UPTEC W 24041
Examensarbete 30 hp
Juni 2024

UPPSALA UNIVERSITET

Vattenbalansmodellering för en gråbergsdeponi i subarktiskt klimat

En jämförelse mellan konceptuell och fysikalisk modell

Adrian Emilson

Civilingenjörsprogrammet i miljö- och vattenteknik

UPPSALA UNIVERSITET

Modeling Water Balance for a Waste Rock Dump in a Subarctic Climate: A Comparison between a Conceptual and a Physical Model

Adrian Emilson

Abstract

During the extraction of ore, waste rock from the excavation of the mine is placed in large dumps, known as waste rock dumps. Waste rock dumps pose significant environmental challenges due to the potential production of leachate containing high concentrations of nitrogen, metals, and other contaminants. Understanding the hydrological dynamics within these dumps is crucial for assessing and mitigating the environmental impacts on nearby recipients. This study aimed to model the water balance at the surface of one of LKAB's waste rock dumps and thereby estimate percolation rates using two different approaches: a physically-based model and a conceptual model. Previously collected meteorological data were used for simulations of a one-dimensional, physically-based model that integrates water and heat flow, and a one-dimensional conceptual model inspired by HBV. Results showed that the conceptual model could reproduce seasonal variations in observed soil moisture but had difficulties in reproducing soil moisture during early summer and daily variations. The physically-based model relatively well reproduced soil moisture and showed good agreement with observed soil temperature. Percolation flow as a percentage of precipitation was 45% for the physically-based model and 67% for the conceptual model, with the main discrepancies arising from evaporation losses and the ability to simulate freeze-thaw dynamics.

Keywords: Water balance modeling, water balance, waste rock, CoupModel, subarctic climate, physical based model, conceptual model

Department of Earth Sciences; Air, Water and Landscape Science; Hydrology Uppsala University (UU), Villavägen 16, SE-752 36, Uppsala, Sweden.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala

Handledare: Thomas Grabs Ämnesgranskare: Roger Herbert

Examinator: Fritjof Fagerlund

Referat

Modellering av vattenbalans vid ytan av en gråbergsdeponi i subarktiskt klimat – en jämförelse mellan konceptuell modell och fysikalisk modell

Adrian Emilson

Vid utvinning av malm läggs bergrester utan malm på stora deponier, så kallade gråbergsdeponier. Gråbergsdeponier utgör betydande miljömässiga utmaningar på grund av dess potential att producera lakvatten innehållande höga koncentrationer av kväve, metaller och andra föroreningar. Att förstå de hydrologiska processer som styr lakvattenbildning i gråbergsdeponierna är avgörande för att bedöma och mildra miljöpåverkan från lakvattenutsläpp på närliggande recipienter. Syftet med studien var att modellera vattenbalansen vid ytan av en av LKAB:s gråbergsdeponier och därigenom uppskatta hur mycket vatten som perkolerar ner i gråbergdeponin med hjälp av två olika metoder: en fysikalisk modell och en konceptuell modell. Tidigare insamlade meteorologiska data användes för simuleringar av en endimensionell fysikalisk modell som integrerar vatten- och värmeflöde, och en endimensionell konceptuell modell inspirerad av HBV. Resultatet visade att den konceptuella modellen kunde återge årstidsvariationer i observerad markfuktighet, men hade svårigheter i att återge markfuktigheten under tidig sommar och dagliga variationer. Den fysikaliska modellen kunde återge markfuktighet någorlunda väl och visade en god anpassning till observerad marktemperatur. Perkolationsflödet som en procentandel av nederbörd var 45% för den fysikaliska modellen och 67% för den konceptuella modellen, med huvudsakliga skillnader som uppstår från avdunstningsförluster och förmågan att simulera frysnings- och upptiningsdynamik.

Nyckelord: Vattenbalansmodellering, vattenbalans, gråbergsdeponi, CoupModel, subarktiskt klimat, konceptuell modell, fysikalisk modell

Institutionen för geovetenskaper; luft-, vatten- och landskapslära; hydrologi, Uppsala universitet, Villavägen 16, SE-752 36 Uppsala, Sverige.

Förord

Jag vill rikta ett stort tack till min handledare, Thomas Grabs, både för möjligheten att utföra detta examensarbete och för den hjälp och de många diskussioner vi haft under arbetets gång. Jag vill även tacka min ämnesgranskare, Roger Herbert, som förutom att hjälpa till med rapporten även bidragit med värdefull information. Ett särskilt tack riktas till Per-Erik Jansson, utvecklare av CoupModel, för din ovärderliga hjälp med att ställa in modellen och för dina förklaringar av hydrologiska processer.

Ett ytterligare tack går till Thomas Grabs för hans tillåtelse att reproducera figurer i min rapport. Jag vill även rikta ett tack till NITREM och LKAB för deras bidrag i konstruktionen och underhållet av den meteorologiska mätstation som jag använt data från i min studie.

Adrian Emilson

Uppsala, juni 2024

Copyright © Adrian Emilson, Institutionen för geovetenskaper; luft-, vatten- och landskapslära; hydrologi.
UPTEC W 24041, ISSN 1401-5765
Digitalt publicerad i DiVA, 2024, genom institutionen för geovetenskaper, Uppsala universitet.
(http://www.diva-portal.org/)

Populärvetenskaplig sammanfattning

Vid brytning av malm deponeras icke-malmhaltigt berg på stora områden, kända som gråbergs- eller sidobergsdeponier. Genom att vatten tränger igenom dessa deponier uppstår en betydande miljöutmaning på grund av den potentiella bildningen av lakvatten, som kan innehålla höga halter kväve, metaller och andra föroreningar. För att kunna bedöma och minska miljöpåverkan från lakvattenutsläpp på närliggande vattendrag är det avgörande att förstå de vattenrelaterade processer som styr bildningen av lakvatten i gråbergsdeponierna.

