha resulterat i felaktigt avgränsade avrinningsområden.

Tillägsfunktionen anpassades för att användas som övriga funktioner i ArcMap och testades på data över Lidingö stad. Resultatet visade att det var möjligt att utnyttja topografiskt baserade metoder för bestämning av avrinningsområden så att avgränsningen istället skedde med utgångspunkt i dagvattennätet. Det är ett steg mot att effektivisera processen för bestämning av avrinningsområden för användning vid hydrologisk modellering av dagvattennätet. Behovet av manuell bearbetning minskas. Ett försök att i ArcMap implementera en funktion för efterbehandling av avrinningsområden gjordes också. Funktionen letar upp avrinningsområden som bedöms som för små och lägger ihop dem med ett närliggande område. Avrinningsområden som beräknats med den modifierade metoden har senare använts vid modellering av dagvattennätet i Molkom, Värmland.

**Nyckelord:** avrinningsområde, dagvattennät, vattendelare, ArcHydro Tools, Python, hydraulisk modellering, Lidingö

*Institutionen för geovetenskaper, Luft-, vatten- och landskapslära, Uppsala universitet,  
Geocentrum, Villavägen 16  
SE-752 36 Uppsala  
ISSN 1401-5765*

## **ABSTRACT**

### **Watershed delineation in urban catchment areas – method for delineating catchment areas in ArcMap based on storm-water drains**

*Karl-Martin Calestam*

Growing populations result in expanding cities. An increase in the amount of impervious surfaces in the area will follow and thereby generate more storm water. The capacity of the drainage system can be evaluated using hydraulic modelling. The model highly depends on the catchment areas, which will determine the water volume each pipe section receive. Watershed delineation is usually done based on the topography. However, if a storm water drainage system is present, it may route the water in a different direction than the slope indicates. ArcHydro Tools is an extension to ArcMap and is commonly used to delineate catchment areas. The method heavily relies on topography during this process. A function has been developed during the course of this project to allow for the drainage system to be the primary source of information for watershed delineation.

This function made sure that outlets in the model were to be evaluated as such, even if they happen to be located in the middle of the area of interest. The water is therefore not necessarily routed to the edge of the elevation model, but can be allowed to stay at the appropriate position. In order to do this, the drainage system was represented as a stream network.

The new method was applied to test data supplied by Lidingö city, Stockholm, Sweden, and included elevation data and information about the drainage system. The result implies that it is indeed possible to use the drainage system as a base for delineation of catchment areas. A more efficient method of calculating catchment areas will reduce the required amount of manual processing, thereby saving time and resources. Another function, for finishing up the resulting catchment areas, is proposed but not fully implemented. The whole process was used to delineate catchment areas for Molkom, County of Värmland, Sweden. The resulting watersheds were later successfully used for modelling the storm water drains in the area.

**Keywords:** catchment area, storm water drain, watershed delineation, ArcHydro Tools, Python, hydraulic modelling, Lidingö

*Department of Earth Sciences, Program for Air, Water and Landscape Science.  
Uppsala University  
Villavägen 16, SE-752 36 Uppsala  
ISSN 1401-5765*

## FÖRORD

Det här examensarbetet omfattar 30 hp och har genomförts som en avslutning på utbildningen till civilingenjör i Miljö- och vattenteknik vid Uppsala universitet. Arbetet har utförts i samarbete med Avdelningen för mark- och vattenresurser på Tyréns AB i Stockholm, med handledning från Lars Marklund. Ämnesgranskare var Sven Halldin, professor i hydrologi vid Institutionen för geovetenskaper, Luft-, vatten- och landskapslära, Uppsala universitet.

Jag vill tacka min handledare Lars Marklund för stöd och idéer under projektets gång samt ämnesgranskare Sven Halldin för hjälp med vidare utformning av projektet och möjliga förbättringar. Jag vill även tacka Hans Hammarlund för att han delar med sig av sin aldrig sinande kunskap om hydrologisk modellering och övriga på Mark och vatten samt VA-avdelningarna för engagemang och trevligt sällskap under arbetets gång. Vidare vill jag tacka Lidingö stad och Karlstads kommun för tillåtelse att använda deras data under projektet och Sten Blomgren på DHI för han bistod med en studentlicens till Mike Urban. Ett särskilt tack vill jag ge till användarna Cyrus, Amber och Dan Mahr på forumet [gis.stackexchange.com](http://gis.stackexchange.com), vilka på helt frivillig basis har hjälpt mig när jag kört fast i min programmering. Jag skänker också en tanke åt Erik Ahlin för att ha initierat idén till det här projektet under sitt examensarbete. Slutligen vill jag tacka de som har ägnat nätterna åt korrekturläsning av den här rapporten; ingen nämnd, ingen glömd.

*Karl-Martin Calestam*

*Uppsala, 2013*

Copyright © Karl-Martin Calestam och Institutionen för geovetenskaper, Luft-, vatten- och landskapslära, Uppsala Universitet.

UPTEC W 13 029, ISSN 1401-5765

Publicerat digitalt vid Institutionen för geovetenskaper, Uppsala universitet, Uppsala, 2013.

## POPULÄRVETENSKAPLIG SAMMANFATTNING

Bebyggelse och hårdgjorda ytor såsom asfalterade vägar gör att vatten inte lika lätt kan rinna ner i marken när det regnar, vilket genererar större vattenvolymer än på naturmark. För att inte få problem med översvämningar och vattenskadade hus finns ofta ett ledningsnät som kan leda bort vattnet. När man vill beräkna hur mycket vatten som rinner till en viss ledning är det avgörande att känna till vilka delar av området som bidrar till just den delen av ledningen, ledningens avrinningsområde. Sedan kan systemet modelleras för en viss nederbördsmängd och det är möjligt att utifrån modellresultatet bedöma vattenmängden som kommer att rinna till ledningen. Avrinningsområden är en av de allra mest grundläggande parametrar som krävs för att beräkna vattenvolymer som genereras i ett område. Avrinningsområden i naturmark kan approximativt beräknas baserat på markens lutning. I stadsmiljö skapar ledningsnätet flödesvägar som inte nödvändigtvis stämmer överens med topografin. Därför kan inte samma metod användas för att avgränsa avrinningsområden där, utan att ledningsnätet tas med i beräkningen.

Lidingö stad har tillhandahållit höjddata och en modell över deras ledningsnät och avrinningsområden för Lidingö har beräknats. Beräkningarna har gjorts i ArcMap, ett program som hanterar och bearbetar digitala kartdata. För att bestämma avrinningsområden har ett tillägsprogram till ArcMap använts, ArcHydro tools. Verktöget är framtaget för bestämning av avrinningsområden på naturmark. Vattenflöden modelleras utifrån topografin, där allt vatten leds till ytterkanten av det undersökta området. Vissa anpassningar har gjorts så att metoden även ska vara användbar i stadsmiljö. Ledningarna har representerats som vattendrag genom att de sänkts ner i terrängen. Det innebär att allt vatten som rinner fram till ledningen därefter fortsätter i denna. Detta är egentligen en förenkling, vattnet kan i verkligheten endast rinna in i ledningen via brunnar och andra anslutningar.

Ett ledningsnät består inte enbart av ledningar med självfall, det förekommer också tunnlar och tryckledningar med pumpstationer. Eftersom dessa ledningar inte kan modelleras som vattendrag, de tillåter inte inflöde av nytt vatten från omgivningen, har de tagits bort från dagvattennätet innan detta sänktes ned i terrängen. Det innebär att ledningsnätet på motsvarande ställen ”tar slut”. Vid avgränsningen av avrinningsområden behöver funktionen få reda på att vattnet inte behöver ledas vidare från dessa punkter. Eftersom en sådan möjlighet inte fanns skapades en ny funktion till ArcMap i Python. Funktionen letar upp de punkter där vattnet ska stanna och sänker ner dem på samma sätt som med ledningsnätet. ArcHydro tools kan sedan identifiera dessa nedsänkta punkter och leda vattnet fram till dem men inte vidare. Korrekta avrinningsområden kan därefter beräknas utifrån ledningsnätet och tillgängliga höjddata.

Den här anpassningen av metoden i ArcHydro tools resulterade i avrinningsområden som tog hänsyn till ledningsnätet och hur vattnet rinner i detta. Metoden kan spara tid när avrinningsområden beräknas, jämfört med den nu dominerande arbetsgången med manuell avgränsning, och minska behovet av manuell bearbetning. Olika typer av efterbehandling kan minska det ytterligare. I den här studien har dessutom hanteringen av avrinningsområden som blir väldigt små undersökts. Genom att slå ihop dessa med närliggande områden kan storleken på filerna hållas nere, vilket är önskvärt ur beräkningssynpunkt. Funktionen kunde inte färdigställas inom tidsramen för projektet.

## DEFINITIONER

ArcHydro Tools	Ett tilläggsprogram till ArcMap med funktioner för hydrologiska beräkningar, däribland avrinningsområden.
Attribut	Information som tillhör en geometri. Det kan exempelvis vara olika mätdata eller beteckningar och dessa data lagras i en tabell.
Avrinningsområde	Den del av omgivningen som bidrar med vatten till den studerade punkten eller vattendraget.
Buffertområde	Område runt en punkt eller linje som också används i beräkningarna.
Dagvatten	Tillfälligt förekommande regn- eller smältvatten som avrinner på markytan.
Diskretisering	Övergång från kontinuerliga data till diskreta, exempelvis när markhöjder representeras med ett värde för varje meter medan det i naturen sker en kontinuerlig förändring.
Grundvattendelare	Grundvattnet rinner åt olika håll, och mot olika vattendrag, på olika sidor om grundvattendelaren.
Höjdmodell	En DEM (Digital elevation model), eller höjdmodell, är ett raster där varje cell har ett värde som representerar markhöjden i området.
Rasterdata	Används för visualisering av kontinuerliga data, såsom markhöjd eller hur föroreningar är fördelade i över en yta. Jämför vektordata.
Rinntid	Den tid det tar för vattnet att rinna en viss sträcka, antingen i en ledning eller i/på marken, vanligtvis avses den tid det tar för vatten att rinna från den mest avlägsna punkten i avrinningsområdet till områdets nedersta punkt; utloppspunkten.
Shapefil	System för lagring av geografiska data i vektorformat. Vad som vanligen kallas för en shapefil består av minst tre olika filer.

Sänka	Lågpunkt i terrängen varifrån vatten inte kan rinna vidare, vilken kan vara både skapad och finnas naturligt i terrängen.
Uppfångningslinje	En ledning som sänkts ned i höjdmodellens representation av terrängen innebär att vatten enbart kan rinna vidare i ledningens sträckning. Det är ett sätt att styra vattenflödet.
Utloppspunkt	Nedre ändpunkt i ledningsnätet. Begreppet används här antingen för att beteckna den punkt där ledningen tar slut eller för att en tunnel eller tryckledning har tagits bort som en del av processen. Vattenflöden modelleras enbart fram till en utloppspunkt.
Vattenväg	När algoritmen i <i>ArcHydro Tools</i> modellerar vattenflöden skapas fiktiva rinnvägar. Beroende på hur ett gränsvärde sätts definieras dessa sedan som vattenvägar, vilka ligger till grund för vidare beräkningar.
Vektordata	Används främst för visualisering av objekt med begränsad utbredning, såsom hus, vägar och punktformade objekt. Jämför rasterdata.
Ytvattendelare	Ytavrinnande vatten rinner åt olika håll och mot olika vattendrag på var sin sida om ytvattendelaren.



# INNEHÅLLSFÖRTECKNING

<b>1. INLEDNING</b> .....	<b>1</b>
1.1. SYFTE .....	2
<b>2. BAKGRUND</b> .....	<b>3</b>
2.1. DAGVATTEN OCH INFILTRATION.....	3
2.1.1. <i>Avrinning</i> .....	3
2.1.2. <i>Dimensionering av ledningsnät</i> .....	4
2.2. DATAHANTERING .....	5
2.2.1. <i>Objektmodellen</i> .....	5
2.2.2. <i>Fältmodellen</i> .....	6
2.2.3. <i>Referenssystem</i> .....	7
2.3. PROGRAMVARA.....	7
2.3.1. <i>Funktioner i ArcMap</i> .....	8
2.3.2. <i>ArcHydro Tools</i> .....	10
2.3.3. <i>Geografisk databehandling med Python</i> .....	14
<b>3. MATERIAL OCH METOD</b> .....	<b>16</b>
3.1. HÖJDDATA .....	16
3.2. HÖJDMODELLEN (DEM).....	16
3.3. LEDNINGSNÄTET .....	17
3.4. HANTERING AV UTLOPPSPUNKTER.....	17
3.5. BESTÄMNING AV AVRINNINGSSOMRÅDEN .....	19
3.6. HANTERING AV SMÅ AVRINNINGSSOMRÅDEN.....	19
<b>4. RESULTAT</b> .....	<b>21</b>
4.1. DATABEHANDLING.....	21
4.2. FUNKTIONER .....	23
4.2.1. <i>BurnSinks</i> .....	23
4.2.2. <i>Polygonedit</i> .....	23
4.3. AVRINNINGSSOMRÅDEN.....	25
4.4. ....	27
<b>5. DISKUSSION</b> .....	<b>27</b>
5.1. BURNSINKS.....	28
5.1.1. <i>Begränsningar</i> .....	28
5.2. POLYGONEDIT .....	29
5.2.1. <i>Begränsningar</i> .....	29
5.3. ALTERNATIVA METODER.....	31
5.4. VIDARE STUDIER .....	31
<b>6. SLUTSATSER</b> .....	<b>33</b>
<b>7. REFERENSER</b> .....	<b>34</b>
7.1. SKRIFTLIGA REFERENSER .....	34
7.2. MUNTliga REFERENSER.....	36
<b>APPENDIX</b> .....	<b>37</b>

## 1. INLEDNING

En ökande befolkning och växande städer leder till att allt mer naturmark ersätts med hårdgjorda ytor som vägar och hustak. Det medför att större mängder dagvatten genereras och behöver tas om hand (Jacobson, 2011). Vattenflöden i stadsmiljö styrs av delvis andra faktorer än i naturlig miljö (Shuster m.fl., 2005); ledningar används ofta för att leda undan vattnet. Det är önskvärt att undersöka ledningssystemens kapacitet för att ta reda på hur väl systemen klarar av att ta hand om vattnet, exempelvis vid extrema regn. Mer hårdgjorda ytor leder till mer dagvatten och större krav på dagvattenledningarna. Modelleringar av dagvattensystemen ger ofta god vägledning för att bedöma om ledningsnätets kapacitet är tillräcklig. En avgörande faktor för att ta reda på hur mycket vatten som kommer att rinna till en viss ledning är ledningens avrinningsområde, det vill säga vilka delar av omgivningen som bidrar med vatten till ledningen.

Bestämning av avrinningsområden i naturmark görs vanligen utifrån topografin. Det bygger på antagandet att vattendelaren, den gräns som delar två avrinningsområden, sammanfaller med höjdpunkter och ryggar i terrängen. I Sverige är det ett vanligt antagande (Region Gotland, 2012; Olofsson, 2012). Men för tätbebyggda områden gäller inte detta antagande längre. Dagvattenledningar och andra sätt att ta hand om och leda bort regnvatten finns nästan alltid installerat. Strömningsvägen behöver inte nödvändigtvis följa topografin eller gravitationen, varför kunskap om topografin inte kan utnyttjas på samma sätt vid beräkning av avrinningsområden.

För att kunna genomföra en meningsfull simulering av dagvattennätet krävs information om hur området ser ut. Till exempel behövs höjddata, information om markanvändning samt ledningsnätets sträckning, lutning och dimensionering. Med den informationen som grund kan avrinningsområden bestämmas utifrån ledningsnätet med hänsyn till topografin. Tillsammans med nederbördsdata och information om mängden hårdgjorda ytor kan sedan storleken på vattenflödet beräknas för ett visst område eller till en viss dagvattenledning. Utifrån det förväntade vattenflödet kan lämpliga ledningsdimensioner beräknas. Även behovet av eventuella fördröjningsmagasin och andra fasta installationer kan utvärderas. Mer detaljerad beskrivning av avrinningsområden ger högre noggrannhet och säkrare bedömning av statusen för en specifik ledning eller brunn. Upplösning och detaljgrad i tillgängliga data om topografin och ledningsnätet sätter gränsen för hur små och välavgränsade avrinningsområden som är möjligt och meningsfullt att bestämma. Schilling (1991) bedömer att för modellering av avrinning i urbana områden krävs data med betydligt högre tids- och rumslig upplösning jämfört med naturmark. Det beror främst på att mer hårdgjorda ytor gör att vattnet rinner fortare, vilket också resulterar i högre toppflöden (Shuster m.fl., 2005). För ett urbant område som är i storleksordningen av 100 ha är det lämpligt med en nederbördsmätning per 3 minuter och för varje 2 km<sup>2</sup> (Berne m.fl., 2004).

Ett antal tillgängliga applikationer som hanterar digital kartinformation, GIS, har inbyggda funktioner för att beräkna avrinningsområden. De utnyttjar i stor utsträckning höjddata för att avgränsa avrinningsområdena och är främst utvecklade för användning på naturmark. Därför tas ingen hänsyn till ett eventuellt ledningsnät. Om inte justeringar

görs riskerar det att leda till felaktiga bedömningar av hur mycket vatten som kommer till en viss ledning och i förlängningen finns risk för att antingen onödigt stora eller riskabelt små ledningar att installeras.

ArcGIS är en programserie som har utvecklats av amerikanska ESRI (Environmental systems research institute) för hantering av digital kartinformation. Till programmen finns en väl utbyggd samling funktioner för modifiering, strukturering och visualisering av data. En uppsättning funktioner som kan användas för beräkning av avrinningsområden finns i tilläggsprogrammet *ArcHydro Tools* (ESRI, 2011). Funktionerna har jämförelsevis stor möjlighet att anpassas efter användarens önskemål och krav vid de olika stegen för att beräkna avrinningsområden. Processen är dock inte utvecklad för att bestämma avrinningsområden i urban miljö.

Hela program som är baserade på ArcGIS har utvecklats. Ett sådant program är Mike Urban, som har utvecklats av DHI (Dansk Hydrologisk Institut) och används till att modellera ledningsnät av olika typ. Programmet har en halvautomatisk funktion för att ta fram avrinningsområden till specifika brunnar, men metoden tar ingen hänsyn till terrängen och resultatet blir därför svåränvändbart utan modifieringar (Hammarlund, pers. medd.). Det finns andra program i Mike-serien som kan användas för att beräkna (topografiska) avrinningsområden för nätverk av vattendrag i naturmiljö. Det är också möjligt att rita avrinningsområden manuellt och att efterbehandla dem i olika grad.