Syftet med denna studie var att modellera vattenbalansen, alltså flödena av vatten, vid ytan av en gråbergsdeponi för att uppskatta mängden vatten som tränger ner i deponin. Två olika modeller jämfördes för att bedöma deras förmåga att beskriva vattenbalansen: en mer avancerad fysikalisk modell och en enklare konceptuell modell.

Platsen för studien var en av LKAB:s gråbergsdeponier, Triangeldeponin. På Triangeldeponin fanns en meteorologisk mätstation som mätte en mängd olika meteorologiska variabler under ett år och åtta månader. Den meteorologiska stationen hade problem vid många tillfällen vilket ledde till att flertalet variabler inte registrerades under långa perioder. För att modellerna skulle kunna köras behövdes kontinuerliga indata. Detta innebar att en dataanalys behövde utföras för att fylla igen luckor i data. Luckorna fylldes igen med hjälp av data från SMHI:s närliggande mätstationer. Utöver luckorna i data fanns det problem med att observerade snödjup, snömängd och nederbörd inte alltid stämde bra överens, vilket försvårade modelleringen.

Den konceptuella modellen skapades under denna studie i programmeringsspråket Python, där inspirationer togs från ett annat program som modellerar vattenrelaterade processer. Den konceptuella modellen är enklare och kräver färre indata. Modellen använder enkla ekvationer för att uppskatta vattenflöden och fokuserar på långsiktiga trender snarare än dagliga variationer.

Som fysikalisk modell valdes programmet CoupModel, vilket används för att skapa detaljerade modeller för att simulera vatten- och värmeflöden. CoupModel använder avancerade ekvationer och tar hänsyn till flera faktorer som påverkar vattenflödet, såsom markens frysning och upptining. Båda modeller anpassades till de lokala förhållandena.

Resultaten visade att den konceptuella modellen kunde återge årstidsvariationer i observerad markfuktighet, men hade svårigheter att återge markfuktigheten under tidig sommar och dagliga variationer. Den fysikaliska modellen kunde återge markfuktighet någorlunda väl och visade en god anpassning till observerad marktemperatur. Den fysikaliska modellen ger mer detaljerade beskrivningar av vattenflödena, medan den konceptuella modellen är enklare att använda och anpassa till olika förhållanden.

Modellerna visar att andelen av nederbörden som tränger ner i deponin var 45 % för den fysikaliska modellen och 67 % för den konceptuella modellen. Skillnaderna beror främst på förluster genom avdunstning hos de båda modellerna. Det finns en del osäkerheter kring avdunstningens faktiska storlek, där underskattning av markfuktighet, särskilt i den konceptuella modellen, tyder på att avdunstningen underskattats. Samtidigt finns det andra argument som tyder på motsatsen.

Under arbetet studerades olika beräkningsmetoder för att uppskatta potentiell avdunstning, alltså hur mycket vatten som potentiellt skulle avdunsta från en plats om tillräckligt mycket vatten fanns tillgängligt. Mer avancerade beräkningsmetoder som Penman-Monteith kan användas när data finns tillgänglig, men en enklare beräkningsmetod som kräver att färre variabler mäts kan vara fördelaktigt att använda i vissa situationer. Hargreaves beräkningsmetod visade sig kunna uppskatta ett trovärdigt värde på potentiell evaporation, där endast maximal och minimal daglig temperatur används.

För att förbättra den konceptuella modellen kan framtida arbete fokusera på att utveckla hur snödynamiken hanteras. Därutöver skulle en integration av marktemperaturdata kunna förbättra modellens noggrannhet då modellen skulle kunna ta hänsyn till när marken fryser och tinar upp.

Direkta mätningar av perkolation och avdunstning skulle underlätta modelleringen och minska de osäkerheter som finns vid kalibrering av modellerna.

Ord- och symbollista

CoupModel - Ett modelleringsverktyg för att simulera mark-växt-atmosfärsystemet.

FC – Fältkapacitet. Den mängd vatten som finns kvar i marken efter att överskottsvattnet dränerats bort.

FE – Faktisk evaporation. Vatten som faktiskt avdunstar från marken.

G – Markvärmeflöde.

h – Tryckpotentialen.

h_a – Luftinträdestrycket.

H – Totala hydrauliska. Summan av tryckpotentialen och lägespotentialen.

HBV – Hydrologiska Byråns Vattenbalansavdelning. Hydrologisk modell för beräkning av avrinning.

I – Infiltration. Vatten som tränger ner i marken från nederbörd eller snösmältning.

k_{mat} – Matriskonduktivitet. Den hydrauliska konduktiviteten i markens matris.

k_{sat} – Totalkonduktivitet. Den hydrauliska konduktiviteten när markprofilen är mättad.

L - Latent värme. Den energi som krävs för att omvandla vatten från flytande till gasform.

PE - Potentiell evaporation. Den mängd vatten som skulle avdunsta om det fanns en obegränsad tillgång till vatten.

Q – Markvattenflödet. Flödet av vatten i marken. Alternativa benämningar: djup perkolation, grundvattenbildning.

R_n – Netto strålning. Den totala energimängden som marken tar emot minus den energi som reflekteras tillbaka till atmosfären.

S_e – Effektiv mättnad.

S_{sm} – Markfuktighet, absolut värde. Den mängd vatten som lagras i marken.

TDR – *Time Domain Reflectometry*. Metod för att mäta markfuktighet och marktemperatur.

ΔS – Förändring i lagringsterm. Den förändringen i vattenlagring som skett över tid. Kan även benämnas $\Delta S/\Delta t$.

θ – Markvattenhalt. Markvattenhalten används synonymt med markfuktigheten när den avser andel av den totala volymen och inte det absoluta värdet.

θ_m – Makroporositet. Den mängd porutrymme som utgörs av makroporer.