### **1.1. SYFTE**

Syftet med det här examensarbetet var att utveckla en metod för att generera avrinningsområden i områden med dagvattensystem. Funktioner för att beräkna avrinningsområden i naturliga miljöer finns redan, men de bygger primärt på topografin och möjligheterna att ta hänsyn till ett eventuellt dagvattennät saknas. Målet med arbetet var att ta fram en ny metod för beräkning av avrinningsområden, som delvis bygger på befintliga funktioner. I funktionen används primärt dagvattennätet för att beräkna fram vattendelarna, det vill säga åt vilket håll vattnet rinner i olika delar av området. De resulterande avrinningsområdena bör vara anpassade för modellering av dagvattennätet i Mike Urban. Mer specifikt:

- Hur kan framtagning av avrinningsområden göras med dagvattennätet som utgångspunkt?

Som en del i detta undersöktes också:

- Hur kan en funktion för korrigerande av avrinningsområden baserat på deras storlek konstrueras och implementeras i ArcMap, där områden som bedöms som för små för att bidra till informationen läggs samman med intilliggande områden och för stora områden delas upp till mindre?
- Hur kan hänsyn tas till punkter i ledningsnätet varifrån vattnet inte fortsätter rinna naturligt, till exempel pumpstationer?

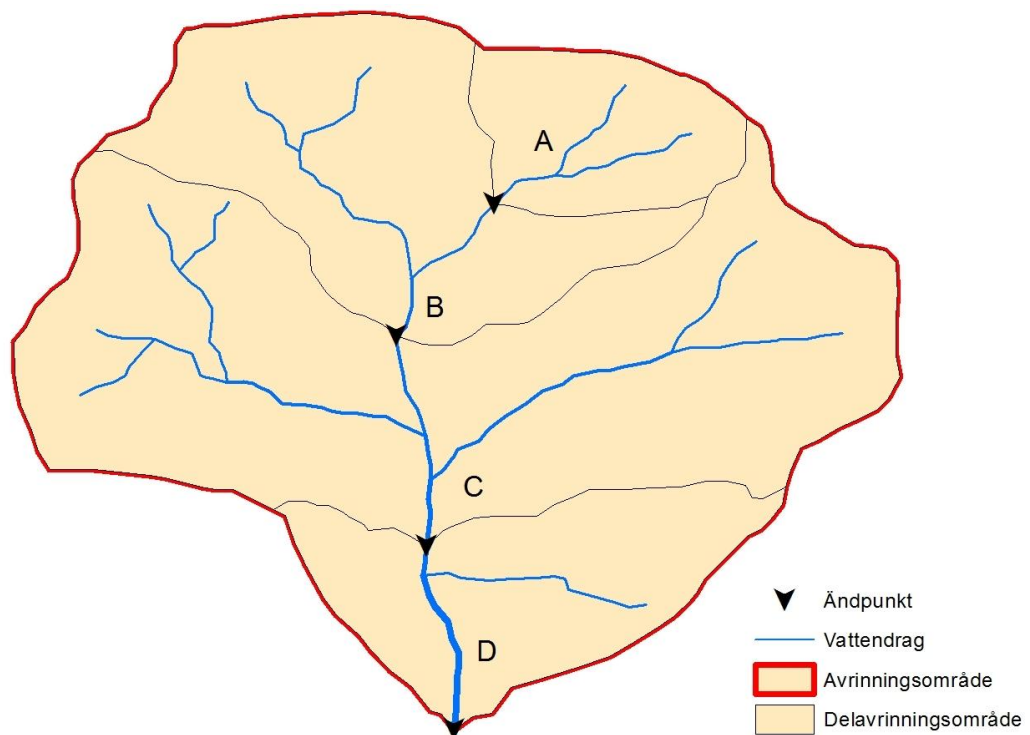
## 2. BAKGRUND

### 2.1. DAGVATTEN OCH INFILTRATION

#### 2.1.1. Avrinning

All nederbörd som faller över ett visst område kommer att på ett eller annat sätt transporteras bort därifrån. Det sker lokalt genom ett par olika processer. En del vatten avdunstar, men den största delen infiltrerar ner i marken. Där bidrar det till grundvattenbildningen, med undantag för det vatten som växtlighet tar upp. Mer ovanligt i vårt klimat, men förekommande, är att vattnet rinner på markytan i naturlig miljö (Grip och Rodhe, 1994).

Det vatten som infiltrerar rinner sedan vidare genom marken för att slutligen nå ett vattendrag och fortsätta mot havet. För varje punkt i terrängen kan man bestämma till vilket vattendrag som vattnet som hamnar där kommer att rinna till, utifrån topografi och geologi. Det kan också finnas lågpunkter i terrängen, områden varifrån vattnet inte kan rinna vidare (Grip och Rodhe, 1994). Genom att utvidga detta koncept kan man göra avrinningsområden, avgränsade ytor där all nederbörd förr eller senare kommer att ledas till ett visst vattendrag (Figur 1). Avrinningsområdena avgränsas av vattendelare. En vanlig förenkling är att ytvattendelaren sammanfaller med grundvattendelaren för regionalt grundvatten, eftersom grundvattenytan oftast är nära markytan vid svenska förhållanden (Grip och Rodhe, 1994). Därför används topografin för att approximera grundvattendelaren, vilken annars kräver utförliga mätningar för att lokalisera. Avrinningsområdenas storlek är en grundläggande faktor vid beräkningar av vattenflöden till de olika utloppspunkterna (Figur 1) (Grip och Rodhe, 1994).



**Figur 1.** Schematisk bild över ett avrinningsområde med delavrinningsområden A-D och ett grenat vattendrag som löper genom landskapet. Varje delavrinningsområde avslutas nedströms i en utloppspunkt ur området.

### 2.1.2. Dimensionering av ledningsnät

Vatten som avrinner från bland annat hårdgjorda ytor, dagvatten, leds ofta till dagvattenledningar när sådana finns, men diken och andra öppna lösningar börjar bli mer vanliga. Det ställs krav på hur mycket regnvatten en dagvattenledning (eller annat system för avledning av dagvatten) ska klara av att avleda, med målet att minimera risken för dämning. Ett mått som används är hur sannolikt det är att ledningen kommer att bli fylld, vilket baseras på hur ofta (statistiskt) ett regn med en viss intensitet och varaktighet inträffar, regnets så kallade återkomsttid (Svenskt Vatten, 2004). Hur mycket av regnvattnet som avrinner på markytan och när ledningen kan beskrivas med avrinningskoefficienten, som är ett tal mellan 0 och 1. Avrinningskoefficienten beror av avrinningsområdets sammansättning och anger hur stor del av nederbörden i området som bidrar till avrinningen. Det påverkas av andelen hårdgjorda ytor i området, som är kopplat till infiltrationen i marken och hur mycket vatten som når ledningen. Även områdets lutning och regnintensiteten spelar in.

Vattenvolymer som når ledningen kan beräknas med olika detaljgrad. Rationella metoden är den enklaste varianten. Den förutsätter att avrinningsområdet är i stort sett rektangulärt och homogent. Metoden tar inte någon hänsyn till olika rinntider eller eventuell skillnad i regnintensitet över området, varför den främst är lämplig för mindre områden (Svenskt Vatten, 2004).

Det dimensionerande flödet,  $Q_{\text{dim}}$ , beräknas enligt (1).

$$Q_{\text{dim}} = A \cdot \varphi \cdot i(t_r) \quad (1)$$

$Q_{\text{dim}}$  – dimensionerande flöde [ $l/s$ ]

$A$  – avrinningsområdets area [ $ha$ ]

$\varphi$  – avrinningskoefficient [–]

$i(t_r)$  – regnintensitet [ $\frac{l}{s \cdot ha}$ ]

$t_r$  – regnets varaktighet [ $s$ ]

För  $t_r$  används samma tid som rinntiden genom området, vid den tidpunkten bidrar alla delar av avrinningsområdet till flödet i utloppspunkten (Svenskt Vatten, 2004). Regnintensiteten  $i(t_r)$  i (1) beräknas enligt Dahlströms formel (2), där konstanterna har bestämts empiriskt (Dahlström, 2010).

$$i(t_r) = 190 \cdot \sqrt[3]{\dot{A}} \cdot \frac{\ln(t_r)}{t_r^{0,98}} + 2 \quad (2)$$

$\dot{A}$  – regnets återkomsttid [ $mån$ ]

Vilken regnintensitet och återkomsttid som används för att dimensionera ledningarna är olika för olika typer av områden. Områden med stadsbebyggelse har generellt högre ställda krav än obebyggd mark. Det beräknade dimensionerande flödet kan sedan utnyttjas för att bestämma lämplig ledningsdimension (Svenskt Vatten, 2004).

Det finns en lite mer komplicerad metod, tid-areametoden, för att beräkna det dimensionerande flödet, där avrinningsområdet delas upp i olika delar baserat på rinntid och markanvändning. Vattenmassorna "leds" i beräkningen nedåt mot utloppspunkten. Det dimensionerande flödet erhålls vid den tidpunkt som det största vattenflödet beräknas vid utloppspunkten. Då antas alla delområden bidra till flödet. Även här används regn med olika återkomsttid och avrinningskoefficienter som beror på områdets sammansättning (Ahlquist-Juhlén, pers. medd.).

## **2.2. DATAHANTERING**

Geografiska data är ofta utrymmeskrävande. Förutom en referens till positionen (vanligen koordinater i ett specificerat koordinatsystem) är det önskvärt att kunna koppla olika attribut till punkten. Det kan till exempel vara jordtyp, markanvändning, halt av olika föroreningar i marken, djup till grundvattenytan och så vidare. Det finns två huvudsakliga och dominerande sätt att lagra, hantera och visualisera informationen på, vektor- respektive rasterformat. De har förhållandevis varierande användningsområden och egenskaper (Harrie, 2008). Det är fullt möjligt att konvertera mellan typerna, dock med förluster i detaljgrad. Det är också både möjligt och lämpligt att kombinera dem i samma kartbild. Ibland används databaser för att lagra filer tillsammans på ett strukturerat och enhetligt sätt. Det underlättar utnyttjandet av filerna (Harrie, 2008).

### **2.2.1. Objektmodellen**

Objektmodellen bygger på att olika objekt representeras som olika objekttyper. Skog, stigar, vägar, hus och så vidare tillhör var sin objekttyp. Med en tvådimensionell referensram kan dessa objekt representeras med punkter, linjer och ytor (polygoner) (Figur 2). Punkter används för objekt där utbredningen i rummet inte är av betydelse i den aktuella skalan, enbart positionen är relevant (Harrie, 2008). Exempel på detta är mätpunkter och mätstationer, byggnader och brunnar. Linjer är lämpligt att använda för objekt som bara har (väsentlig) utbredning i en dimension, så som bäckar, ledningar och vägar. Polygoner fungerar för homogena områden med väldefinierade gränser; sjöar, åkrar, parkeringsplatser eller administrativa indelningar (Harrie, 2008).

Geografisk information som baseras på objektmodellen kallas för vektordata. I vektordata lagras objekt som ordnade koordinatpar (Harrie, 2008). Fördelarna med detta är bland annat att det är lätt att knyta olika data, attribut, till objekten. Ett objekt definieras av de koordinater som ändpunkter eller hörn har. Objektet varken deformeras eller tappas detaljgrad av att till exempel ändra skala. Formatets jämförelsevis begränsade filstorlek medför också att det effektivt går att rita upp informationen på skärmen (Harrie, 2008).

Amerikanska ESRI (se avsnitt 2.3) har utvecklat en mycket spridd filtyp för vektorformat; shapefilen. Den finns i tre olika varianter beroende datatyp: punkt, linje eller polygon. Formatet stöds av så gott som alla GIS-hanterande applikationer. Det som till vardags kallas för shapefil består egentligen av ett flertal (minst tre) olika filer.

### 2.2.2. Fältmodellen

Ett alternativ till att lagra data kopplat till en viss koordinat är att använda fältmodellen. Det är lämpligt för kontinuerliga, varierande parametrar, där varje punkt i studieområdet har ett värde. Marktäckning, temperatur och markhöjd är exempel på sådana parametrar (Harrie, 2008).

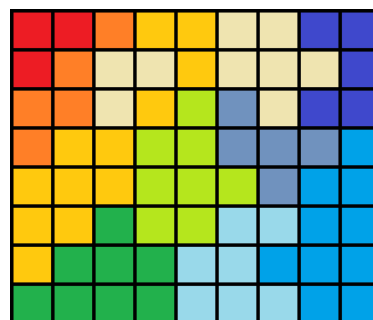
Representationer enligt fältmodellen, raster, täcker hela ytan och består av en mängd rutor, celler (Figur 3). Varje cell kan vanligtvis bara ha ett värde kopplat till sig (Harrie, 2008). Upplösningen styrs av storleken på cellerna och kan variera stort beroende på vad rastret ska användas till och tillgänglig indata. För årsmedeltemperaturer över jordklotet kan det vara lämpligt med rasterceller i storleksordningen 100 meter till flera kilometer medan det för höjddata till hydrologisk modellering ofta används celler med 2 till 50 meter långa sidor. När cellerna ökar i storlek får varje cell representera en större yta. Därmed kommer information om lokala variationer att gå förlorad i kartbilden men i gengäld krävs inte lika mycket lagringsutrymme.

Raster kan genereras från vektordata, vanligen punkter och/eller linjer. Eftersom det (på någon skala) finns utrymmen mellan punkterna och linjerna är det inte relevant att direktöversätta dessa vektordata till ett raster rakt av. För att undvika att det uppstår hål i rastret används istället olika metoder för att uppskatta vilket värde det "borde vara" i utrymmena däremellan. Se avsnitt 2.3.1 för en kort beskrivning av här relevanta interpolationsmetoder och interpolerande funktioner i tillgänglig programvara. För lagring av raster finns en mängd olika filtyper som fungerar på olika sätt.

Det är fullt möjligt att representera exempelvis marktäckning med polygoner i vektorformat, men diskretiseringsproblematiken (övergång från kontinuerliga till diskreta data) och noggrannheten vid skapandet kan resultera i att hål uppstår mellan polygonerna. Det innebär att vissa ytor inte har någon (tilldelad) marktäckning och vidare användning av data kan kompliceras. Med representation enligt fältmodellen undviks det problemet, eftersom det finns celler överallt och varje cell har tilldelats ett värde. I gengäld gör rastrets cellstruktur att gränser bara kan dras där en cell tar slut, vilket leder till begränsningar i detaljgraden (Harrie, 2008). Att representera enskilda objekt blir därför opraktiskt med fältmodellen. Det finns inget annat sätt än att tilldela en hel cell ett värde som representerar objektet, varför alla objekt blir minst en cell stora (Harrie, 2008).



**Figur 2.** Exempel på geometrierna punkt, linje och polygon.



**Figur 3.** Exempel på raster. Varje ruta representerar en rastercell.

### 2.2.3. Referenssystem

Placeringen av objekt anges relativt kända referenspositioner. Därför krävs ett koordinatsystem med minst en fixerad, väldefinierad referenspunkt. Koordinatsystemet kan då kallas för ett referenssystem. Tvådimensionella referenssystem är enkla att förstå och använda, men eftersom jordytan är buktande så innebär det vissa problem. För högre noggrannhet är det lämpligt att använda ett tredimensionellt, elliptiskt koordinatsystem som tar hänsyn till jordytans krökning, men det sker på bekostnad av krångligare beräkningar (Harrie, 2008). Den mest använda ellipsoiden för att representera jordytan kallas GRS 80 (Lantmäteriet, 2012), och är ett resultat av satellitmätningar. En annan ellipsoid som används av bland annat GPS-system kallas för WGS 84, den sammanfaller till 1 mm när med GRS 80. Horisontellt läge på jordellipsoiden anges med longitud och latitud, vilket anges som vinklar utgående från en nollmeridian genom Greenwich (longitud) respektive ekvatorn (latitud). Den tredje dimensionen anges som höjden vinkelrätt över ellipsoidens yta (Harrie, 2008).

SWEREF 99 är det referenssystem som används i Sverige sedan 2007. Det bygger på GRS 80 och det europeiska referenssystemet ETRS 89, vilka är baserade på projektionen *Transversal Mercator*. Det är uppbyggt kring en medelmeridian i Sverige och därefter indelat i tolv zoner (Harrie, 2008). Dessa används på kommunal nivå för att minimera de lokala avbildningsfelen. Tidigare har RT 90 använts som nationellt referenssystem. Det bygger dock på en annan elliptisk modell av jordklotet och skillnaden kan uppgå till flera hundra meter. Därför är det avgörande att veta vilket referenssystem en viss datamängd tillhör (Harrie, 2008).

I Sverige används tre olika höjdsystem; RH 00, RH 70 samt RH 2000, vilket är det senast införda systemet. Höjderna i RH 70 och RH 2000 refereras till uppmätta data över havets medelnivå i Amsterdam, vilket även höjdsystemen i våra grannländer gör. Systemet RH 00 har istället medelvattenytan i Stockholm år 1900 som referensnivå (SMHI, 2010).

### 2.3. PROGRAMVARA

Det finns en mängd olika program för att hantera digital kartinformation, både med öppen källkod och patentskyddade program. ArcGIS är en proprietär programserie för hantering av geografisk data som tagits fram av amerikanska ESRI (Environmental Systems Research Institute). ArcMap och ArcCatalog är två program som har använts här, den senare är en filhanterare medan den förra används för visualisering, behandling och presentation av kartdata och samband dem emellan. ArcMap innehåller dessutom många funktioner för att modifiera data, som i så gott som samtliga fall dessutom är möjliga att köra från en kommandorad med ett Pythonkommando (avsnitt 2.3.3). Detta kan utnyttjas för att skapa funktioner som modifierar data och avrinningsområden framtagna med ett befintligt system, till exempel ArcHydro Tools.

Mike Urban är ett program som har utvecklats av DHI för att modellera ledningsnät av alla typer och som baseras på ArcMap. Programmet används i stor utsträckning av kommuner i Sverige för att lagra data om och undersöka kapaciteten hos installerade ledningssystem (Hammarlund, pers. medd.).



Därutöver finns ett flertal (gratis) program med öppen källkod, varibland Quantum GIS (QGIS), GRASS och SAGA GIS kan nämnas. Dessa behandlas inte vidare i den här rapporten.

### 2.3.1. Funktioner i ArcMap

Nedan presenterade verktyg och funktioner är ett urval av dem som har använts under projektet och varit av vikt för att nå resultatet. Undantaget är avsnittet om *Hydrology Toolset*, som använts sparsamt eftersom *ArcHydro Tools* (avsnitt 2.3.2) är mer omfattande. De som inleds med en asterisk (\*) anropades från de under projektet skapade Pythonfunktionerna. All information i detta stycke kommer från ArcGIS Resource Center Online Desktop Help 10.0 (ESRI, 2012b).

*Editing* (Redigering) är egentligen inte en funktion, utan snarare en uppsättning verktyg som grafiskt kan användas för att ändra och lägga till information i kartmaterialet. Det är därför ofta använt vid till exempel (manuell) digitalisering av papperskartor och ortofoton. Det är möjligt att skapa och ändra punkter, linjer och polygoner i befintliga och nya filer. Ett antal kommandon finns för att underlätta arbetet, så som att flytta hörn, spåra befintliga linjer och rita olika former. Det är också möjligt att ändra attributvärden när redigering har aktiverats.

\**Append* (Lägg till) tar en fil (fungerar med både raster och shapefiler) och lägger till alla (eller specificerade) geometrier till en annan fil. Det går även att kopiera över attribut, förutsatt att tabellhuvudena stämmer överens.

\**Dissolve* (Sammanmälta) används för att foga samman alla geometrier i en fil eller baserat på något (statistiskt) villkor. Här har funktionen utnyttjats för att lägga ihop två angränsande polygoner till en större, men andra användningsområden finns.

*Erase* (Klipp bort) tar bort de delar i en fil som grafiskt överlappar med en annan, och kan på så vis skapa hål i exempelvis raster eller polygoner, eller ta bort mätpunkter i ett visst område. Funktionen fungerar med både raster- och vektordata samt i kombination.

*Extract by mask* (Extrahera med mall) tar bort alla objekt eller celler i en fil som ligger utanför utsträckningen av en annan fil. Det kan minska datamängden genom att ta bort för tillfället ointressanta data så att beräkningstiden kan hållas nere.

*Int* (Heltal) avrundar alla cellvärden i ett raster nedåt till närmaste heltal. Det är ett krav för vissa funktioner att cellvärdena är lagrade som heltal.

*Intersect* (Skärning) undersöker vilka delar av indatafilerna som överlappar och skriver endast de överlappande delarna till utdatafilen. Funktionen kan användas med både raster- och vektordata. Funktionen är lämplig att använda för att minska ner datamängden så att beräkningstiden kan optimeras, eller för att klippa bort data som ligger utanför det studerade området.

\**Merge* (Sammanfoga) kombinerar flera filer av samma typ (punkt, linje eller polygon) till en fil. Ingen förändring i vilka geometrier som finns eller hur de ser ut sker, de läggs bara ihop till en gemensam fil.

*Minus* (Subtraktion) subtraherar värdet i cellerna i ett raster från motsvarande celler i ett annat raster.

\**Point to raster* (Punkt till raster) konverterar vektordata, punkter, till ett raster. Ingen interpolation sker, ett attribut från varje punkt översätts till ett värde i en rastercell. Om flera punkter hamnar inom samma cell är det möjligt att specificera vilket värde som prioriteras.

*Raster to polygon* (Raster till polygon) konverterar rasterdata till polygoner baserat på cellvärdena. Den här funktionen kräver att alla cellvärden är heltal.

*Topo to raster* (Topografi till raster) interpolerar fram en hydrologiskt korrekt höjdmodell (raster) från höjddata i punktform och/eller höjdkurvor. Funktionen interpolerar mellan indata med avståndsviktad medelvärdesinterpolation (IDW) (se nedan) och tar särskild hänsyn till att vattendelare och vattenvägar är bevarade från indata samt att det finns ett sammanbundet strömnätverk i höjdmodellen.

*Hydrology toolset* (Hydrologiverktyg) är en samling i ArcMap inbyggda funktioner som finns i *Spatial Analyst toolbox*. De kan användas för att modellera vattenflöden och rinnvägar. Verktygen utgör ett alternativ till nedan presenterade *ArcHydro Tools*, om än inte lika omfattande.

### **Avståndsbaserad interpolation**

Interpolation används för att sträcka ut mätdata i punktform eller linjer över en yta och skapa en kontinuerlig modell som tydligare visualiserar förändringar i rummet. Det finns en mängd olika beräkningsmetoder och algoritmer för att anpassa och tilldela värden mellan datapunkterna.

*Inverse distance weighting (IDW)* (Avståndsviktad medelvärdesinterpolation) är en metod för lokal interpolation från punktdata. Metoden bygger på antagandet att det är mer sannolikt att närliggande punkter än de långt bort är relaterade till det interpolerade värdet (Naoum och Tsanis, 2004). Värdet för en rastercell beräknas utifrån ett antal punkter i närområdet, som också viktas mot hur nära cellen de ligger. Punkter som ligger nära får en större inverkan på cellvärdet än de som är längre bort. Det är ibland lite problematiskt att definiera kriterier för vilka punkter som tas med när det viktade medelvärdet beräknas. Ofta används en viss sökradie runt cellen och alla punkter som faller innanför utnyttjas. Det innebär att antalet punkter kan variera i olika delar av området, ibland kanske punkter saknas helt. En för stor sökradie kan däremot resultera i att många celler istället får medelvärdet av (i stort sett) alla punkter, vilket blir alltför homogent för att kunna anses ge någon värdefull information (Harrie, 2008). En annan variant är att istället ta ett visst antal punkter närmast cellen. Vid en skev fördelning av data kan antalet punkter i en viss riktning från cellen få större tonvikt och därmed förskjuta värdet (Harrie, 2008). Det finns ytterligare parametrar och aspekter att ta hänsyn vid användning av den här interpolationsmetoden, samt föreslagna lösningar till dessa, men de diskuteras inte vidare här.

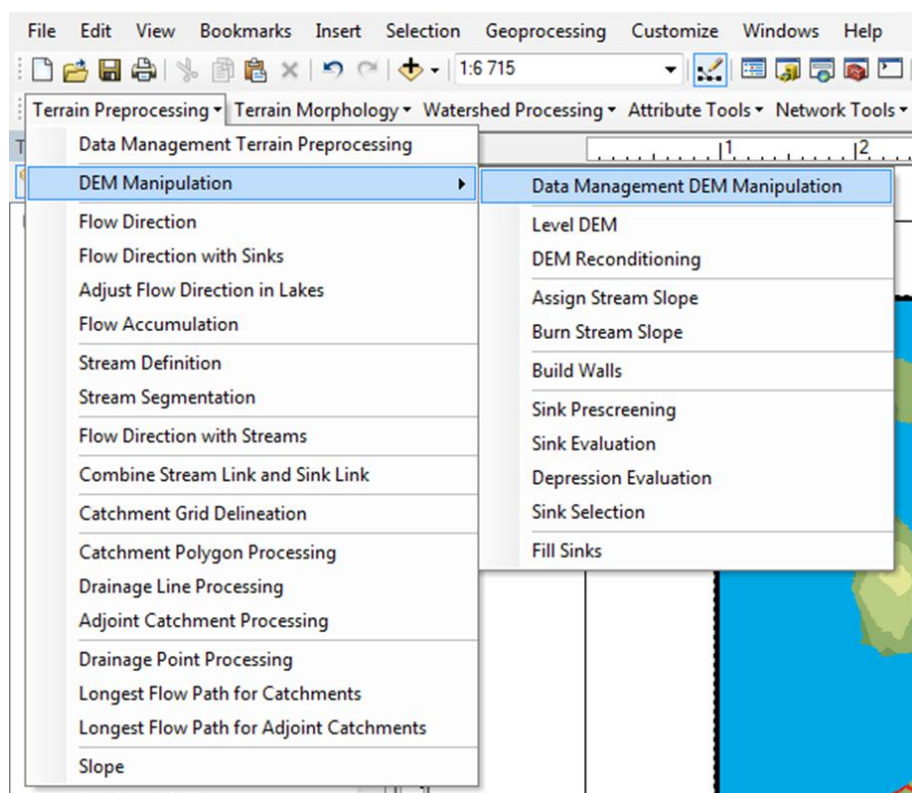
*Kriging* är en regionaliserad interpolationsmetod som bygger på ett semivariogram (Naoum och Tsanis, 2004). Semivariogrammet är ett statistiskt mått som, lite förenklat,

tas fram baserat på variansen mellan par av mätvärden och hur långt ifrån varandra de ligger. Istället för att vikta mätvärdena mot avståndet viktas de mot variogrammet för använda data (Harrie, 2008).

*Thiessen-polygoner* är en kvalitativ interpolation, den tar enbart hänsyn till den närmaste mätpunkten (Naoum och Tsanis, 2004). Polygoner skapas runt mätpunkterna som innefattar all yta i omgivningen vars närmaste mätpunkt är den aktuella mätpunkten. Det ger homogena delar, där varje delytta tilldelas det värde som mätpunkten har (Harrie, 2008). Metoden har inte använts i den här studien, men är mycket vanlig vid hantering av nederbördsdata (Naoum och Tsanis, 2004), vilka är väsentliga vid modellering av dagvattenflöden.

### 2.3.2. ArcHydro Tools

*ArcHydro Tools* (AHT) är en samling funktioner som tagits fram av ESRI (Figur 4) och som baseras på deras *ArcGIS Hydro Data Model*. AHT version 2.0 kan installeras som en utvidgning till ArcMap 10.0. Verktøyen innefattar bland annat behandling av höjdmodeller för att kunna ta fram avrinningsområden och flödesvägar, modifiering och anpassning av avrinningsområdena när de skapats och en del ytterligare efterbehandling. Om inte annat anges är de beskrivna funktionerna från gruppen *Terrain preprocessing tools* (Verktøy för förbehandling av topografin) och presenteras i den ordning som de har använts. All information i detta avsnitt har hämtats från Arc Hydro GP Tools 2.0 Tutorial (ESRI, 2011).



Figur 4. Översikt över de olika grupperna av funktioner i ArcHydro tools.

*DEM Reconditioning* (Anpassning av höjdmodellen) sänker ner önskade, fördefinierade vattenvägar i höjdmodellen. Värdet i cellerna i höjdmodellen som sammanfaller med dessa linjer minskas med ett visst värde. Det gör att allt vatten som rinner fram till vattenvägen fortsätter att rinna i den och leds inte åt ett annat håll, till exempel på grund av hur topografin ursprungligen såg ut. Tre parametrar kan specificeras för att anpassa processen. *Stream buffer* (buffertområde) anger hur många celler i bredd runt vattendraget som ska sänkas för att få en gradvis ökande lutning ner mot vattendraget. *Smooth drop/raise* (mjuk nedsänkning/höjning) anger, i höjdmodellens höjdenhet, hur djupt ned buffertområdet ska sänkas, medan *Sharp drop/raise* (skarp nedsänkning/höjning) bestämmer hur djupt de celler som ligger rakt under vattenvägen sänks. Funktionen kan även användas till att markera barriärer i terrängen såsom vägbankar och liknande genom att höja upp dessa. Lämpliga parametervärden beror på hur resultatet ska användas samt hur höjdmodellen ser ut, till exempel hur branta partier det finns samt skillnaden mellan högsta och lägsta höjdvärde.

*Sink prescreening* (Förundersökning av sänkor) identifierar var det finns sänkor i höjdmodellen. De ”fylls” genom att ändra cellvärdet till samma som omgivande celler så att vattnet kan rinna förbi punkten, men bara om avrinningsområdet till sänkan är mindre än ett angivet värde. Denna process körs främst för att minska beräkningstiden för efterföljande funktioner.

*Sink evaluation* (Utvärdering av sänkor) skapar polygoner för området där sänkorna finns och beräknar deras tillrinningsområden.

*Sink selection* (Urval av sänkor) väljer ut de sänkor som inte ska fyllas igen i efterföljande steg och utgår från de polygoner som skapades i *Sink evaluation*. Dessa kan specificeras med hjälp av upp till fyra kriterier, där ett minimumvärde anges för att polygonen ska räknas som en sänka och inte senare fyllas igen. Kriterierna som kan specificeras är minsta djup, minsta area, minsta volym och minsta tillrinningsområde. För de sänkor som inte ska fyllas ändras ett värde i attributtabeln.

*Fill sinks* (Fyll sänkor) fyller igen sänkor i höjdmodellen genom att ändra cellvärdet till att motsvara omgivande cellers värden. Vattnet tillåts då att rinna förbi området. Det är dock inte önskvärt att fylla igen alla sänkor, varför funktionen tar hänsyn till värdet i attributtabeln som justerades i *Sink selection*. Den resulterande höjdmodellen har inte några sänkor bortsett från där de har specificerats att vara kvar. Vatten från alla celler kommer därmed att kunna rinna till kanten av höjdmodellen.

*Flow direction with sinks* (Flödesriktningar med sänkor) tar för varje cell fram åt vilket håll vatten som faller i cellen kommer att rinna. De sänkor som tidigare har definierats tillåts vara ändpunkter för flödet, liksom ytterkanterna på höjdmodellen. Vattnet antas bara rinna i en riktning, mot den av de omgivande cellerna med det lägsta värdet. Funktionen finns även utan tillägget om sänkor och fungerar på motsvarande sätt.

*Flow accumulation* (Ackumulerat flöde) räknar för varje cell ut antalet celler uppströms vars vatten rinner till den aktuella cellen. Detta värde lagras i cellen i ett nytt raster.

*Stream definition* (Definiera strömmar) definierar utifrån det ackumulerade flödet var det förväntas rinna tillräckligt mycket vatten för att det ska räknas som en vattenväg. Definitionen utgår från antalet uppströms celler (Figur 5), vilket kan direktöversättas till tillrinningsområdets area. Det är i det här steget som antalet avrinningsområden indirekt bestäms.

*Stream segmentation* (Uppdelning av strömmar) skapar ett raster med strömsegment utifrån resultatet i *Stream definition*. Vattenvägarna delas upp i de punkter där de går ihop eller isär (Figur 6). Varje segment ges ett unikt identifikationsnummer som är spårbart genom den fortsatta processen.

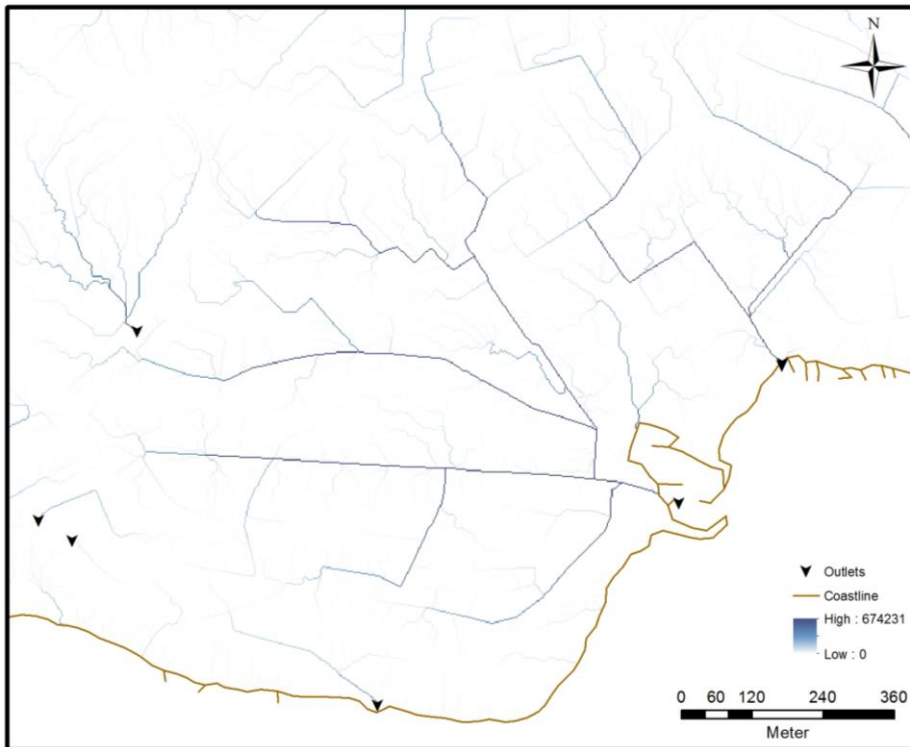
*Combine stream link and sink link* (Kombinera strömmar och sänkor) skapar ett raster där både sänkor och nu definierade strömmar ingår. Detta används som indata till efterföljande funktioner.

*Catchment grid delineation* (Avgränsning av avrinningsområden) tilldelar varje cell ett värde baserat på till vilket vattendrag vattnet från cellen rinner. Värdet är baserat på de unika identifikationsnumren som tilldelades strömsegmenten i *Stream segmentation*.

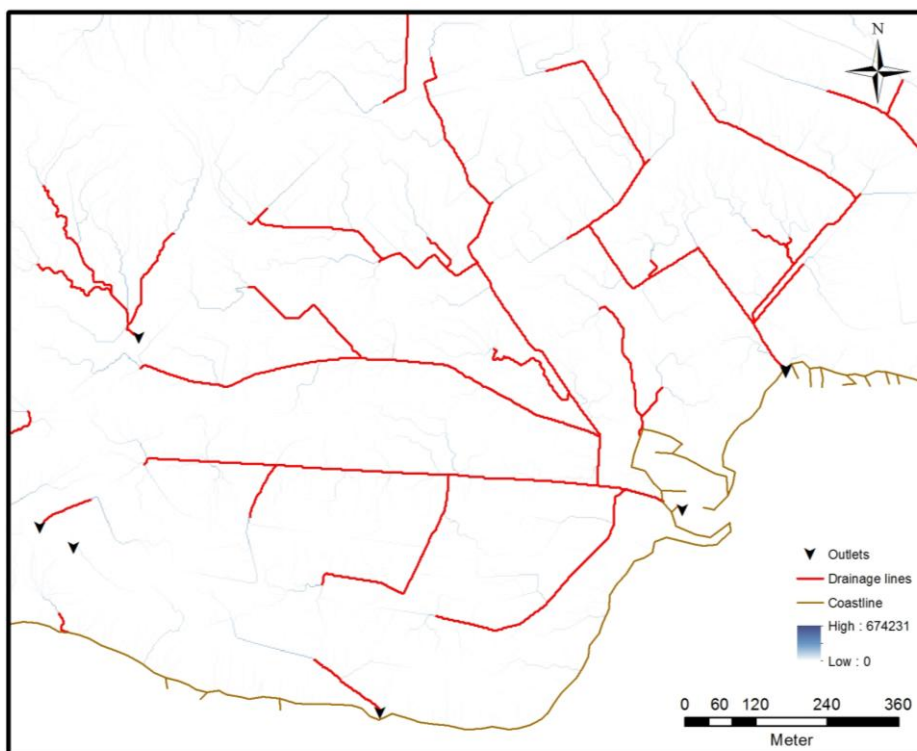
*Catchment polygon processing* (Bearbetning av avrinningsområden) utgår från cellvärdena som beräknades i föregående steg och konverterar varje unikt område till en polygon, vilket motsvarar avrinningsområdet.