θ_s – Totalporositet. Den totala mängden porutrymme i marken, alternativt vattenhalten vid totalmättnad.

λ – en anpassningsparameter relaterade till porstorleksfördelningen i marken.

Innehållsförteckning

1 Introduktion 1
1.1 Tidigare studier 1
1.2 Syfte 2
1.3 Frågeställningar 2
2 Bakgrund 3
2.1 Energibalans 3
2.1.1 Evaporation 5
2.1.2 Värmeflöde genom marken 6
2.2 Hydrologiska processer som styr vattenbalansen på ytan av en gråbergsdeponi 7
2.2.1 Vattenbalans 7
2.2.2 Snö och snösmältning 8
2.2.3 Vattenflöde genom marken 9
3 Platsbeskrivning och tidigare fältresultat 11
3.1 Meteorologisk station 13
3.2 Tidigare studier vid LKAB:s järnmalmsgruva i Kiruna 13
4 Metod 14
4.1 Dataanalys 15
4.1.1 Fylla igen dataluckor 15
4.1.2 Maximum- och minimumvärden 17
4.1.3 Nederbördskorrigering 17
4.1.4 Snödjupskorrigering 17
4.2 Potentiell evaporation 17
4.3 Konceptuell modell 19
4.3.1 Snörutin 19
4.3.2 Markrutin 20
4.3.3 Kalibrering 22
4.4 Fysikalisk modell 23
4.4.1 Programbeskrivning 23
4.4.2 Arbetsgång 24
5 Resultat 28
5.1 Beräkning av potentiell evaporation 28
5.2 Nederbörd och infiltration 31
5.3 Markfuktighet och marktemperatur 33
5.4 Perkolation och faktisk evaporation 38
6 Diskussion 41
6.1 Potentiell evaporation 41

6.2 Jämförelse av den konceptuella modellen mot den fysikaliska modellen 42
6.2.1 Nederbörd och infiltration 42
6.2.2 Markfuktighet och marktemperatur 43
6.2.3 Perkolation och faktisk evaporation 44
6.2.4 Allmän jämförelse och tidigare studier 46
6.3 Osäkerheter 47
6.3.1 Konceptuell modell 47
6.3.2 Fysikalisk modell 47
6.3.3 Allmänna osäkerheter 48
6.4 Framtidsutsikter och utvecklingsmöjligheter 49
7 Slutsats 50
8 Källförteckning 51
9 Bilagor 54

1 Introduktion

Behovet av metaller och mineraler ökar, drivet av elektrifiering, förnybar energi och digitalisering, därför har efterfrågan på tillgång till dessa råvaror blivit alltmer central. Europa, som en gång var den främsta regionen för gruvproduktion under 1800-talet, har gradvis minskat sin betydelse och står idag bara för en liten del av den globala gruvproduktionen. För att öka sin självförsörjning och minska beroendet av länder som Kina har EU antagit Critical Raw Materials Act. Sverige, med sina stora mineraltillgångar och flera nya fyndigheter, kan därför vara en viktig del av att säkra den globala leveranskedjan av både traditionella och kritiska metaller. Utöver detta erbjuder gruvindustrin betydande ekonomiska möjligheter (Fernandez 2023; Byman 2024).

Samtidigt måste gruvindustrin sträva efter att minimera sin miljöpåverkan för att uppfylla de lagkrav som finns (SGU 2020). En av de centrala utmaningarna är hur gråberg hanteras. Gråberg är det icke malmrika berg som bryts för att komma åt malmen, som efter brytning läggs på deponi och bildar då så kallade gråbergsdeponier (alternativt sidoberg) (Byman 2024). Deponierna är ofta stora, och till en början, omättade högar som förvaras på marken nära intill gruvan (Amos et al. 2015).

Med tiden utsätts gråbergsdeponierna för det lokala klimatet där regn- och snövatten tillåts infiltrera och därefter perkolera ner genom berget. Under denna process tar vattnet upp olika ämnen och bildar lakvatten. Vilka ämnen som förs med lakvattnet beror på vad det är för typ av restmaterial. Om gråberget består av järnsulfider kan dessa oxidera, vilket kan leda till ett lakvatten med lågt pH, så kallat surt lakvatten (*acid mine drainage*). I många fall leder inte reaktionerna i deponierna till att det bildas surt lakvatten, men det kan fortfarande resultera i att höga koncentrationer av metaller och sulfat följer med lakvattnet ut, så kallat neutralt lakvatten (*neutral mine drainage*) (Amos et al. 2015). Rester från sprängmedel som använts vid sprängning följer ofta med gråberget till deponin och kan leda till ett lakvatten med högt innehåll av kväveföreningar, framförallt nitrat och ammonium som är beståndsdelar till det vanligaste sprängmedlet som används idag (Hendry et al. 2018).

Från gråbergsdeponierna rinner lakvattnet ut till omkringliggande recipienter. Där kan lakvattnet med höga kvävehalter leda till eutrofiering och höga halter tungmetaller kan leda till toxiska miljöer för de organismer som befinner sig i eller runt omkring recipienterna (Amos et al. 2015; Hendry et al. 2018). Det blir därför viktigt för gruvindustrin att undersöka hur påverkan på omgivningen kan minskas för en hållbar gruvindustri som kan fortsätta lång tid framöver.

1.1 Tidigare studier

Många studier har undersökt vattenflöde och ämnestransport i gråbergsdeponier för att förstå och minimera miljökonsekvenserna av lakvattenbildning. Flertalet av dessa fokuserar främst på ämnestransport och kemiska reaktioner som sker i gråbergsdeponierna (se till exempel Linklater et al. 2005; Demers et al. 2013; St-Arnault et al. 2019; Seigneur et al. 2021) eller påverkan av olika jordtäcken för minimering av lakvattenbildning från gråbergsdeponier (se till exempel Hendry et al. 2018; Ramasamy & Power 2019; Ma et al. 2020). Det finns också studier som fokuserat på vattenbalansen vid gråbergsdeponier utan jordtäcke, som Neuner et al. (2013) har gjort vid diamantgruvan i Diavik, Kanada. Neuner et al. (2013) undersökte omättat vattenflöde och dess inverkan på lakvattenbildningen från gråbergsdeponier i ett område med kontinuerlig permafrost. Deras experiment inkluderade uppsättningen av två testhögar med insamlingssystem för att studera infiltration och vattenflöde.

Villeneuve et al. (2017) undersökte nettoperkolation, som en rest av vattenbalansen (se avsnitt 2.2.1), och utsläppet av tre lösta ämnen med hjälp av ett dräneringsdike vid foten av en gråbergsdeponi från en kolgruva i Elk Valley, British Columbia. De fann att nettoperkolationen i deponin är högre än i det omkringliggande naturliga avrinningsområdet.

Förekomsten av preferentiellt flöde (se avsnitt 2.2.3.3) beror på flera faktorer, inklusive infiltrationsintensitet, lufttemperatur och materialegenskaper (Blackmore et al. 2014; Amos et al. 2015). För att ta hänsyn till olika flödesregimer modellerade Blackmore et al. (2014) vattenflöde och ämnestransport med Hydrus-1D, med en mobil-immobil modell med dubbel porositet vid gruvan i Antamina, Peru. De fann att den implementerade modellen väl simulerade observerade vattenflöde och ämnestransport. Vidare fann de att närvaron av ”snabba flödesvägar” (*rapid flow paths*) var begränsad (0,1 %). Keller et al. (2015) använde MACRO 5.1, en dubbelpermeabilitetsmodell, för att förutsäga omättat flöde i en planerad gråbergsanläggning i nordvästra Pakistan och fann att den stora evaporationen från platsen ledde till minimal djup perkolation och en lång fördröjning från det att gråbergsdeponin anläggs till dess att utflödet från deponin påbörjas.

De unika utmaningarna med att modellera hydrologiska processer i gråbergsdeponier i subarktiska klimat inkluderar årlig temperaturvariation, snöfall, frysning och upptining av marken, samt vindens påverkan på snödynamik. Att modellera dessa processer med data endast samlad vid ytan är ovanligt men nödvändigt för att förstå klimatets påverkan på lakvattenbildning.

Ett tidigare försök (Atmosudirdjo 2019) att modellera vattenbalansen gjordes i samma område som den gråbergsdeponin som undersöks i denna studie återfinns. Från två småskaliga gråbergsdeponier, konstruerade 2014, mättes ett lakvattenflöde under två år. Flödet ut från de småskaliga gråbergsdeponierna var betydligt mindre 2015 än 2016, vilket indikerade en ackumulering av vatten i det torra gråberget innan större flöden kunde lakas ut. Simuleringen gjordes med Hydrus-1D, där det menades på att det inte fanns några indikationer om att preferentiellt flöde skulle vara ett signifikant flöde. Studien hade dock svårt att återskapa de observerade flödena på grund av problem med tids- och rumsupplösning samt svårigheter att modellera marktemperaturen. En rekommendation var att använda en annan modelleringsstrategi som bättre anpassar sig till frysnings- och upptiningsprocesserna (ibid.).

Att modellera de hydrologiska processerna som styr vattenflödet genom gråbergsdeponier är avgörande för att minska miljökonsekvenserna av lakvattenbildning. Trots tidigare studier finns det en tydlig kunskapslucka när det gäller modellering av vattenbalans i subarktiska klimatförhållanden. Årliga variationer i temperatur, snödynamik och markens frysning och upptining skapar unika utmaningar som inte fullt ut har hanterats i befintlig forskning. Genom att utveckla modeller som bättre anpassar sig till dessa förhållanden kan vi förbättra förståelsen av klimatets påverkan på lakvattenbildning och därmed bidra till att minska gruvdriftens miljöpåverkan.

1.2 Syfte

Syftet med denna studie var att genom hydrologisk modellering av vattenbalansen vid ytan på gråbergsdeponin ”Triangeldeponin” kvantifiera mängden vatten som infiltrerar och vidare perkolerar genom deponin och under vilken del av året detta sker. Vidare var syftet att jämföra en konceptuell modell och en fysikalisk modell.

1.3 Frågeställningar

- Vilken enkel beräkningsmetod kan användas för att estimera potentiell evaporation i gråbergsdeponin?
- Hur varierar mängden vatten som perkolerar och evaporerar under årets olika perioder?
- Jämfört med en avancerad fysikalisk modell, hur väl står sig en simplare konceptuell modell när det gäller att beskriva vattenbalansen vid ytan av gråbergsdeponin?

2 Bakgrund

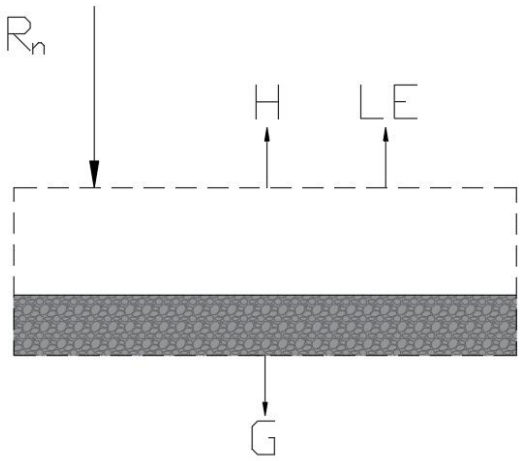
Detta avsnitt syftar till att ge en bakgrund till de koncept och processer som är relevanta för att uppfylla ovanstående syfte och frågeställningar. Energibalansen är ett grundläggande koncept som används i den fysikaliska modellen och ligger även till grund för att kvantifiera potentiell evaporation, vilket beskrivs vidare i avsnittet om evaporation. Några exempel på ekvationer för beräkning av potentiell evaporation presenteras för att ge läsaren förståelse hur sådana ekvationer kan formuleras. Vatten- och värmeflöde i marken är centrala koncept, där vattenflödet är det mest centrala för hela studien. Dessa koncept relaterar till att beskriva hur vattnet rör sig genom gråbergsdeponin. Kunskaper om vattenflöde används direkt i både den konceptuella och den fysikaliska modellen, medan värmeflödet används direkt i den fysikaliska modellen och har en indirekt påverkan i den konceptuella modellen. Eftersom den undersökta deponin är snötäckt under delar av året, är snödynamik viktig för modelleringen av vattenbalansen.