*Drainage line processing* (Bearbetning av vattenvägarna) konverterar strömsegmenten som skapades i *Stream segmentation* till vektoriserade linjer. Identifikationsnumren bevaras.

*Find next downstream line* (Identifiera nedströms flöde) lägger till ett attributfält till shapefilen som skapades i *Drainage line processing* där det står identifikationsnumret till det strömsegment som följer närmast nedströms. Den här funktionen tillhör gruppen *Attribute tools* (Attributverktyg).



**Figur 5.** Ackumulerade vattenflöden i terrängen, med ledningsnätet representerat som vattendrag. Ju större yta som avvattnas genom en specifik cell desto högre värde ges den, det vill säga den blir mörkare blå.



**Figur 6.** Definierade vattenvägar (drainage lines) utifrån ackumulerade flöden från Figur 5. De celler som avvattnar minst en viss area definieras som en vattenväg. Dessa delas sedan upp i segment där de går ihop, avrinningsområden beräknas slutligen för varje sådant segment.

### 2.3.3. Geografisk databehandling med Python

Python är ett programmeringsspråk med öppen källkod som kan användas till en mängd olika applikationer (Python.org, 2013). Det finns en begränsad mängd inbyggda funktioner men desto fler importerbara paket med funktioner, så kallade moduler (*modules*). När matematiska funktioner behövs, som till exempel kvadratrötter, trigonometriska funktioner och logaritmer, kan modulen `math` importeras. På samma sätt, om vektor- och matrisoperationer önskas, kan modulen `numpy` användas. Nämnade moduler kommer vanligtvis med vid en standardinstallation av Python, men det finns också ett stort antal moduler som olika organisationer och användare har skapat, ofta för att fylla ett specifikt behov. Dessa behöver laddas ner och installeras separat från respektive källa. Ett exempel på en sådan fristående modul är Geospatial Data Abstraction Library (GDAL) som används för programmatisk hantering av raster- och vektordata. Modulen har givits ut med öppen källkodslicens av The Open Source Geospatial Foundation (OSGeo) (OSGeo, 2012). Det är dessutom fullt möjligt, och rekommenderat, att skapa en modul för uppgifter som utförs flera gånger och som lätt kan återanvändas, istället för att programmera om avsnittet för varje användning.

ESRI har från version 9 av ArcMap inkorporerat möjligheten att använda Python för att automatisera repetitiva eller krångliga processer. Då introducerades `arcpy`-modulen, varifrån i stort sett alla funktioner i ArcMap kan köras. En större förändring av `arcpy`-modulens funktion och användning och en uppgradering till Python 2.6 gjordes med lanseringen av ArcMap version 10, samtidigt som nya sätt att arbeta lades till (ESRI 2012a). Verktygen i ArcToolbox kan köras genom ett kommando i Python-fönstret, förutsatt att motsvarande licens finns installerad. ArcMap tillåter också körning av skript; filer som innehåller en sekvens av funktioner som lätt kan återanvändas. Dessa skript kan dessutom användas för att skapa ett användardefinierat verktyg, och med de i ArcMap inbyggda hjälpmedlen kan dessa anpassas för att enkelt användas som vilket annat verktyg i ArcToolbox.

Det finns en mängd olika typer av objekt i Python (ej att förväxla med objektmodellen, som är helt orelaterad). Objekten används för att skilja till exempel text från heltal från decimaltal eller komplexa tal (learnpython.org, 2013). De ger en nödvändig distinktion när olika funktioner används som kräver en viss typ av indata. Förutom de redan nämnda objekttyperna kan det vara värt att ta upp *lists* (listor) och *dictonaries* (lexikon). Listor kan närmast jämföras med vektorer, en lång rad värden som kan anropas baserat på positionen i listan (tutorialspoint.com, 2013). Olika sökningar kan också göras över listor, liksom sortering och diverse matematiska operationer. Ett lexikon däremot, har per definition ingen ordning. Istället innehåller lexikonet par av värden, där det första är en nyckel (*key*) och det andra ett värde (*value*). Nyckeln används för att referera motsvarande värde i lexikonet. Eftersom ordning i det här sammanhanget inte är relevant att tala om kan ett lexikon inte sorteras, men i likhet med en lista går det bra att iterera över alla värden. Det här är särskilt användbart när man inte vill ha numeriska nycklar utan exempelvis landsnamn med tillhörande befolkningsmängd eller när man inte har en kontinuerlig nummerserie. Att ta reda på befolkningsmängden i ett visst land kräver då bara att man ”vet” vad landet heter och anropa nyckeln (landsnamnet) som är kopplat till värdet (befolkningsmängden) i lexikonet (tutorialspoint.com, 2013).

Utöver dessa objekttyper finns en hel mängd andra, varav många är specifika för vissa moduler och funktioner. `arcpy` utnyttjar till exempel ett antal objekttyper för att bland annat referera geometrier och vid redigering. För att utföra vissa operationer på objekt kan man anropa en metod (*method*) på objektet. Vilka metoder som går att anropa beror på objekttypen och situationen. För listor finns exempelvis metoder för att sortera (*sort*), lägga till värden på slutet eller på specificerad plats (*append*), vända på listan och så vidare. Det finns också ett större antal metoder som kan anropas på till exempel textsträngar, tal och geometriska objekt.



### 3. MATERIAL OCH METOD

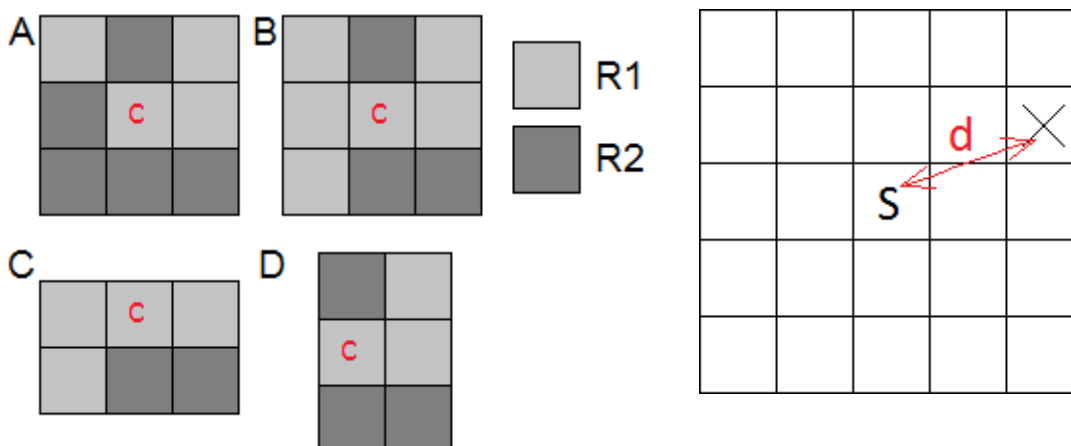
I kapitlet beskrivs vilka data som har använts, höjddata och ledningsnät, och hur dessa data har behandlats. Metod och användning av de funktioner som skapats för att manipulera data beskrivs också, samt hur avrinningsområden beräknades med hjälp av dessa funktioner.

#### 3.1. HÖJDDATA

Lidingö stad har tagit fram höjddata med laserskanning för hela Lidingön vilka har använts som testdata. Ön är ungefär 3000 ha stor (SCB, 2001). Punkterna hade sedan (automatiskt) delats upp beroende på vad som mätts; vattenytor, mark eller "ej mark", det vill säga byggnader, träd och dylikt. Dessa olika klassificeringar finns specificerade i datafilerna som lagrar laserdata, las-filer. Las-punkterna hade även konverterats till rasterformat (nedan kallat `2mx2m_grid`) och efterbehandlats av Lidingö stad i de delar som täcker bebyggda områden. På ytor däremellan hade hänsyn tagits till mätdata genom att interpolera, varför överensstämmelsegraden med verkligheten där kan antas vara lägre.

#### 3.2. HÖJDMODELLEN (DEM)

De av Lidingö stads inskannade datapunkter som täcker vatten och mark interpolerades med ArcMap-funktionen *Topo to raster* till ett raster (`2mx2m_interpolerat`). Cellsidans längd sattes till 2 meter baserat på medelavståndet mellan punkterna. Detta raster skiljer sig inte markant från `2mx2m_grid` i bebyggda områden, medan det i obebyggda områden finns större avvikelser. Genom att subtrahera de två rastren från varandra kunde de områden där avvikelserna var mindre än 0,2 meter identifieras. Eftersom `2mx2m_grid` var efterbehandlat av Lidingö stad det vara det mest noggranna rastret i de områden som efterbehandlats. I övriga celler antogs `2mx2m_interpolerat` vara det som gav den bästa representationen, eftersom `2mx2m_grid` där enbart hade interpolerats utan hänsyn till mätdata.



**Figur 7.** Rastercellen *c* med omgivande celler tillhörande antingen raster R1 eller R2. A-D visar olika varianter av utsnitt som används i funktionen.

**Figur 8.** Utsnitt av ett raster med två buffertceller runt sänkcellen, S. Avståndet, *d*, till en godtycklig cell i utsnittet representeras av pilen och mäts mellan cellernas mittpunkter.

De celler som blev utvalda från det ena rastret med den här metoden var inte heltäckande över en begränsad del av ytan, utan hål på några få celler förekom. Om ett lite större område valdes ut att ta värden från ett av rastren var det inte önskvärt att enstaka celler tas från det andra rastret, av kontinuitetsskäl. En funktion (Appendix A) skrevs därför i Matlab för att täcka igen dessa hål. För varje cell beräknade funktionen antalet omgivande celler som tillhörde vartdera rastret. Om den aktuella cellen,  $c$ , tillhörde det aktuella rastret,  $R1$  (Figur 7A), och upp till ett visst antal,  $x$ , av de omgivande cellerna tillhörde  $R1$  tillskrevs cellen det andra rastret,  $R2$ . Om  $x$  eller fler av de omgivande cellerna tillhörde  $R1$ , antogs cellen vara i ett större område av  $R1$  och tillhörigheten ändrades inte (Figur 7B). På motsvarande sätt bedömdes tillhörigheten om cellen tillhörde  $R2$ . Vilket värde för  $x$  som ska användas var väldigt känsligt. Ett för lågt värde tillät inte ändringar på annat än ensamma celler, medan ett för högt värde istället konverterade hela datamängden till det ena rastret utan urskiljning. Eftersom endast den aktuella cellen kunde ändras när funktionen kördes var flera iterationer nödvändigt, särskilt för att komma åt hål som var större än några få celler. Vissa hål var omöjliga att komma åt med den här metoden, varför antalet celler som undersöks varierades (Figur 7C-D). Utnyttjandet av dessa två metoder medförde dock att alla underkanter (Figur 7C) respektive högerkanter (Figur 7D) tillfördes en cell, varför de har använts sparsamt.

I ArcMap skapades polygoner som täcker in de delar som skulle klippas från varje raster och dessa delar sammanfogades till ett enda raster, *mosaicraster\_02*. Rastret klipptes med strandlinjen till Lidingö, så att sjöar och allt som är utanför ön blev definierat som *NoData*. Dessa ytor har exkluderats från all vidare bearbetning. Därefter användes *mosaicraster\_02* som höjdmmodell för samtliga beräkningar.

### 3.3. LEDNINGSNÄTET

Ahlin (2012) tog utifrån givna data fram en fungerande modell över Lidingös dagvattennät. Där ingick bland annat stamledningar, brunnar, pumpstationer, diken, tunnlar och utloppspunkter. Eftersom dagvattennätet här approximeras som vattendrag var det endast intressant hur vattnet rinner på ytan och fram till dagvattenledningen. Därför togs tunnlar och pumpstationer bort från ledningsnätet, då tunnlar inte tillåter inflöde från omgivande mark och tryckledningar transporterar vatten uppåt. Vanliga ledningar som anslöt till en pumpstation eller fortsatte i en tunnel ersattes med en utloppspunkt.

### 3.4. HANTERING AV UTLOPPSPUNKTER

För att framgångsrikt beräkna avrinningsområdena med hänsyn till dagvattensystemet krävdes att i förväg definiera var det fanns utloppspunkter. För detta ändamål programmerades en funktion, *BurnSinks*, i Python till ArcMap, som var körbar på samma sätt som inbyggda verktyg (Appendix C). Den här funktionen finns även som Matlabskript (Appendix B). Funktionen utnyttjade kommandon från ett externt bibliotek till Python, GDAL, vilket installerades separat enligt instruktioner i Appendix E.

Indata till *BurnSinks* var höjdmodellen, en vektordatafil i punktformat där utloppspunkterna fanns angivna samt ett värde för hur stort område runt varje sänka som skulle fungera som buffertområde. För varje (utlopps-)punkt lokaliserades motsvarande

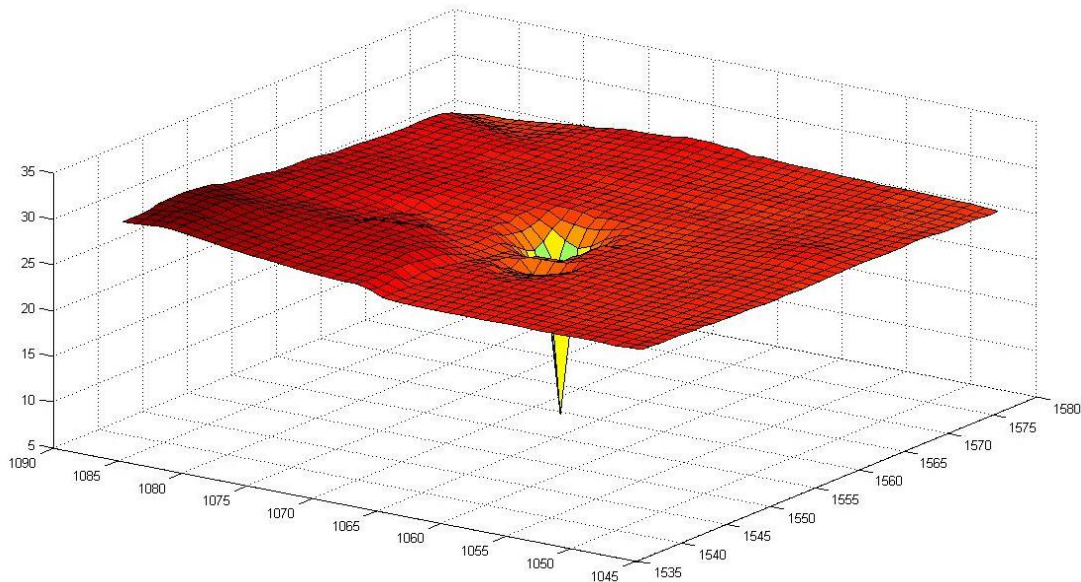
rastercell, sänkcellen, med en precision som beror av upplösningen på rastret. Ett kvadratisk avsnitt av rastret, mängden celler styrt av det användardefinierade buffertavståndet, klipptes ut. Därefter subtraherades ett värde från varje cell beroende på avståndet,  $d$ , till sänkcellen, enligt ett matematiskt samband (3).

$$D(x) = 20e^{-x} \quad (3)$$

Avståndet  $d$  mellan sänkcellen  $S$  och varje annan cell i buffertområdet beräknades trigonometriskt (Figur 8). I sänkcellen subtraherades  $D(0)$ . För de övriga cellerna skapades en vektor med värden för  $x$  i intervallet från 0 till 5 med steg om 0,01. Dessa sträcktes ut över det definierade buffertavståndet (eftersom funktionen skulle fungera för variabelt avstånd). För celler med avstånd  $d$  till sänkcellen och som hamnade innanför buffertavståndet,  $B$ , minskades värdet med  $M$  enligt ekvationerna (4) och (5). För de celler i utsnittet där avståndet till sänkcellen var större än buffertavståndet gjordes ingenting. I ekvationerna (4) och (5) ger  $\text{längd}(x)$  antalet värden i vektorn  $x$  och  $L$  ger ett värde en viss sträcka in i  $x$  som är proportionell mot avståndet till sänkcellen jämfört med buffertavståndet.

$$L = x \left( \frac{d}{B} \cdot \text{längd}(x) \right), \quad d \leq B \quad (4)$$

$$M = D(L) \quad (5)$$



**Figur 9.** Höjdmodellen vid en sänka, uppritat i Matlab. Här är buffertavståndet satt till 10 meter (5 celler), varje cellsida är två meter.

Det ledde till att runt varje sänkceld beräknades en konformad nedsänkning med sänkcelden i centrum, istället för att enbart sänka en enstaka cell (Figur 9). Det tillåter att, under modelleringen av avrinningsområdena, vatten kan rinna in i sänkan från omgivningen på ett mer tillförlitligt sätt, även om sänkan inte hamnade exakt i ledningens ändpunkt.

De celler som ingick i det studerade området runt sänkan men där avståndet till sänkcelden är större än buffertavståndet, det vill säga de celler som är i hörnen, har inte påverkats. Sänkcelds som hamnar utanför hela systemet, och alltså saknar värde, exkluderas i algoritmen. Detsamma gäller för de celler i varje urklipp som hamnar utanför området, även om sänkcelden hade ett värde definierat. När sedan alla celler som ska ändras hade processats i urklippet sparades dessa nya cellvärden tillbaka till rasterfilen. Sedan började programmet om igen för nästa sänkceld och fortsatte så tills alla sänkcelds hade behandlats.

### **3.5. BESTÄMNING AV AVRINNINGSMRÅDEN**

Framtagning av avrinningsområden gjordes i ArcMap 10.0 med *ArcHydro Tools* genom att använda samtliga funktioner som beskrivits i avsnitt 2.3.2 i den ordning de presenteras. Med utgångspunkt från höjdmodellen kördes först den inom projektet skapade funktionen *BurnSinks*, där alla fördefinierade och enligt avsnitt 3.3 inlagda utloppspunkter sänktes ned i rastret. Alla ledningar sänktes sedan ned på motsvarande sätt med *DEM Reconditioning*. Flödesvägarna vid en hydraulisk modellering av området kom därmed att påverkas på så sätt att allt vatten som ”rann” till en ledning ”föll” ner i en djup ränna varifrån det inte kunde ta sig ur. Eftersom rännan skapades genom att subtrahera ett konstant värde från den ursprungliga höjdmodellen, bevarades den lutningen i området. Vattnet rann därför fortfarande enligt topografin, men i dessa påtvingade vattenvägar. De gropar som skapades med *BurnSinks* användes som avrinningsområdenas nedersta utloppspunkter (vanligen användes dessutom ett antal lokala utloppspunkter uppströms). Ledningarna, som representerades av de djupa rännorna, slutade i dessa gropar. Vatten behövde sedan inte ledas vidare härifrån i modellberäkningen utan kunde ”försvinna” i gropen. Detta motsvarade att vatten rann vidare i en tryckledning eller tunnel, till vilken det inte kunde rinna in mer vatten. Dessa kunde därmed inte tas med i modellen representerade som vattendrag.