2.1 Energibalans

Termodynamikens första huvudsats säger att energi inte kan skapas eller förstöras utan kan endast omvandlas från en form till en annan. Det är bevarandet av energi som energibalansen bygger på, detta inom en viss volym (eller avgränsat system), där den totala energin är densamma men kan fördelas mellan olika energikällor (Nationalencyklopedin, u.å.a). Energibalansens olika delar för ett mark-atmosfärssystem kan beskrivas med ekvationen

$$R_n = G + H + L \cdot E \tag{11}$$

där R_n är nettostrålningen som kommer till markytan från solen. G är värmeflödet från markytan ner i marken. H är det sensibla värmeflödet, vilket uppstår på grund av temperaturskillnader mellan luft och markytan. $L{\cdot}E$ är det latenta värmeflödet, vilket uppstår på grund av vatten som evaporerar eller kondenserar, där L är ångbildningsvärme och E är evaporationshastigheten. Termerna i energibalansen är ofta uttryckta i enheten W/m², energi per tidsenhet per ytenhet (Rodhe 2000). Energibalansen för ett mark-atmosfärssystem illustreras i Figur 1. Om R_n är positiv innebär det att (strålnings)energi tillförs till systemet (markytan). Den tillförda energin kan fördelas mellan att värma upp marken (positivt G), värma upp atmosfären (positivt H) och att evaporera vatten (positivt $L \cdot E$). Fördelningen av energi mellan de olika komponenterna beror på markytans temperatur, lufttemperatur, fuktighet och vattentillgänglighet i marken (Rodhe 2000).



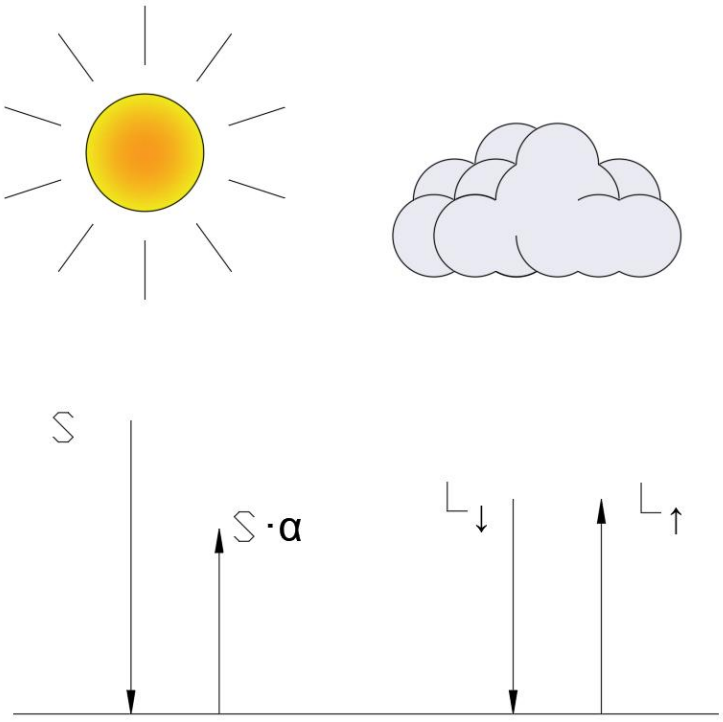
Figur 1. Energibalansen för ett mark-atmosfärssystem. R_n är nettostrålningen in i systemet, H är det sensibla värmeflödet, LE är det latenta värmeflödet och G är värmeflödet i marken. H, LE och G är positiva i riktningen ut från systemet.

Om *H*, *L·E* eller *G* är negativa betyder det att energi kommer in i kontrollvolymen, antingen genom att sensibel värme flödar in, vattenånga kondenserar och frigör energi, eller markytan värms upp av värmeflöde från marken.

R_n i ekvation (1) är summan av de vertikala komponenterna av direkt och diffus strålning från solen (S) och långvågiga strålning från atmosfären ($L_{\downarrow}$), subtraherat summan av den reflekterade kortvågiga strålningen (αS) och emitterad långvågig strålning ($L_{\uparrow}$). Detta kan sammanställas i följande ekvation

$$Rn = (1 - \alpha) \cdot S + L_{\downarrow} - L_{\uparrow} \quad (2)$$

Hur mycket av den kortvågiga strålningen som reflekteras beror alltså av storleken av α (albedo), som är en egenskap hos markytan vars storlek förändras under året. $L_{\downarrow}$ är den inkommande och $L_{\uparrow}$ den utgående långvågstrålningen. En översikt av nettostrålningens komponenter visas i Figur 2. Nettostrålningen blir då summan av flödena och kan vara både positivt (strålning till marken) och negativt (strålning från marken) (Rodhe 2000).



Figur 2. Nettostrålningens komponenter. Inkommande kort- och långvågig strålning samt utgående kort- och långvågig strålning.

Summan av den inkommande och utgående kortvågiga strålningen är alltid positiv, eftersom utflödet är direkt beroende av storleken på inkommande kortvågig strålning genom markytans albedo (som bara kan anta värde mellan 0 och 1, beroende på markytans egenskaper). Summan av den långvågiga strålningen kan, å andrasidan, vara både positiv och negativ. Det beror på att den utgående långvågiga strålningen inte bara består av reflekterad inkommande strålning ($= [1 - \varepsilon]L_{\downarrow}$) utan också av värmestrålning (Nationalencyklopedin, u.å.c), vars storlek kan beskrivas med Stefan-Boltzmanns lag

$$E = \varepsilon \sigma T^4 \quad (3)$$

där E är den emitterade energin (en del av $L_{\uparrow}$), ε är emissiviteten, σ är Stefan-Boltzmanns konstant och T är markytans temperatur i Kelvin (Nationalencyklopedin, u.å.b).

2.1.1 Evaporation

När termen *evaporation* används i hydrologin, är det för att beskriva flödet av vattenånga från land och vatten till atmosfären. Evapotranspiration är en annan vanligt använd term för att beskriva flödet av vattenånga från land, här inkluderas även transpirationen - som är ångvattenflödet via vegetationen (Rodhe 2000). Eftersom vegetationen var nästan obefintlig på platsen som denna studie utfördes på, kommer termen evaporation användas i denna studie för att beskriva ångvattenflödet. Denna distinktion är viktig då många beräkningsmetoder för potentiell evapo(transpi)ration utgår ifrån att det finns växtlighet som transpirerar alternativt att det finns en referensgröda (*reference crop evapotranspiration*).

Evaporation representerar nettoutflödet av vattenmolekyler från vätskefasen till ångfasen av vatten. Flödet är drivet av skillnaden i ångtryck mellan atmosfären och den yta som evaporation sker ifrån, där flödeshastigheten är proportionell mot skillnaden (Rodhe 2000). Evaporationshastigheten påverkas av flera faktorer. Hög temperatur vid ytan ökar mättnadsångtryck och därmed evaporationshastighet. Evaporation är en energikrävande process som tenderar att kyla ner ytan. För att bibehålla en hög temperatur vid ytan, och därmed ett högt mättnadsångtryck, behöver energi därför tillföras till ytan, genom ett nettoinflöde av energi, genom den process som beskrivits i avsnittet om energibalansen. Torr luft och effektiva transportmekanismer, som drivs av vindinducerad turbulens, gynnar evaporationen genom att avlägsna vattenånga från luften precis ovanför ytan. Tillgång till vatten vid ytan är avgörande för evaporationen, med markvatten, våtmarker, sjöar och öppna vattenytor som potentiella källor. Utöver dessa utgör snö en våt yta för evaporation, men evaporationshastigheten är begränsad på grund av den begränsade ångtrycksskillnaden mellan snöytan och atmosfären (ibid.).

Potentiell evaporation (PE) är den mängd vatten som skulle evaporera från en yta om det fanns en obegränsad mängd tillgängligt vatten (Domenico & Schwartz 1998). Det finns flera olika sätt att uppskatta potentiell evaporation, i denna studie kommer endast olika beräkningsmetoder att jämföras och inga ytterligare mätningar kommer att göras för att uppskatta PE. En vanlig beräkningsmetod för att beräkna potentiell evaporation är Penman-ekvationen, som beräknar PE som en funktion av nettostrålning, vindhastighet, lufttemperatur och luftfuktighet, och formuleras som

$$PE = \frac{\Delta \cdot R_n + \gamma \cdot (e_a - e_s) \cdot f(U)}{L \cdot \rho \cdot (\Delta + \gamma)} \tag{4}$$

där Δ är lutningen på kurvan ångtryck/temperatur vid jämviktstemperaturen, R_n är nettostrålningen, γ är psykrometer konstanten (*psychrometric constant*), $f(U) = 2.63(1 + 0.536 \cdot U)$, och där U är vind hastigheten vid 2 meter, L är ångbildningsvärmen, ρ är vattnets densitet, e_a är det faktiska ångtrycket och e_s är mättnadsångtrycket (Penman 1948; Oudin et al. 2005).

Det finns många andra beräkningsmetoder för at beräkna potentiell evaporation som beror av olika parametrar. Hargreaves och Samani tog 1985 fram Hargreaves ekvationen

$$PE = 0{,}0023 \frac{R_e}{L \cdot \rho} \cdot (T_{max} - T_{min})^{0,5} \cdot (T + 17{,}8) \tag{5}$$

där R_e är utomjordisk (*extraterrestrial*) solstrålning, T_{max} och T_{min} är maximala respektive minimala dagliga temperaturen, T är den genomsnittliga lufttemperaturen. Ekvationen kräver endast att temperaturen (max och min) mäts. R_e kan uppskattas med hjälp av latituden för den studerade platsen.

Priestley-Taylors (1972) ekvation är en annan vanligt använd ekvation för bestämmandet av potentiell evaporation

$$PE = \frac{\alpha_{pt} \cdot \Delta \cdot R_n}{L \cdot \rho \cdot (\Delta + \gamma)} \tag{6}$$

där α_{pt} är en kalibreringskoefficient som har standardvärdet 1,26. Priestley-Taylor är en förenkling av Penman-Monteiths ekvationen som endast kräver att nettostrålningen mäts.

Det sista ekvationen för beräkning av potentiell evaporation som beskrivas närmare i denna studie är Oudins beräkningsmetod

$$\begin{cases} PE = \frac{R_e}{L \cdot \rho} \frac{T + 5}{100}, & om\ T > -5\ °C \\ PE = 0, & om\ T < -5\ °C \end{cases} \tag{7}$$

som togs fram för att på ett mer simpelt sätt beräkna den potentiella evaporationen jämfört med andra vanligt använda metoder men fortfarande få bra resultat vid modellering av regn-avrinningsmodeller (Oudin et al. 2005).