För *Sink selection* testades ett flertal parameteruppsättningar i syftet att välja ut enbart de relevanta sänkor, det vill säga de som skapades med *BurnSinks*. I de fall där ingen uppsättning fungerade ändrades djupet varmed sänkor brändes ned med *BurnSinks*. I övrigt kördes funktionerna i den ordning som de presenterats ovan.

### **3.6. HANTERING AV SMÅ AVRINNINGSMRÅDEN**

Varje delsträcka av en definierad vattenväg tilldelades ett eget avrinningsområde (avsnitt 2.3.2). Därför var de avrinningsområden som skapades av varierande storlek, till exempel blev de jämförelsevis små där flera strömmar gick ihop. Ett avrinningsområde som endast täckte några få celler, men även några hundra kvadratmeter, var inte meningsfullt att ha kvar. De ger upphov till väldigt stora resultatfiler vid senare modellering av dagvattennätet, eftersom det blir onödigt många avrinningsområden. För

sådana små områden skrevs en funktion för att slå ihop dem med ett närliggande område. För att vara konsekvent med vilket annat område som sammanfogningen skedde valdes det område som låg närmast nedströms. En funktion skrevs i Python, *PolygonEdit*, som var tänkt att utföra detta (Appendix D). Funktionen kunde inte färdigställas inom den tid som avsatts för projektet. Nedan beskrivs hur funktionen var tänkt att fungera.

Först krävdes att nedströmsområdet till varje avrinningsområde identifierades. Till det användes en funktion i *ArcHydro Tools*; *Find next downstream line* (avsnitt 2.3.2), vilket resulterade i en kolumn med nedströmsområdets identifikationsnummer. Den här funktionen fungerade bara på filen med av modellen framtagna vattendrag. Viss bearbetning var därför nödvändig innan det gick att utnyttja identifikationsnumren för att koppla till ett visst område.

Indata till funktionen var polygoner med de framtagna avrinningsområdena samt ovan beskrivna fil med vattendrag och kolumn med nedströmsområden. Det första steget var att identifiera och plocka ut de områden som var mindre än en given storlek, vilket var en användardefinierad storhet. För varje utvalt område identifierades dess nedströms område. Båda polygonerna kopierades över till en ny fil där de sammanfogades till en enda polygon. Den här polygonen lades till i en kopia av indatafilen; utdatafilen. Slutligen raderades alla små polygoner som nu sammanfogats med andra.

## 4. RESULTAT

Kapitlet beskriver först de delresultat som togs fram under behandlingen av data och var avgörande för beslut om hur nästa steg skulle tas. Därefter följer en beskrivning av hur de i projektet framtagna funktionerna implementerades och utnyttjades i Arcmap, samt resultatet från beräkningar med dem. Slutligen presenteras de framtagna avrinningsområdena, vilka även jämförs med hur de sett ut utan ovan beskrivna modifieringar av beräkningsmetoden.

### 4.1. DATABEHANDLING

Under bearbetningen av raster- och vektordata beräknades parametrar av värde för den fortsatta arbetsgången.

- För de mätdata som Lidingö stad tillhandahöll beräknades medelavståndet mellan punkterna till 1,3 meter, vilket låg till grund för beslutet att göra ett raster med 2 meter långa sidor på rastercellerna.
- Avvikelsen mellan det av Lidingö tillhandahållna rastret (2mx2m\_grid) och det från punkterna framtagna rastret (2mx2m\_interpolerat) uppgick som mest till 14 meter. I medeltal avvek värdet med 1,48 meter. I 82,3 % av cellerna var avvikelsen mindre än 0,2 meter (Tabell 1).

**Tabell 1.** Andel av cellerna som avviker mer än en viss höjd vid en jämförelse mellan 2mx2m\_grid och 2mx2m\_interpolerat.

Differens [m]	0,1	0,2	0,5	1	2	5	10
Andel avvikande celler [%]	22,5	17,7	13,5	11,3	8,6	3,98	0,8

- När de två rastren skulle sammanfogas kördes Matlabfunktionen ett flertal gånger för att eliminera ensamma celler och små kluster som låg i ett större område av andra celler tillhörande det andra rastret. Efter de första tio itereringarna hade en stor del av hålen åtgärdats. Den elfte iterationen kördes enligt Figur 7D och den 16:e enligt Figur 7C. Totalt kördes funktionen 20 gånger innan det stora flertalet av ensamma celler var åtgärdade.
- Förbehandling av ledningsnätet ledde till att 15 linjer som representerade tunnlar togs bort och elva nya utloppspunkter lades till för att ersätta dem. Dessutom raderades fem tryckledningarna, och fyra utloppspunkter placerades där en pumpstation var markerad. 39 diken lämnades orörda.
- När *DEM Reconditioning* kördes på det framtagna rastret (mosaicraster\_02) användes dagvattenledningarna som vattenvägar, som sänktes ned i höjdmodellen. I en höjdmodell med ett höjdomfång på < 100 m, vilket var fallet här, noterades att antalet buffertceller bör hållas lågt, lämpligen 1, samt att 10 m mjuk nedsänkning (*Smooth drop*) och 100 m skarp nedsänkning (*Sharp drop*) gav ett tillräckligt bra resultat.

- Inför *Sink prescreening* uppmättes arean av en konstruerad sänka till 11 m<sup>2</sup>. För att dessa inte skulle fyllas igen sattes areagränsen till 8 m<sup>2</sup>. Det resulterade i ett mycket litet antal igenfyllda sänkor (18 av cirka 40 000), varför detta steg skulle kunna ha utelämnats.
- När de sänkor som skulle bevaras valdes ut med *Sink selection* sattes parametern som undersöker minsta djup till 10 meter, övriga användes inte (= 0). Det resulterade i att alla sänkor som låg inom höjdmodellen valdes ut.
- *Stream definition* kördes med olika tröskelvärden för hur stor yta som skulle avvattnas via en specifik cell för att cellen skulle räknas som en del av ett vattendrag. Standardvärdet på 1 % av totala ytan testades först, sedan minskades tröskelvärde successivt och antalet avrinningsområden på den 3000 ha stora ön noterades (Tabell 2). Ett lägre tröskelvärde resulterade i fler avrinningsområden eftersom fler strömsegment skapades och det var utifrån dem som avrinningsområdenas utloppspunkter definierades. För den hydrauliska modelleringen är det oftast inte relevant med avrinningsområden som är mindre än 0,5 ha, vilket motsvarar ungefär två villatomter. Notera att i tabellen är endast avrinningsområdenas medelstorlek angiven.

**Tabell 2.** Antalet avrinningsområden vid beräkningar med olika tröskelvärden och deras genomsnittliga area för varje fall.

Angivet tröskelvärde [ha]	2,7	2,0	1,6	1,35	1,2	0,8
Antal skapade områden	528	732	923	1087	1295	1988
Genomsnittlig area [ha]	5,7	4,1	3,3	2,8	2,3	1,5

## 4.2. FUNKTIONER

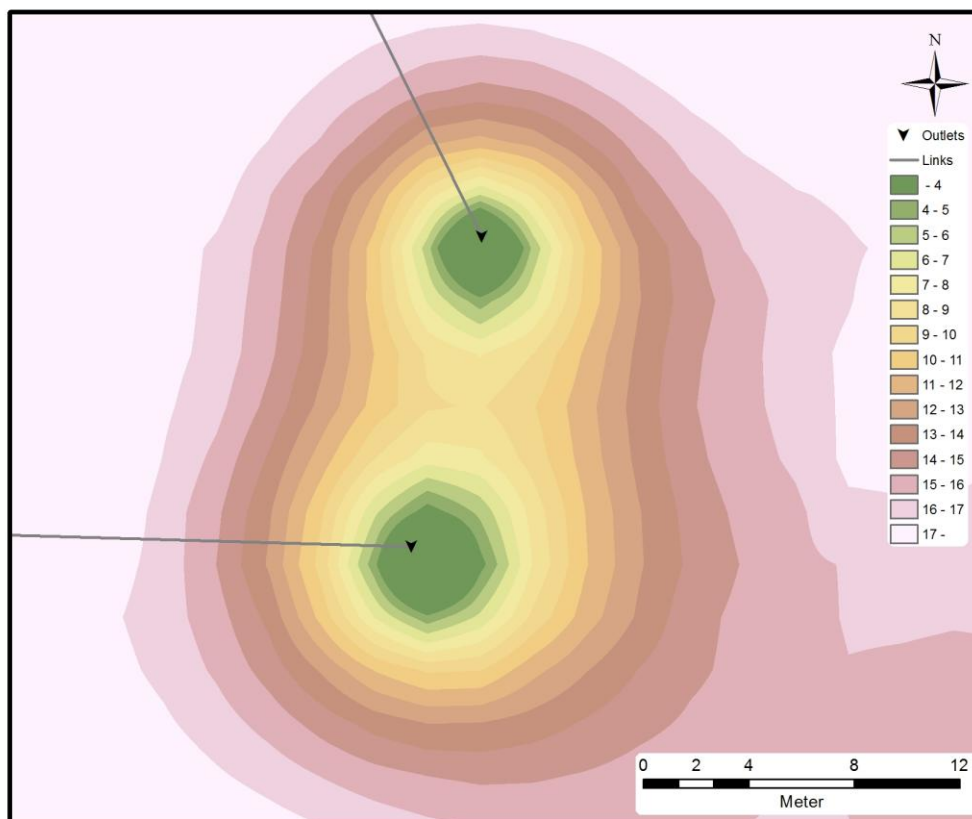
### 4.2.1. BurnSinks

Den Pythonbaserade funktionen *BurnSinks* implementerades i ArcMap på ett sådant sätt att den gick att använda på samma sätt som övriga, inbyggda funktioner (Figur 11). Även hjälptexter och vägledande instruktioner inkluderades.

Vid användningen av *BurnSinks* specificerades hur stort buffertområdet runt utloppspunkten skulle vara. Området sänktes ned gradvis för att skapa den konliknande formen (Figur 9). För att inte påverka höjdmodellen mer än nödvändigt önskades ett så litet buffertområde som möjligt, men som fortfarande uppfyllde sitt syfte. För höjdmodellen över Lidingö noterades att ett buffertområde med 2 meters radie (1 cell i vardera riktningen) var tillräckligt för att metoden i *ArcHydro Tools* skulle kunna identifiera punkten som en sänka (Figur 10). Funktionen kördes innan *DEM Reconditioning* när avrinningsområdena skulle beräknas.

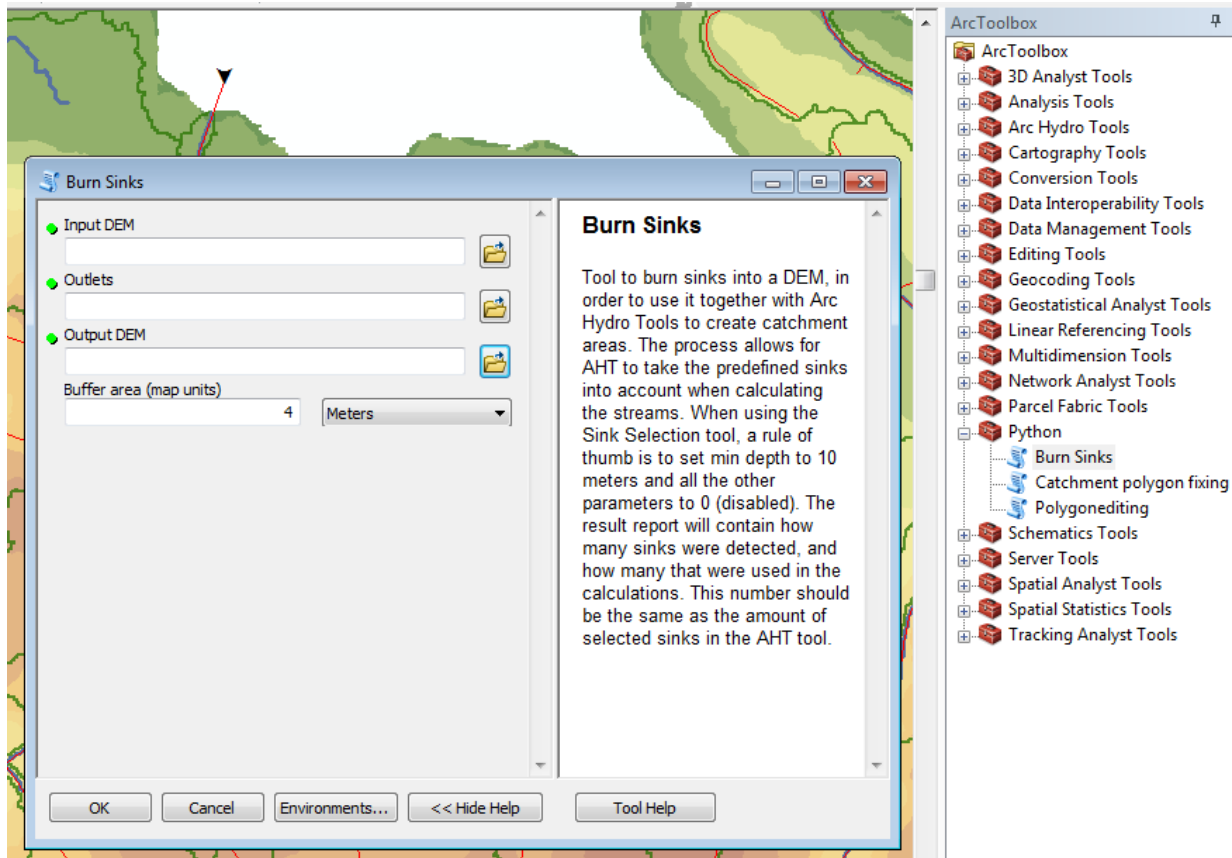
### 4.2.2. Polygonedit

Pythonskriptet *PolygonEdit* färdigställdes aldrig inom ramen för projektet. I likhet med *BurnSinks* implementerades skriptet som en funktion i ArcMap (Figur 12).

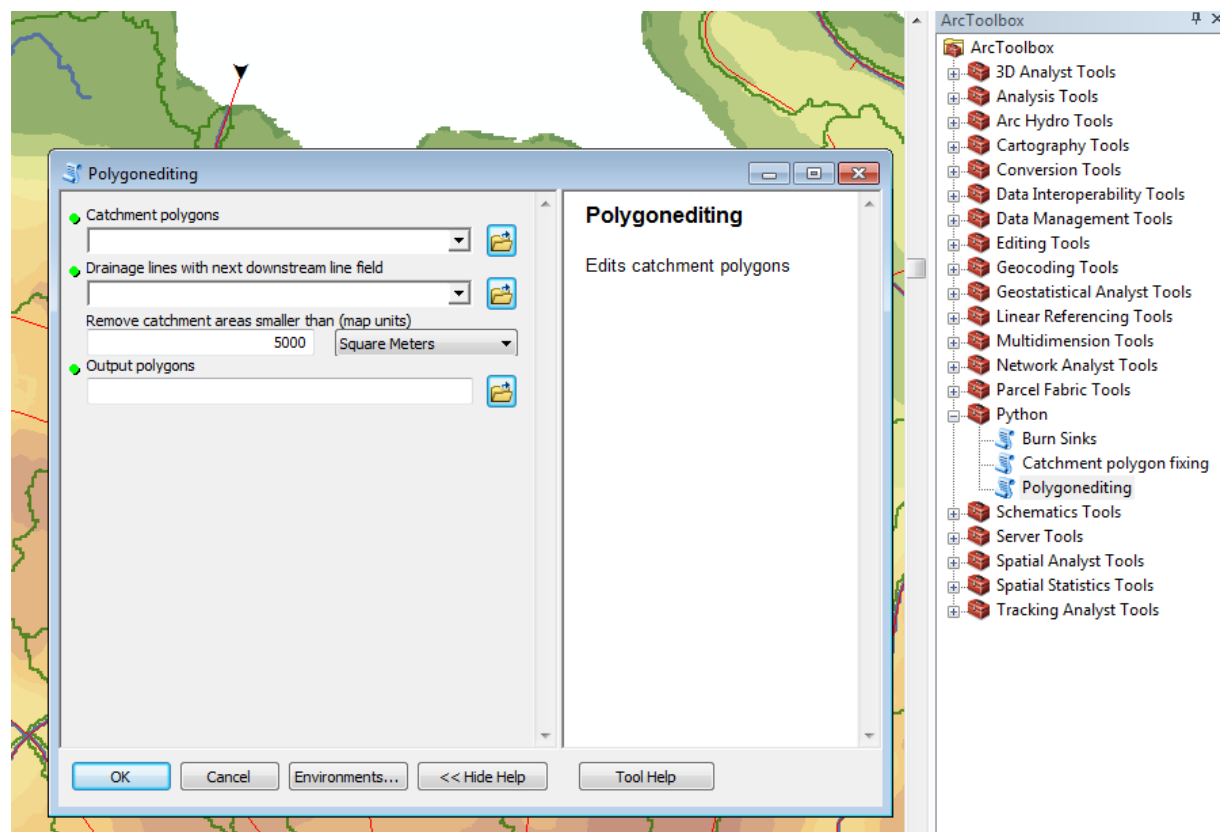


**Figur 10.** Nedsänkning av utloppspunkter i höjdmodellen. Här har 8 meters buffertområde använts.





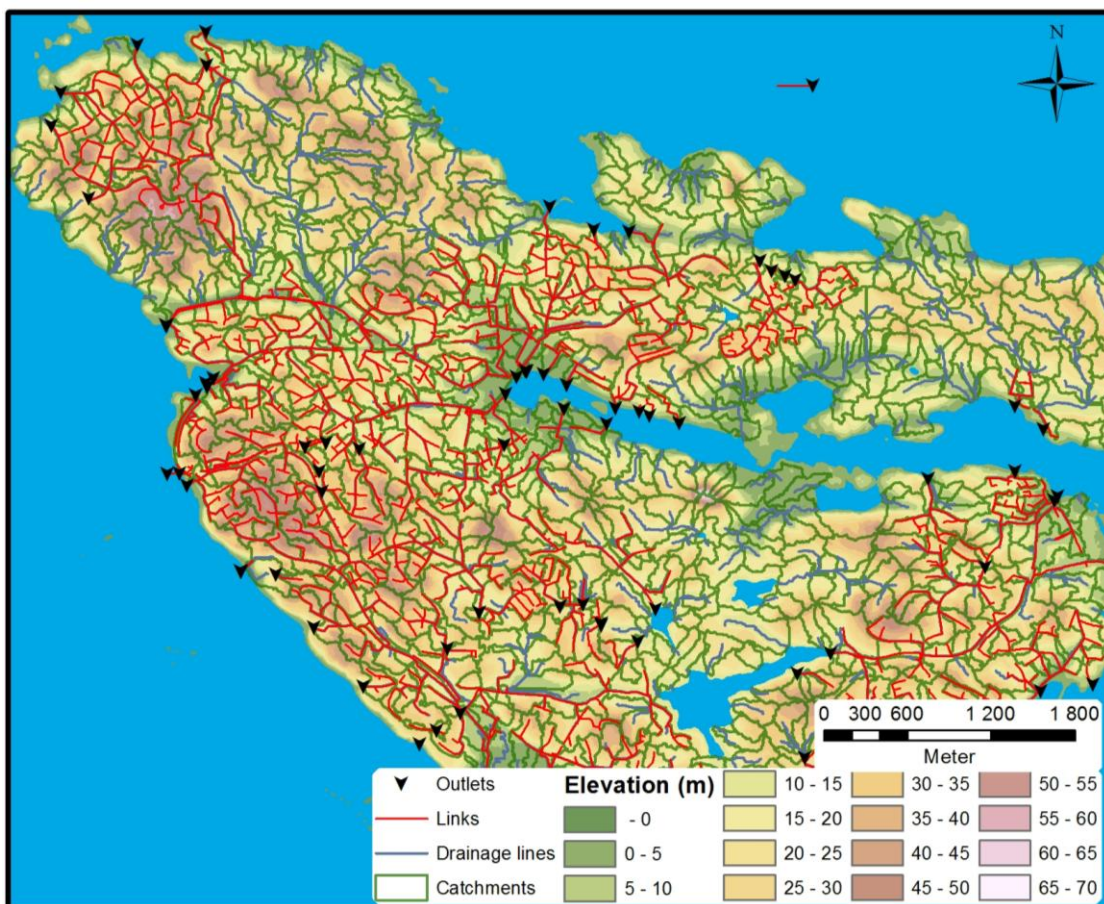
Figur 11. Skärmbild som visar hur funktionen *BurnSinks* ser ut när den används från ArcMap.