Vid hydrologisk modellering beräknas den faktiska evaporationen ofta som en bråkdel av den potentiella evaporationen, där bråkdelen beror på markens fuktstatus. När marken är tillräckligt våt så är bråkdelen 1,0, det vill säga den faktiska evaporationen är lika med potentiella evaporationen. När jorden torkar ut minskar fraktionen och om uttorkningen fortsätter når den till slut noll (Rodhe 2000).

2.1.2 Värmeflöde genom marken

Markens temperatur beror av det energiutbyte mellan marken och atmosfären (se avsnitt 2.1) och värmerörelser i marken. Markens temperatur (och värmeflöde) påverkar, förutom energibalansen, kemiska reaktioner, biologiska processer och vattnets rörelse i marken (Messing 2013).

Värmetransport i marken sker genom värmeledning eller värmekonvektion. Värmeledning sker genom överföring av värmerörelseenergi mellan partiklar och värmekonvektion är transport av värme genom en fluid i rörelse (Messing 2013). Transport av värme med vatten genom konvektion kan öka värmerörelserna avsevärt, eftersom mycket latent värme kan bindas eller frigöras vid kondensation/evaporation och frysning/smältning av vatten. Värmeledningen i marken bestäms av markegenskaperna värmekapacitet och värmeledningsförmåga. Värmekapaciteten kan beskrivas som en summa av flera markkomponenter (till exempel vatten, luft, lermineral, kvarts) enligt följande ekvation

$$C_h = \sum_i C_{h,i} \Phi_i = \sum_i \rho_i c_i \Phi_i \tag{8}$$

där C_h är den totala volumetriska värmekapaciteten, $C_{h,i}$ är den volumetriska värmekapaciteten för komponent i, Φ_i är volymandelen, c_i är den specifika värmekapaciteten, ρ_i är densiteten för komponent i. Olika markkomponenter bidrar olika mycket till värmekapaciteten (Messing 2013).

Fouriers lag beskriver värmeledning från områden där marktemperaturen är hög till områden är låg, och är definierad enligt ekvationen

$$f_h = \kappa \frac{\partial T}{\partial s} \tag{9}$$

där f_h är värmeflödet, κ är värmeledningsförmågan och $\partial T/\partial s$ är temperatur gradienten som driver värmeflödet. Värmeledningsförmågan, likt värmekapaciteten, är olika för olika beståndsdelar av marken (Messing 2013). Tillsammans beskriver ekvation (8) och (9) hur värme flödar genom marken.

2.2 Hydrologiska processer som styr vattenbalansen på ytan av en gråbergsdeponi

2.2.1 Vattenbalans

Vattenbalansen för en viss plats kan med ord beskrivas som

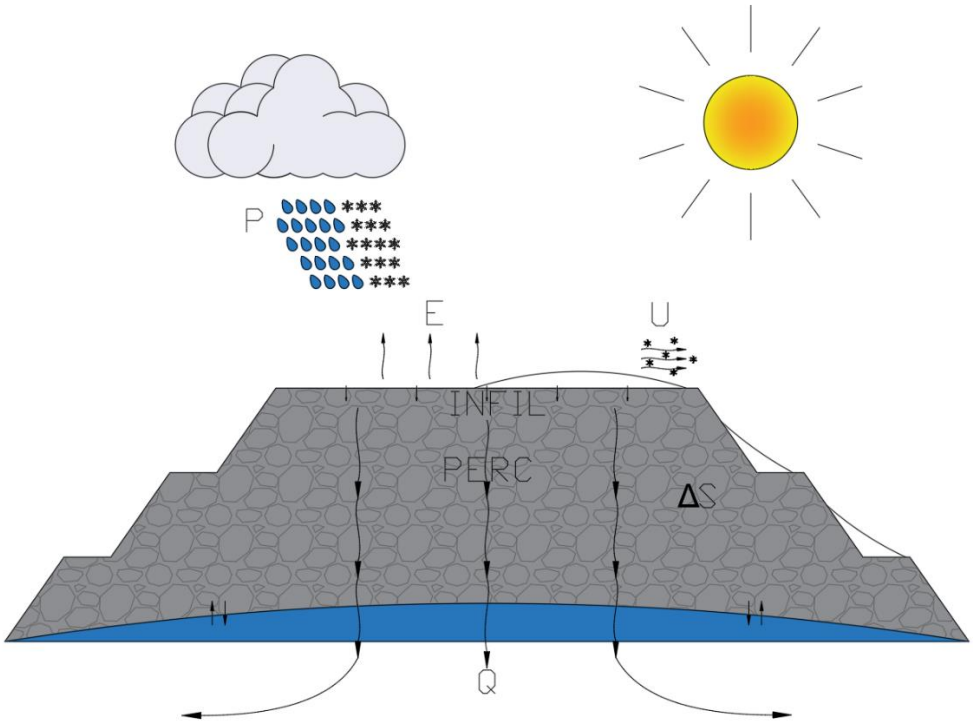
$$Inflöde - utflöde = förändring\ av\ vattenlager \tag{10}$$

Generellt utgör nederbörd inflödet av vatten, och utflödet består av evaporation, ytavrinning och markvattenflöden. Skillnaden mellan in- och utflöde blir då förändringen av mängd vatten som lagras på platsen. En mer detaljerad ekvation, allmänt känd som vattenbalansekvationen, kan skrivas som

$$P = R + Q + E + \Delta S \tag{11}$$

där P är nederbörd (*precipitation*), R är ytavrinning (*surface runoff*), Q är markvattenflödet och ΔS representerar förändringen av vattenlagringen (Domenico & Schwartz 1998). Vanligtvis kan lagringstermen ignoreras när vattenbalansen undersöks över lång tid på en viss plats, till exempel ett avrinningsområde, men är av stor vikt när vattenbalansen undersöks över korta tidsperioder (Rodhe 2000). Men för en gråbergsdeponi som till en början är omättad innan den utsätts för det lokala klimatet kommer vatten att lagras i deponin över tiden och därför vara av betydelse när hela gråbergsdeponin tas i hänsyn (Domenico & Schwartz 1998; Amos et al. 2015).