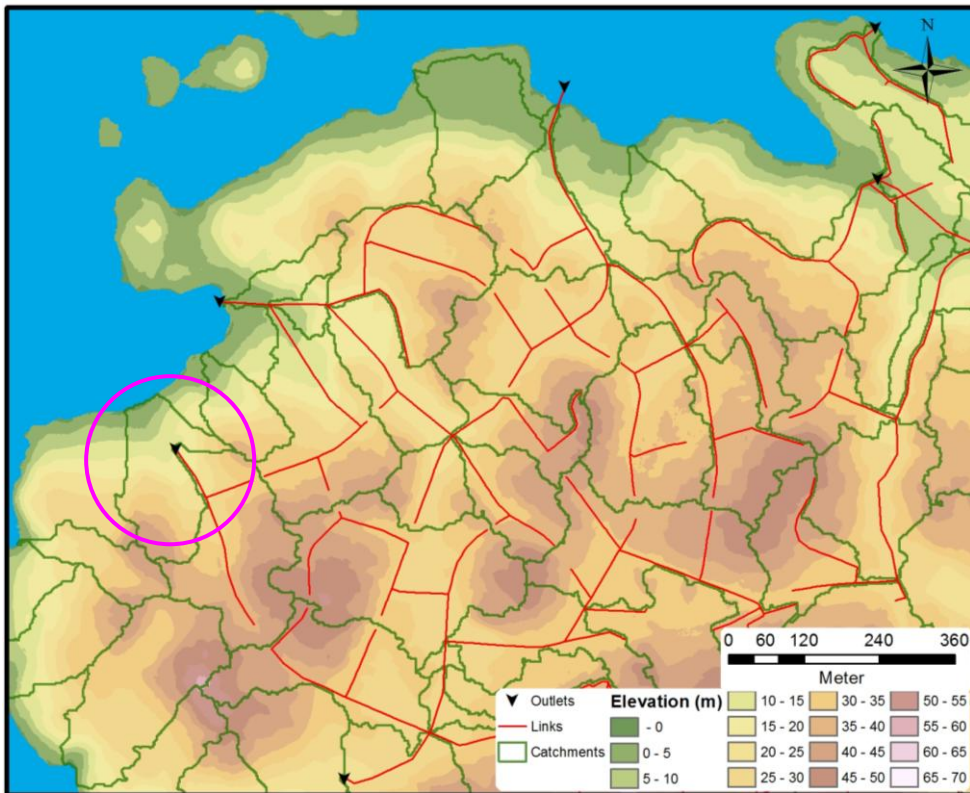
Figur 12. Skärmbild av funktionen *PolygonEdit* i ArcMap.

### 4.3. AVRINNINGSSOMRÅDEN

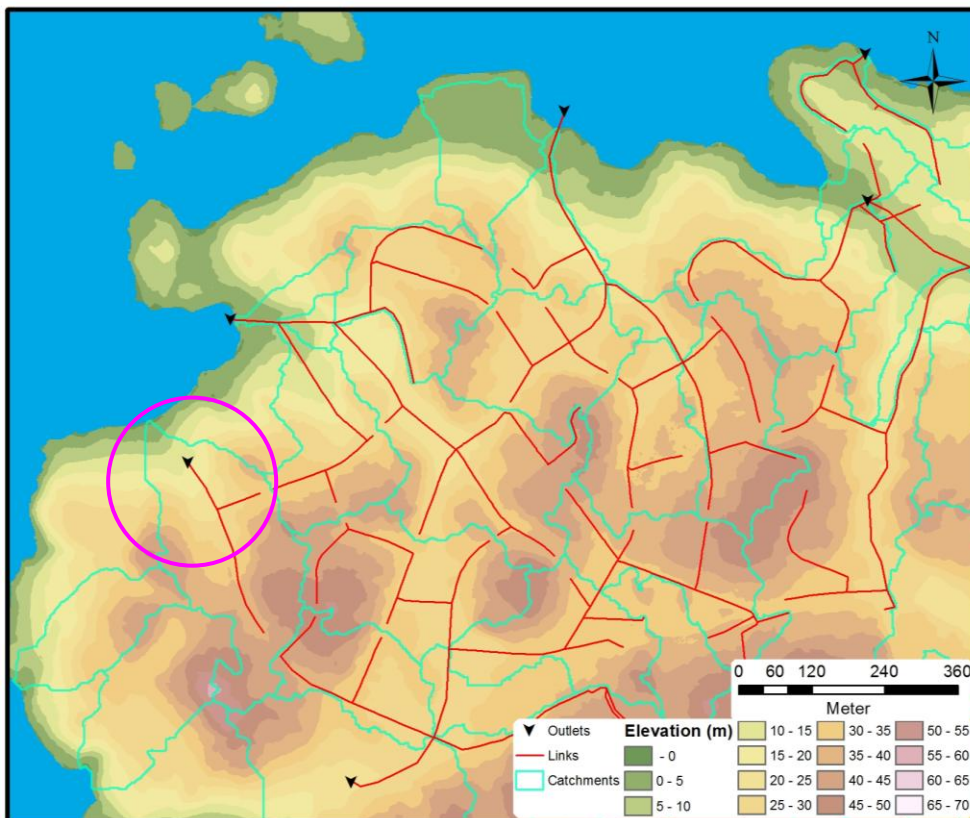
Avrinningsområden togs fram för Lidingö (Figur 13). En förstoring i den nordvästra delen visar mer i detalj hur avrinningsområdena anpassats efter dagvattennätet (Figur 14). Notera att vid den markerade utloppspunkten tar avrinningsområdet slut trots att den ligger en bit in från strandlinjen. En jämförelse med hur avrinningsområdet hade sett ut utan att hänsyn tagits till de lokala utloppspunkterna togs också fram (Figur 15). Vid framtagningen användes *mosaic\_raster02* som höjdmödel. Här beräknades 42 sänkor av totalt 88 stycken ligga innanför rastret, vilka kunde väljas ut med *Sink selection* efter att *BurnSinks* använts (med parametervärden enligt ovan). Det resulterade i 1 132 avrinningsområden med en medelarea på 23 200 m<sup>2</sup> varav 171 var mindre än 5 000 m<sup>2</sup>.



**Figur 13.** Beräknade avrinningsområden för Lidingö. Gröna linjer representerar vattendelare och röda dagvattenledningar. Trianglarna är utloppspunkter.



**Figur 14.** Avrinningsområden i en del av nordvästra Lidingö. Gröna linjer representerar vattendelare och röda dagvattenledningar. Trianglarna är utloppspunkter. Den markerade utloppspunkten är längst ned i ett avrinningsområde, vattnet från det området leds följaktligen inte till havet.



**Figur 15.** Avrinningsområden i en del av nordvästra Lidingö. Turkosblå linjer representerar här vattendelare och röda dagvattenledningar. Den markerade utloppspunkten tas inte hänsyn till vid beräkningen av avrinningsområden, varför det berörda avrinningsområdet slutar i havet.



## 5. DISKUSSION

Många kommuner och andra aktörer använder idag modellering när ledningars kapacitet ska bedömas. Avrinningsområden är en central parameter vid dessa modelleringar. De förhållandevis enkla modifieringar som presenteras i den här rapporten har visat att det är möjligt att anpassa en befintlig metod för bestämning av avrinningsområden så att den kan ta hänsyn till dagvattennätet. Metoden har utvärderats på data över Lidingön, men även testats för att ta fram avrinningsområden över ett annat område i Molkom, Värmland. Manuell bearbetning kunde inte undvikas helt, främst eftersom fastighetsgränserna samt att avgränsningarna inte delade av byggnader ansågs överordnat topografien (Hammarlund, pers. medd.). Dagvatten som samlas in på en viss fastighet eller från en viss byggnad leds i normalfallet till en och samma dagvattenledning, det är ovanligt med fastigheter som är anslutna till mer än en ledning. För dagvattennät har de naturliga avrinningsområdena betydelse och är bra att ha som utgångspunkt, eftersom vatten från hela arean bidrar till flödet i ledningarna. De resulterande avrinningsområdena användes sedan vid modellering av ledningssystemet i Molkom (Näsman-Melander, pers. medd.).

Möjligheten att hantera pumpstationer och tunnlar i form av lokala utloppspunkter ledde till att allt vatten inte behövde "färdas" till kanterna av höjdmodellen. Det var en förutsättning för att få korrekt avbildade avrinningsområden. Ledningsnätet modelleras som vattendrag och det innebär en del generaliseringar. Dynamiken och funktionen hos en ledning är förstås inte samma som för en bäck. Den största skillnaden, som här inte har tagits någon hänsyn till, är att till varje delsträcka av en bäck kan vatten rinna obehindrat från terrängen. Men en dagvattenledning kan vattnet endast nå genom de brunnar som installerats (bortsett från eventuell inträngning direkt i ledningen i marken). Det bygger på att marken ovanför ledningen har planerats på ett sådant sätt att ytvattenavrinningen leds till dagvattenbrunnen.

Jankowfsky m.fl. (2012) har gjort en studie som jämför olika automatiska beräkningssätt med att manuellt och i fält avgränsa avrinningsområden för ett område i östra Frankrike. Bland de jämförda metoderna finns flera olika varianter av att bränna in ledningsnät, bäckar, byggnader och vägar i höjdmodellen för att sedan utnyttja för avgränsningen av delavrinningsområdena. En av slutsatserna i studien är att avrinningsområdena kan beräknas relativt bra, om modellering kombineras med ett visst mått fältarbete. En höjdmodell med lägre upplösning (25 m) användes, vilket delvis skulle kunna förklara att de kommer fram till att djupet varmed ledningarna bränns in har stor betydelse.

I en studie av Chen m.fl. (2003) har jämfört olika, vid tidpunkten aktuella, programvaror och funktioner för att bränna in ett ledningsnät i en höjdmodell. Slutsatsen är även här att manuella mätningar (vilket används som referens) inte helt kan ersättas av digitala beräkningar, men att bränna in ledningsnätet är en metod som med rätt indata kan resultera i avrinningsområden som är användbara.

## 5.1. BURNSINKS

*BurnSinks* fungerade bra ihop med *Sink selection* för att identifiera och bevara utloppspunkterna vid framtagningen av avrinningsområdena. *BurnSinks* utnyttjar en öppning i *ArcHydro Tools* metod för att beräkna avrinningsområden, som egentligen är avsedd för att förhindra att områden som ligger naturligt lågt i det undersökta området fylls igen. Vidare beräkningar kan sedan visa eventuella översvämmade områden och lågpunkter i terrängen. Utmaningen var att skapa en sänka som var lätt att identifiera med *Sink selection* och samtidigt innebar så liten förändring av den ursprungliga höjdmodellen som möjligt. Det resulterade i djup sänka med liten yta. Det var inte alltid som det fungerade helt problemfritt. Av ej undersökt anledning lyckades algoritmen inte med att identifiera vissa sänkor. En lösning blev att köra *Sink selection* fler gånger med mindre variationer i parametrarna och addera resultatet till tidigare utvalda sänkor. *BurnSinks* räknade också hur många av utloppspunkterna som låg inom rastret (det var inte helt ovanligt med utloppspunkter utanför), vilket användes för att manuellt kontrollera att antalet utvalda sänkor stämde.

Resultatet skiljer sig på förväntat sätt när avrinningsområden som tagits fram med hjälp av *BurnSinks* jämförs med standardmetoden (Figur 14-15). Den inringade utloppspunkten fungerar som avrinningsområdets utloppspunkt om metoden används.

Programmering är inte en vetenskap med endast en korrekt lösning. Det finns många olika sätt att besvara en och samma frågeställning, vissa är effektivare än andra. *BurnSinks* kan exempelvis utvecklas med bättre problemhantering, resultat- och felmeddelanden samt ytterligare användarinställningar. Det är antagligen också möjligt att utan allt för stora förändringar skriva om funktionen så att GDAL-biblioteket inte utnyttjas, vilket skulle underlätta de initiala installationskraven inför användandet av funktionen.

### 5.1.1. Begränsningar

Metoden som används i *BurnSinks* tar inte någon hänsyn till när två sänkor ligger närmare varandra än buffertavståndet. Cellerna som överlappar kommer att sänkas ned dubbelt. En avståndskontroll skulle kunna införas för att hantera detta, men problemet bör inte medföra några avgörande skillnader i resultatet. För väldigt stora raster kan det uppstå problem med minneshantering. Eftersom hela rastret läses in som en vektor och tillfälligt lagras i RAM-minnet kommer datorns kapacitet att styra hur stora raster som algoritmen klarar att hantera. Möjliga lösningar är till exempel att enbart ett område runt varje enskild sänkcell läses in som en vektor och sparas tillbaka till rastret efter att beräkningarna gjorts. För nästa sänkcell läses sedan en ny bit av rastret in. Det finns möjlighet att specificera på vilken plats i koordinatsystemet som inläsningen ska börja i den funktion som används för att lagra rastret i vektorform. Tyvärr är det inte lika enkelt att lagra över endast en del av ett raster med nya värden, varför detta ansågs som en onödigt krånglig lösning även om det teoretiskt skulle kunna möjliggöra hantering av närmast oändligt stora raster. En bugg i ArcMap 10.0 (åtminstone SP 1-3) gör också att minnet inte töms förrän programmet stängs, varför konsekutiva körningar kan ge ett felmeddelande. Lösningen är oftast att starta om ArcMap.

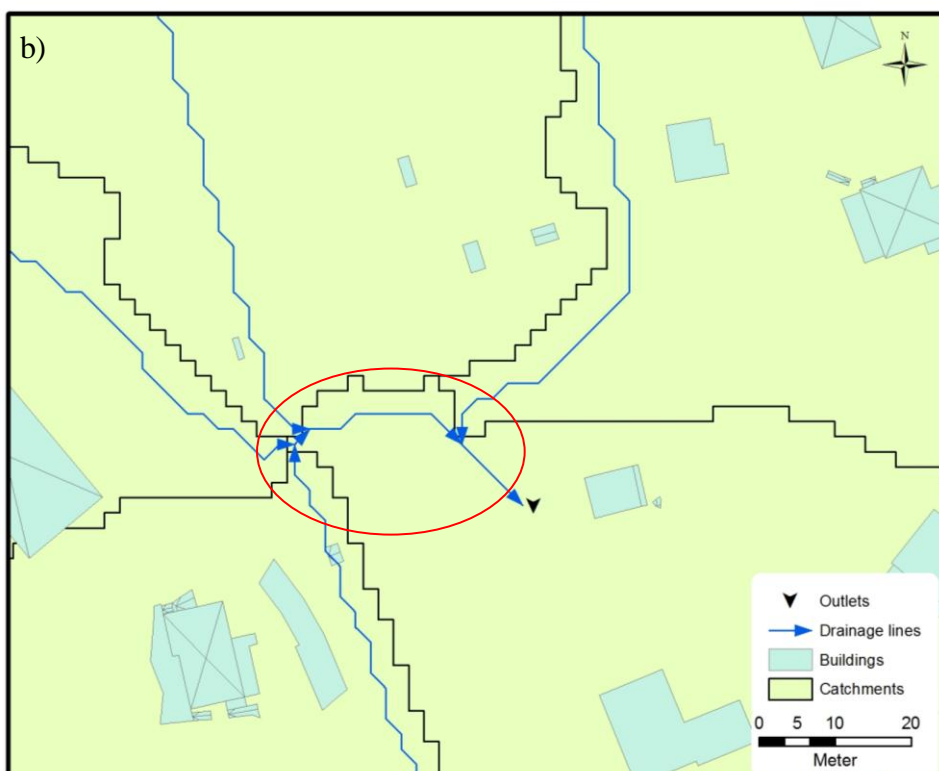
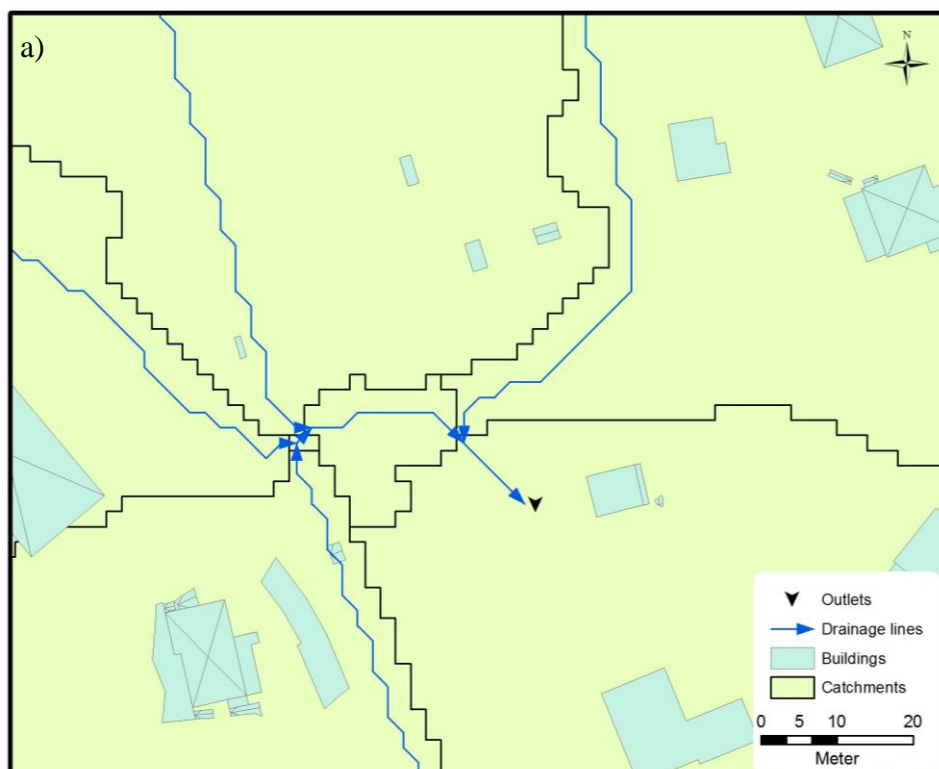
En enklare metod för att öka den maximala möjliga storleken på rastret, som också har sina begränsningar, är att ändra att värdena lagras i *int*-format istället för *float*. Det innebär att alla data lagras som heltal istället för decimaltal, vilket förstas påverkar detaljgraden. Varje cellvärde tar därmed mindre minnesutrymme, varför större raster kan hanteras. Den här ansatsen bygger dock fortfarande på att hela rastret simultant lagras i minnet, med de begränsningar som det innebär.

## 5.2. POLYGONEDIT

Funktionen *PolygonEdit* var tänkt att användas till efterbehandling av de framtagna avrinningsområdespolygonerna. Eftersom alltför små områden bidrar till tunga resultatfiler vid modellering av dagvattensystemet (det blir många områden som behöver modelleras) utan att tillföra tillräckligt med information var det önskvärt att ta bort dessa. Vad som räknas som för små avrinningsområden är förstas en bedömningsfråga. Förutom ett minsta värde för arean är ett möjligt kriterium att se till att alla områden innehåller minst en dagvattenbrunn. Finns inte någon brunn så kan inte vattnet som avrinner från området ta sig ner i dagvattenledningen. Det är bara kriteriet baserat på en minsta yta som har implementerats i funktionen i dagsläget. I Figur 16 visas hur resultatet skulle ha blivit om funktionen hade fungerat som planerat. Två polygoner i de framtagna avrinningsområdena i Figur 16a) har sammanfogats manuellt till ett nedströms liggande område (Figur 16b), det området som utloppspunkten ligger i. Pilarna på linjerna visar flödesriktningen.

### 5.2.1. Begränsningar

Vidareutveckling och förbättring av algoritmen kan innefatta att den testas för ovan nämnda kriterium om att varje område bör innesluta en brunn. Det kan göras genom att utnyttja befintliga funktioner i ArcMap för geografiskt överlappande objekt. Det förekommer också att områden väljs ut i algoritmen som inte har något nedströmsområde, främst för att dessa innesluter en utloppspunkt. I de fallen görs i nuläget ingenting, men det kan vara lämpligt att åtminstone ett varningsmeddelande skrivs ut som anger vilka områden det berör. Det kan också uppstå komplikationer om två ”för små” områden följer på varandra i flödesriktningen. Eftersom geometrierna inte uppdateras direkt kan båda läggas till sina respektive nedströmsområden, vilket blir fel för det övre området eftersom det läggs till ett område som sedan inte finns kvar. Det kan lösas genom att varje nytt utvalt område och dess nedströmsområde kontrolleras mot tidigare behandlade områden för att undvika överlappningar. I de fallen får en speciell metod appliceras, som istället adderar det aktuella området till ett tidigare skapat, sammanfogat område. Funktionen i sin nuvarande utformning kopierar inte över alla attributvärden, vilket skulle kunna läggas till för att bevara data. Nya, unika identifikationsnummer kan också vara värdefullt att addera till de nyskapade geometrierna.



**Figur 16.** Avrinningsområden som a) är beräknade med ArcHydro Tools och b) som de skulle ha blivit efter att *PolygonEdit* körts, om funktionen hade fungerat.

### 5.3. ALTERNATIVA METODER

Avsaknaden av utvecklade och publicerade verktyg för bestämning av avrinningsområden i urban miljö kan bero på flera saker. Det kan vara ett område som inte är prioriterat eftersom befintliga funktioner, de som antingen inte alls tar hänsyn till topografin eller enbart bygger på topografin, anses tillräckliga för de avsedda användningsområdena. Det kan vara problematiskt att ta fram en metod som på ett realistiskt sätt tar hänsyn till ledningsnätet vid framtagning av avrinningsområden. Det kan helt enkelt ha ansetts för svårt eller inte varit möjligt eller värt besväret att ta fram en algoritm som tar in ledningarna i beräkningarna.

De modellerare som har konsulterats anser inte att de befintliga funktionerna täcker deras behov, varför de vanligen tar fram avrinningsområden genom att rita ut dem manuellt (Hammarlund, pers. medd.). Att utveckla en ny algoritm för bestämning av avrinningsområden där ledningsnätet räknas in kan förstås göras på en mängd olika nivåer, från att skapa en helt ny metod till olika grad av modifikation av befintliga metoder. Den här rapporten är ett exempel på det senare. En helt ny algoritm kan visserligen vara kostsam att ta fram och det är möjligt att det inte finns tillräckligt kommersiellt underlag för en sådan investering. I slutfasen av projektet framkom att DHI (2012) har lagt till en funktion i sin senaste utgåva av Mike Urban (version 2012) som, enligt uppgift, beräknar avrinningsområden utifrån både topografiska data och ledningsnät. Det är dock inte klarlagt om funktionen faktiskt är fungerande ännu, eller bara en anvisning om vad som ska komma. De tillgängliga licenserna och tiden tillät inte en utvärdering av den nya funktionen inom det här projektet, men inkorporeringen av en sådan länk i programmet visar att det är genomförbart eller åtminstone pågår arbete att ta fram en sådan här funktion.

### 5.4. VIDARE STUDIER

De funktioner som utvecklats här har inte testats fullständigt och inte med många olika datauppsättningar. Koden kan säkerligen förbättras med avseende på robusthet och noggrannhet. Dessutom kan implementering av ovan diskuterade vidareutvecklingar och problemhanteringar underlätta användningen och utvärderingen av resultatet. Under projektet har också föreslagits att samtliga brunnar används som utloppspunkter i *BurnSinks*. Det skulle ge ett avrinningsområde till varje brunn.

Som nämns i frågeställningarna fanns vid projektets början även idéer om att skapa en funktion som hanterar alltför stora avrinningsområden. Det skulle kunna göras genom att utnyttja *BurnSinks*. De områden som bedöms som för stora, vilket kriterium som nu är lämpligt för det, väljs ut efter en första beräkning av avrinningsområden. På en lämplig plats definieras och lagras sedan en delningspunkt i varje område, förslagsvis utmed en vattenväg. Därefter körs *BurnSinks* med delningspunkterna som sänkor och hela processen görs om igen. På så vis tvingas en utloppspunkt att skapas där de stora avrinningsområdena skulle delas och ett nytt område beräknas för den delen som är uppströms.

Fastighetsgränser är, som beskrivits ovan, en förhållandevis viktig faktor när avrinningsområdena används vid modellering av dagvattensystemet. En funktion som



kan ta hänsyn till fastighetsgränserna och anpassa områdena efter dem har diskuterats. Diskontinuiteten i gränserna och det faktum att inte alla linjer är relevanta för en sådan användning innebär dock vissa programmeringsutmaningar. Vägsträckningar skulle också kunna utnyttjas för att efterbehandla avrinningsområdena. Vanligen sammanfaller ledningarna med vägar, där fastigheter på båda sidor om vägen dräneras till ledningen. Med nuvarande, i grunden topografibaserade, metod kan det bli avvikelser om området ligger i en sluttning. Avgränsningen sker då precis nedanför vägen. De fastigheter som ligger längre ner i terrängen än vägen hamnar i ett annat avrinningsområde, trots att de (om lutningen inte är allt för kraftig) sannolikt är kopplade till ledningen i vägen. Att skapa ett buffertavstånd runt vägarna där inte gränser till avrinningsområdena får gå kan vara en lösning, eventuellt med ett filter där endast de vägar som är i ett visst lutningsintervall och riktade utmed sluttningen tas med. Detta implementeras lämpligen i kombination med funktionen om fastighetsgränserna så att passande avstånd ansätts.

En jämförelse mellan metoden som presenterats här och andra metoder har gjorts inom ett parallellt examensarbete (Gustafsson, 2013). De resulterande avrinningsområdena har testats i modelleringar av systemet, liksom manuellt avgränsade avrinningsområden och områden som bestämts bland annat med de metoder som finns tillgängliga i Mike Urban. Resultatet har ännu inte sammanställts.

## 6. SLUTSATSER

- Det är möjligt att modifiera och utvidga en befintlig metod (*ArcHydro Tools*) för bestämning av avrinningsområden så att den tar hänsyn till dagvattennätet.
- Pumpar och annat som kan ses som utloppspunkter i systemet hanteras som sänkor i beräkningarna och med här framtagen funktion kan dessa bevaras vid bestämningen av avrinningsområdena.
- De resulterande avrinningsområdena kräver fortfarande manuell bearbetning innan de kan användas till modellering av dagvattennätet. De utgör dock en värdefull utgångspunkt, eftersom den omgivande terrängen bidrar till flödet i ledningarna.
- Ytterligare funktioner för efterbehandling, främst den ännu inte fungerande funktionen för hantering av små avrinningsområden, har potential att minska behovet av manuell bearbetning.

## 7. REFERENSER

### 7.1. SKRIFTLIGA REFERENSER

- Ahlin, E. (2012) *Modellering av dagvattennät utgående från markhöjder*. Examensarbete i miljö- och vattenteknik, Institutionen för geovetenskaper, luft-, vatten- och landskapslära, Uppsala Universitet, ISSN 1401-5765
- Berne, A., Delrieu, G., Creutin, J.D., Obled, C. (2004) *Temporal and spatial resolution of rainfall measurements required for urban hydrology*. Journal of Hydrology, 299, sid. 166-179
- Chen, M., Tucker, C., Vallabhaneni, S., Koran, J., Gatterdam, M., Wride, D. (2003) *Comparing different approaches of catchment delineation*. CDM, Cincinnati.
- Dahlström, B. (2010) *Regnintensitet – en molnfysikalisk betraktelse*. Svenskt Vatten Utveckling, Rapport 2010-05
- DHI (2012) *Mike Urban Release Note 2012*. Tillgänglig <http://releasenotes.dhigroup.com/2012/MIKEURBANrelinf.htm> (2013-02-14)
- ESRI (2011) *Arc Hydro GP Tools 2.0 Tutorial*. ESRI. Tillgänglig: [http://www.unc.edu/courses/2010spring/geog/591/001/students/cbuch/ex3/ArcHydroTools\\_2\\_0/Arc%20Hydro%20GP%20Tools%202.0%20-%20Tutorial.pdf](http://www.unc.edu/courses/2010spring/geog/591/001/students/cbuch/ex3/ArcHydroTools_2_0/Arc%20Hydro%20GP%20Tools%202.0%20-%20Tutorial.pdf)
- ESRI (2012a) *ArcGIS Resource Center, Desktop Help 10.1; Arcpy*. Tillgänglig <http://resources.arcgis.com/en/help/main/10.1/index.html#/000v000000v7000000> (2013-02-20)
- ESRI (2012b) *ArcGIS Resource Center, Desktop Help 10.0*. Tillgänglig: <http://help.arcgis.com/en/arcgisdesktop/10.0/help/> (2013-01-24)
- Grip, H., Rodhe, A. (1994) *Vattnets väg från regn till bäck*. Hallgren & Fallgren Studieförlag AB, Karlshamn 2003
- Gustafsson, H. (2013) *Utvärdering av avrinningsområden i Mike Urban*. Examensarbete vid civilingenjörsprogrammet i Samhällsbyggnad, KTH, Stockholm.
- Harrie, L. (2008) *Geografisk informationsbehandling*. Forskningsrådet Formas, Stockholm 2008
- Jacobson, C.R. (2011) *Identification and quantification of the hydrological impacts of imperviousness in urban catchments: A review*. Journal of environmental management, 92, sid. 1438-48
- Jankowfsky, S., Branger, F., Braud, I., Gironás, J., Rodriguez, F. (2012) *Comparison of catchment and network delineation approaches in complex suburban environments: application to the Chaudenne catchment, France*. Published online in: Hydrological processes, 2012

Lantmäteriet (2012) *Infoblad n:o 9 Sweref 99 och WGS 84*. Tillgänglig  
[http://www.lantmateriet.se/Global/Kartor%20och%20geografisk%20information/GPS%20och%20m%C3%A4tning/Referenssystem/Inforande\\_av\\_nya\\_referenssystem/info\\_blad-9.pdf](http://www.lantmateriet.se/Global/Kartor%20och%20geografisk%20information/GPS%20och%20m%C3%A4tning/Referenssystem/Inforande_av_nya_referenssystem/info_blad-9.pdf) (2013-02-15)

Learnpython.org (-) *Tutorial*. Tillgänglig  
<http://www.learnpython.org/page/Variables%20and%20Types> (2013-02-20)

Naoum, S., Tsanis, I.K. (2004) *Ranking spatial interpolation techniques using a GIS-based DSS*. Global nest: the international journal. Vol 6, nr 1, sid. 1-20

Olofsson, B. (2012) *Grundvattentillgång för småhusbebyggelse vid Färgårdsviken, Östhammars kommun*. Tillgänglig  
[http://www.osthammar.se/PageFiles/29993/Bilaga\\_1\\_Vattenforsorjning\\_Fargardsviken\\_slutversion.pdf](http://www.osthammar.se/PageFiles/29993/Bilaga_1_Vattenforsorjning_Fargardsviken_slutversion.pdf) (2013-02-20)

OSGeo; The Open Source Geospatial Foundation (2012) *GDAL/OGR Info Sheet*. Tillgänglig: [http://www.osgeo.org/gdal\\_ogr](http://www.osgeo.org/gdal_ogr) (2012-12-19)

Python.org (-) *About*. Tillgänglig <http://python.org/about/> (2013-02-20)

Region Gotland (2012) *Register till miljö- och hälsoskyddsnämndens protokoll vid extra sammanträde den 23 augusti 2012*. Tillgänglig  
<http://www.gotland.se/imcms/MHN120823> (2013-02-20)

SCB (2001) *Öar i Sverige*. Statistiska meddelanden, MI 50 SM 0101. ISSN 1403-8978

Schilling, W. (1991) *Rainfall data for urban hydrology: what do we need?*. Atmospheric Research, 27, sid. 5-21

Shuster, W.D., Bonta, J., Thurston, H., Warnemuende, E., Smith, D.R. (2005) *Impacts of impervious surface on watershed hydrology: A review*. Urban water journal, Vol. 2, utg. 4

SMHI (2010) *Höjdsystem och havsvattenstånd*. Tillgänglig:  
<http://www.smhi.se/kunskapsbanken/oceanografi/hojdsystem-och-havsvattenstand-1.13582> (2013-03-07)

Svenskt Vatten (2004) *Publikation P90: Dimensionering av allmänna avloppsledningar*. ISSN: 1651-4947

Tutorialspoint.com (-) *Python variable types*. Tillgänglig  
[http://www.tutorialspoint.com/python/python\\_variable\\_types.htm](http://www.tutorialspoint.com/python/python_variable_types.htm) (2013-02-20)

## **7.2. MUNTliga REFERENSER**

Ahlquist-Juhlén, Marta (2012) muntligen. Projektledare Ramböll

Hammarlund, Hans (2012-2013) muntligen. Specialist, hydrologisk modellering

Marklund, Lars (2012-2013) muntligen. Tekn Dr Geohydrolog

Näsman Melander, Emelie (2013) muntligen. VA-ingenjör

## APPENDIX A – MATLABKOD FÖR EDITERING AV RASTER

```
% Script to remove gaps of a couple of cells in a raster
% © Karl-Martin Calestam, 2012-10-10 [calestam@gmail.com],
% Uppsala universitet, Uppsala & Tyréns AB, Stockholm

% Read the raster in geotif-format
[Z R bbox] = geotiffread('lid.tif');
data = Z;
clc

% Find the size of the raster
[m n] = size(data);

for i = 2:m-1
    for j = 2:n-1
        if data(i,j) == 255
            data2 = data(i-1:i+1,j-1:j+1);
            t = find(data2==255);
            if numel(t) < 3
                data(i,j) = 0;
            end
        end
    end
end

% Save the output for next iteration (replace Z above)
ext10_1 = data;
% Save data to file
imwrite(data,'ext10_1.tif','tif');
```

## APPENDIX B – MATLABKOD FÖR ATT SÄNKA NED UTLOPPSPUNKTER

```
% Script manipulating a DEM to create sinks at given locations
% specified in another raster with the same extent.
% Script by Karl-Martin Calestam, 2012-10-15 [calestam@gmail.com]
% Uppsala universitet, Uppsala & Tyréns AB, Stockholm

% Clear command window, clear workspace, close windows
clc, clear all, close all

%% Import data
% Load files
infile = 'lid.tif';
sinkfile = 'outlets.tif';
outfile = 'lid_ed6b.tif';
[Z1 R_dem bbox_dem] = geotiffread(infile); % DEM for the area
[Z2 R_out bbox_out] = geotiffread(sinkfile); % Outlet points in raster
format % with same extent as the
DEM.

dem = double(Z1);
outlets = double(Z2);
clear Z1 Z2 % Clearing to free up memory space
% For this specific case:
% Outlets have NoData value = 255. 88 non-nodata entries
% The DEM have NoData value ~= -3.4028235e+038.

nodata_dem = dem(1,1);
nodata_out = outlets(1,1);

% Finding the locations and number of the outlets
[n d] = find(outlets ~= nodata_out);
nr_out = length(n);

%% Define area
% Calculate the slope in an external function
% Prompting the number of buffer cells around the sink
cell = input('No. of buffer cells (default 4 cells, max 10 cells): ');
);
intTest=~mod(cell,1); % Testing for integers

if isempty(cell) == 1 || cell < 0 || intTest ~= 1 || cell > 10
    cells = 4;
    display('Out of range or not an integer, using default')
else
    cells = cell;
end

%% Editing
% Calculating a slope based on an exponential function
x_list = 1:0.01:5;
slope = -exp(-x_list)*27.2*2; %Modifies so that the first value is -
10 [m]
len = length(slope);
ssk = [];
```

```

% Evaluating for each of the sinks with the specified number of buffer
cells
% around it.
for m = 1:nr_out
    % Extracting data from the DEM, specified number of cells in each
    % direction starting from the sink cell.
    data = dem(n(m)-cells:n(m)+cells,d(m)-cells:d(m)+cells);
    si = size(data);
    % Identifying the sink cell in data
    n2 = cells+1; d2 = cells+1;
    if data(n2,d2) == nodata_dem
        ssk = [ssk m];
    else
        % Subtracting the slope value from the sink cell
        data(n2,d2) = data(n2,d2)+slope(1);
    end
    % Iterating over all of the extracted cells
    for k1 = 1:si(1)
    for k2 = 1:si(2)
        % Calculating the distance between each individual cell and
the
        % sink cell.
        % Locating the the active cell in data
        n1 = k1; d1 = k2;
        % Using Pythagoras theorem to calculate the distance. Cells
that
        % fall outside of the max distance (specified number of cells)
will
        % not be edited.
        dist = sqrt((n2-n1)^2 + (d2-d1)^2);
        % Retrieving a value from slope that corresponds to the
current
        % distance.
        decr = len * dist/cells;
        % Subtracting the slope value from the raster value
        if decr <= len && decr ~= 0 && data(k1,k2) ~= nodata_dem
            data(k1,k2) = data(k1,k2)+slope(round(decr));
        end
    end
end

    % Saving the edited data to the original DEM
    dem(n(m)-cells:n(m)+cells,d(m)-cells:d(m)+cells) = data;
end
sskn = numel(ssk);
display(['Sink(s) no. ' num2str(ssk(1:floor(sskn/2))])])
display(['num2str(ssk(ceil(sskn/2+1):end)) ' are located in a NaN
cell,'])
display(['which is ' num2str(sskn) ' out of ' num2str(nr_out) '
sinks.'])
display(' ')

% Writing the edited DEM to a file
info = geotiffinfo(infile);
geotiffwrite(outfile, single(dem), R_dem, 'GeoKeyDirectoryTag',
info.GeoTIFFTags.GeoKeyDirectoryTag)

%% Mapping
% Make a surface plot for user specified sink
% Mapshow is for projected data, geoshow for unprojected data
(lat,lon)

```



```

nod = find(dem == nodata_dem);
dem2 = dem;
clear outlets % Clearing memory to free up space for calculations
dem2(nod) = NaN;
clear nod

plott = input(['Input sink ID [1 - ' num2str(nr_out) ' ] to create a
surface plot for the sink. Enter to quit. ']);

% Providing the possibility to create more plots for different sinks
side = 20;
while isempty(plott) == 0
    if plott <= nr_out && plott > 0
        figure(plott)
        r = n(plott)-side:n(plott)+side;
        s = d(plott)-side:d(plott)+side;
        surf(r,s,double(dem2(r,s)))
        caxis auto
        plott = input('For additional plots, input another sink ID.
Enter to quit. ');
    else
        display('Sink ID out of bounds')
        plott = [];
    end
end
end

```

## APPENDIX C - PYTHONSKRIPT FÖR ATT SÄNKA NED UTLOPPSPUNKTER

Till denna kod har inspiration och hjälp hämtats på ett forum för GIS-användning;  
[gis.stackexchange.com](http://gis.stackexchange.com).

```
# -*- coding: cp1252 -*-
# Python script to edit geotiff files
# Modified from different sources by Karl-Martin Calestam, 2012-11-08
# Main sources: *http://bubuitalia.wordpress.com/2009/10/29/python-gdal-simple-file-processing/
#*http://www.gis.usu.edu/~chrisg/python/2009/lectures/ospy\_slides4.pdf
# *http://gis.stackexchange.com/
"""
Insert your DEM raster file in .tif-format.
Input sink file, only point shapefiles are valid inputs.
Output file, might need to be added to your map document manually.
Insert the buffer area around each sink that will be burned down. Only
positive inputs, and in map units.
"""

## 1. Importing modules
# Required libraries: ArcPy, OSGEO: GDAL/OGR, NumPy, Sys, OS, Custom
module arcmapmods
try:
    import os, sys, osgeo, arcpy, math, arcgisscripting
except ImportError:
    sys.exit('Failed to load main modules, check your Python
bindings.')
```

```
try:
    from osgeo import gdal, ogr
    from osgeo.gdalconst import *
    from osgeo import gdal_array
except ImportError:
    import gdal, ogr
    from gdalconst import *
    import gdalnumeric as gdal_array
except:
    sys.exit('Failed to load modules (GDAL/OGR)')
```

```
try:
    import numpy
except ImportError:
    import Numeric as numpy
except:
    sys.exit('Failed to load modules (NumPy)')
```

```
try:
    import arcmapmods as AM
except ImportError:
    sys.exit('Failed to load local modules (ArcMapModules)')
```

```
## -----
## 2. Assigning input parameters and setting the environment
# Reading input from user
inDEM = arcpy.GetParameterAsText(0)
Outlets = arcpy.GetParameterAsText(1)
outDEM = arcpy.GetParameterAsText(2)
buffr = arcpy.GetParameterAsText(3)
```

```

if buffr[0] > 0:
    area = int(buffr[0:2])
else:
    area = 4 # Setting to default value if non-positive input

# Set working directory to current directory
old_dir = os.getcwd()
try:
    os.mkdir(old_dir + '/Temp')
except WindowsError:
    r = 1; s = 1
    while r == 1 and s < 50:
        try:
            os.mkdir(old_dir + '/Temp' + str(s))
            r = 0
        except:
            s = s+1
            if s == 50:
                sys.exit('You have ran this tool more than 50 times in
the same folder, change location')
            pass
    del r, s

##directory = os.path.dirname(os.path.realpath(inDEM))
os.chdir(old_dir + '/Temp')
arcpy.env.workspace = old_dir + '/Temp'

## -----
## 3. Open the DEM
# Register GDAL drivers
gdal.AllRegister()

# Open DEM and get metadata
dem = gdal.Open(inDEM, GA_ReadOnly)
if dem is not None:
    cols = dem.RasterXSize
    rows = dem.RasterYSize
    desc = arcpy.Describe(inDEM)
    xmin = desc.extent.xmin
    ymin = desc.extent.ymin
    cellw = desc.meanCellWidth
    cellh = desc.meanCellHeight
    nodata = desc.noDataValue
    sr = desc.SpatialReference
else:
    sys.exit('Failed to open dataset ' + inDEM)

if cellw == cellh:
    cellsz = cellw
else:
    sys.exit('Cell size in input raster is not square, e.g. differs in
x and y direction.')

## -----
## 4. Converting sink shapefile to raster with same extent as inDEM
arcpy.env.overwriteOutput = True
arcpy.env.extent = inDEM
AM.SendMsg('eE')
try:
    outlet_raster = 'med_raster'

```

```

    arcpy.PointToRaster_conversion(Outlets, "", outlet_raster,
"MAXIMUM", "", cellsz)
except arcpyscripting.ExecuteError:
    r = 1; s = 1
    while r == 1 and s < 50:
        try:
            outlet_raster = 'med_raster' + str(s)
            arcpy.PointToRaster_conversion(Outlets, "", outlet_raster,
"MAXIMUM", "", cellsz)
            r = 0
        except:
            s = s+1
            if s == 50:
                sys.exit('You have ran this tool more than 50 times in
the same folder, change location')
            pass
arcpy.env.overwriteOutput = False
AM.SendMsg('2hn')
msg4 = 'PointToRaster successfully executed, ' + outlet_raster + '
created.'
AM.SendMsg(msg4)

## -----
## 5. Create slope function
slope = []
x_list = numpy.arange(0,6,.01)
for i in x_list:
    slope.append(-math.exp(-i)*30)
length = len(slope)
AM.SendMsg('3')

## -----
## 6. Edit DEM at the location of all the sinks
# Reading outlet raster as array
outl_arr = arcpy.RasterToNumPyArray(outlet_raster, "", cols, rows,
None)
nodata_o = outl_arr[0,0]
# Locating the positions of the outlets in the raster
f = numpy.argwhere(outl_arr != nodata_o)
sink_amount = len(f)
msg7 = str(sink_amount) + ' sink points have been detected.'
AM.SendMsg(msg7)
outl_arr = None
# Reading the DEM as array
try:
    dem_arr = arcpy.RasterToNumPyArray(inDEM, "", cols, rows, None)
except MemoryError:
    dem = None
    try:
        dem_arr = arcpy.RasterToNumPyArray(inDEM, "", cols, rows,
None)
    except MemoryError:
        sys.exit('Memory limit reached, try restarting ArcMap, cut
your raster in smaller parts or use better computer')

# area = input buffer area in meters
# cellsz = cellsize in meters
buffers = area/cellsz
non = []

for i in range(sink_amount):

```

```

# Extract part to investigate
index = f[i]
if dem_arr[index[0], index[1]] != nodata:
    r_low = index[0] - buffers
    r_up = index[0] + buffers + 1
    c_low = index[1] - buffers
    c_up = index[1] + buffers + 1
    data = dem_arr[r_low:r_up, c_low:c_up]
    dim = math.sqrt(numpy.size(data))
    n1 = math.floor(dim/2)
    for j1 in range(dim):
        for j2 in range(dim):
            n2 = data[j1,j2]
            dist = math.sqrt((n1-j1)**2 + (n1-j2)**2)*cellsz
            decr = length * dist / area
            if decr < length and decr != 0 and n2 != nodata:
                # Edit the cells that are non-nodata
                data[j1,j2] = n2 + slope[int(math.floor(decr))]
            elif decr == 0:
                data[j1,j2] = n2 + slope[0]
            elif decr == length:
                data[j1,j2] = n2 + slope[-1]
    # Write the edits back to original array
    dem_arr[r_low:r_up, c_low:c_up] = data
else:
    non.append(f[i])

sinks = sink_amount - len(non)
msg8 = str(len(non)) + ' sinks are located in a NoData area and were
excluded.\nA total of ' + str(sinks) + ' have been used.'
AM.SendMsg(msg8)

## -----
## 7. Save array to raster
newraster = arcpy.NumPyArrayToRaster(dem_arr, arcpy.Point(xmin, ymin),
cellw, cellh, nodata)
arcpy.DefineProjection_management(newraster, sr)
# Save new file
try:
    newraster.save(outDEM)
except RuntimeError:
    k = 1; m = 1
    while k == 1 and m < 20:
        try:
            newraster.save(outDEM + '_' + str(m))
            k = 0
        except RuntimeError:
            m = m+1
            if m == 20:
                sys.exit('Filename for output file already used max
number of times')

## -----
## 8. Close files
dem = None; outlet_raster = None; dem_arr = None; outlet_rast = None
outl_arr = None; newraster = None

os.chdir(old_dir)
arcpy.env.workspace = old_dir
AM.SendMsg('Tool successfully completed')

```

## APPENDIX D – PYTHONKOD FÖR ATT LÄGGA IHOP AVRINNINGSSOMRÅDEN MED EN VISS STORLEK

Till denna kod har inspiration och hjälp hämtats på ett forum för GIS-användning;  
[gis.stackexchange.com](http://gis.stackexchange.com).

```
# -*- coding: cp1252 -*-
# Python script to edit geotiff files
# Modified from different sources by Karl-Martin Calestam, 2012-11-18

## 1. Importing modules
# Required libraries: ArcPy, NumPy, Sys, OS, Custom module arcmapmods
try:
    import os, sys, arcpy, math, arcgisscripting#,osgeo
except ImportError:
    sys.exit('Failed to load main modules, check your Python
bindings.')
```

```
try:
    import numpy
except ImportError:
    import Numeric as numpy
except:
    sys.exit('Failed to load modules (NumPy)')
```

```
try:
    import arcmapmods as AM
except ImportError:
    sys.exit('Failed to load local modules (ArcMapModules)')
```

```
## -----
## 2. Assigning input parameters and setting the environment
# Reading input from user
inpgs    = arcpy.GetParameterAsText(0)
indrln   = arcpy.GetParameterAsText(1)
buffr    = arcpy.GetParameterAsText(2)
outpgs   = arcpy.GetParameterAsText(3)
cellsz   = 2; AM.SendMsg('Fix cellsize management',1)
```

```
# Set working directory to current directory
old_dir  = os.getcwd()
directory = os.path.dirname(os.path.realpath(inpgs))
os.chdir(old_dir)
arcpy.env.workspace = old_dir
AM.SendMsg('Import and workspace set, fix paths',1)
```

```
## -----
## 3. Read in the polygon attribute table and chose appropriate ones
AM.SendMsg('Unit handling in buffer area', 1)
wherel = '"' + 'Shape_Area' + '" < ' + str(buffr)
```

```
# Create dictionary for all geometries
polyrows = arcpy.SearchCursor(inpgs)
geometryDictionary = {}

for prow in polyrows:
```

```

        geometryDictionary[prow.GridID] = prow.Shape
del prow, polyrows

# Create dictionary which contains the GridIDs of the downstream
catchment area
nextrows = arcpy.SearchCursor(indrln)
nextDownDictionary = {}
nDD = {}

for nrow in nextrows:
    nextDownDictionary[nrow.GridId] = nrow.NextDownID

for k in dict.keys((nextDownDictionary)):
    clause = '"' + 'HydroID' + '" = ' + str(nextDownDictionary[k])
    nr = arcpy.SearchCursor(indrln, clause)
    for n in nr:
        nDD[k] = n.GridID
del n, nr

polyrows = arcpy.SearchCursor(inpgs, where1)
tempMerged = r'C:\GIS\New_data\tempMerged.shp'
tempPgs = r'C:\GIS\New_data\tempPgs.shp'
AM.SendMsg('Make dynamic paths', 1)
arcpy.env.overwriteOutput = True
arcpy.Copy_management(inpgs, outpgs)
arcpy.env.overwriteOutput = False
grids = []

# grd1 = GridID on the current polygon
# grd2 = GridID on the polygon downstream
for prow in polyrows:
    grd1 = prow.GridID
    grd2 = nDD.get(grd1)
    AM.SendMsg( 'grd2 ' + str(grd2) + ' <--' )
    if grd2:

        geometry1 = prow.Shape
        geometry2 = geometryDictionary[grd2]

        arcpy.env.overwriteOutput = True
        AM.SendMsg( 'grd1 ' + str(grd1))

        arcpy.Merge_management([geometry1, geometry2], tempMerged)
        arcpy.Dissolve_management(tempMerged, tempPgs)
        arcpy.env.overwriteOutput = False
        arcpy.Append_management(tempPgs, outpgs, "NO_TEST")
        grids.append(prow.GridID)

if grids:
    for g in grids:
        whereU = '"GridID" = ' + str(g)
        upd = arcpy.UpdateCursor(outpgs, whereU)
        for up in upd:
            upd.deleteRow(up)

del prow, polyrows
del up, upd
AM.SendMsg('Done!')
```

## APPENDIX E – INSTRUKTIONER: INSTALLING GDAL AND OGR FOR PYTHON ON WINDOWS

### Overview:

1. *Get the GDAL core files*
2. *Get the Python bindings*
3. *Install GDAL*
4. *Edit Environment Variables*
5. *Install Python bindings*

Note that you will need administrator rights on your computer.

### 1. Get the GDAL core files

This scary looking site: <http://www.gisinternals.com/sdk/> is where you download GDAL. Scroll down to the bottom of the top-most table and click on the relevant link in the 'Downloads' column – either 32 bit (second to last row) or 64 bit (last row).

From the new page download the GDAL Core file: `gdal-[version]-[build]-core.msi` (e.g. `gdal-18-1600-core.msi`)

### 2. Get the Python Bindings

Fortunately the Python bindings are located right here as well, called `GDAL-[gdalVersion.system]-py[pythonVersion].[exe/msi]` (e.g. `GDAL-1.8.0.win32-py2.6.exe`), just make sure to download the version matching your installed version of Python.

### 3. Install GDAL

Simply run the GDAL Core installer downloaded in step 1, and note the install path if you change it.

### 4. Edit Environmental Variables

WARNING: Don't make any mistakes here (like deleting things), or your system might not work...

To bring up the Environment Variables dialogue in Windows XP:

Right click on My Computer, select Properties, go to the Advanced tab, then click Environment Variables



On Windows 7:

Right click on Computer, click on Advanced system settings on the left, then click Environment Variables

In the 'System variables' pane, scroll the box down and select the 'Path' variable, then go to Edit. Place the cursor in the box, and hit End to get the cursor at the end, add a semi-colon (;) and then enter the path to your GDAL installation – so, for my installation, I added this:

```
;C:\Program Files (x86)\GDAL
```

Just be careful not to insert any additional spaces before or after the path (see comments)! Also, add the path somewhere in the beginning, or else it might cause some conflict with existing paths and not work properly. Click Ok, then click New and enter the following:

Variable name: GDAL\_DATA

Variable value is the path to the GDAL data directory – so for me:

```
C:\Program Files (x86)\GDAL\gdal-data
```

To test the installation, open up your command prompt (type cmd in the Run dialogue), type `ogr2ogr` and hit enter. This is one of the programs included with GDAL; if your Path is set correctly it will be accessible from anywhere. If some stuff about usage and options is printed out, GDAL is installed correctly and the Environment variables worked. If it instead says: `'ogr2ogr' is not recognized as an internal or external command, operable program or batch file.`, there is possibly a problem. First off try restarting the computer, if `ogr2ogr` still doesn't work check your Environment variables for spelling mistakes or typos.

## 5. Install Python bindings

Run the Python bindings installer downloaded in step 2.

To test the entire setup, open a Python prompt and enter:

```
from osgeo import ogr
```

Hit enter, then:

```
from osgeo import gdal
```

If both of these run without printing out anything, you are good to go!

Stacy Rendall; <http://pythongisandstuff.wordpress.com/2011/07/07/installing-gdal-and-ogr-for-python-on-windows/>