Vattenbalansen för en gråbergsdeponi visas i Figur 3. Vatten kommer till gråbergsdeponin genom nederbörd, nederbörd som kommer som snö ackumuleras på ytan. Snön smälter med tiden eller förflyttas bort med vinden. Regn- och smältvatten kan återgå till atmosfären genom evaporation eller infiltrera marken och vidare perkolera ner genom gråbergsdeponin där lagring vatten kan ske. Efter vattnets väg genom gråberget har det bildat lakvattnet när det lämnar deponin. Ytavrinning har inte tagits med som en del av vattenbalansen, då markens grova textur leder till stor infiltrationskapacitet och därmed ingen bildning av ytavrinning av större betydelse.



Figur 3. Vattenbalansen för en gråbergsdeponi, P representerar nederbörden i form av regn eller snö, E representerar flödet av vattenånga genom evaporation, Q representerar markvattenflödet, ΔS representerar förändringen av vattenlagring över tid i deponin. Även andra processer som har påverkan på flödet av vattnet har

märkts ut, U representerar vindens påverkan på framför allt snöackumuleringen, INFIL representerar infiltrationen, PERC representerar att vattnet perkolerar genom berget.

2.2.2 Snö och snösmältning

Under vintern kan stora mängder nederbörd samlas i snötäcket, evaporationen är låg under samma tid så kan snötäcket ha stor betydelse för grundvattenbildning och avrinningens storlek (Rodhe 2000). Detta innebär att snötäckets storlek, tiden för snösmältning och i vilken hastighet snön smälter kan vara viktiga variabler att ta hänsyn till i hydrologisk modellering likt den som gjorts i denna studie.

Snötäckets storlek beskrivs enklast med snöns vattenekvivalent (SWE, *Snow Water Equivalent*), vilket definieras som det vattendjup som skulle erhållas om all snö skulle smälta. SWE anges i mm vattendjup och kan beräknas enligt följande ekvation

$$SWE = \rho \cdot h \tag{12}$$

där ρ är snöns densitet [kg/m³] och h är snöns djup [m]. Snöns densitet varierar över säsongen där nyfallen snö kan väga 100 kg/m³ och uppemot 400 kg/m³ för smältande snö. Densiteten ökar gradvis under vintern på grund av packning från överliggande snö och vissa hydrologiska processer (Rodhe 2000). Om ekvation (12) skrivs om något kan SWE beräknas från snöns vikt, om den vägs jämnfördelat på en våg med känd area.

Snötäckets storlek kan variera stort inom ett geografiskt område på grund av flera faktorer. Höjden för platsen påverkar mängden nederbörd som faller som snö, där generellt högre höjder har lägre temperaturer och större nederbörd, vilket då kan leda till ökande mängd snö med ökande höjd (Rodhe 2000). Vinden kan omfördela snön genom att erodera den från områden av höga vindhastigheter till platser med relativt lägre vindhastigheter, skyddat av topografiska särdrag. Detta gäller särskilt områden i vegetationsfattiga platser med mycket snöfall och relativt starka vindar (Hiemstra et al. 2002).

För att smälta snö krävs mycket energi. En modifierad form av energibalansen (se avsnitt 2.1) kan användas för att beräkna energin tillgänglig för att smälta snön och därmed snösmältningshastigheten enligt följande ekvation

$$Q_{smält} = R_n - H - L \cdot E - G + Q_p = L_s \cdot M \tag{13}$$

där $Q_{smält}$ är energin tillgänglig för smältning, R_n är nettostrålningen, H är det sensibla värmeflödet, $L{\cdot}E$ är den latenta värmen som frigörs ovanpå snön, G är värmeflödet mellan snön och marken, Q_p är värmen som kommer genom varmt regnvatten, L_s är smältvärmen och M är snösmältningshastigheten (Rodhe 2000). Som tidigare beskrivit kan flödena av energi ske både från och till snön (markytan). När luften är tillräckligt fuktig kommer vattenånga kondensera vilket leder till att latent värme blir tillgängligt för snösmältning. När däremot luftens ångtryck är lägre än snöns mättnadsångtryck kommer avdunstning att ske, vilket är energikrävande och minskar därmed drastiskt energin som är tillgänglig för snösmältning. Energin som kommer från regn är relativt liten, även vid större regn. Förhållanden som är gynnsamma för snösmältning är stor nettostrålning, varm luft (sensibelt värmeflöde till snötäcket), fuktig luft som ger upphov till kondensation och stark vind som ger upphov till transport av vattenånga och värme till snötäcket (ibid.).

I denna studie kommer graddagsmetoden att användas vid modellering av snötäcket i den konceptuella modellen (se avsnitt 4.3), som är förenklad till att endast ha ett temperaturberoende enligt följande ekvation

$$M = C_{FMAX}(T - T_T) \tag{14}$$

där M [mm d^{-1}] är smältningshastigheten, C_{FMAX} [mm/dag/°C] är graddagsfaktorn, T [°C] är lufttemperaturen, T_T [°C] är tröskeltemperaturen. Snötäcket är ett poröst medium som kan lagra vatten till en viss kapacitet. Snöns vattenlagrande kapacitet kan beskrivas som en fraktion av snötäckets vattenekvivalent, *CWH* [-], där vattnet hålls kvar av kapillärkrafter. Således kommer inte vatten att lämna snötäcket förrän denna kapacitet överskridits. Om temperaturen sjunker under tröskeltemperaturen så kan vatten som har smält (och som finns kvar i snön) återigen frysa (Rodhe 2000; Seibert 2005), vilket sker enligt ekvationen

$$RFZ = C_{FR} C_{FMAX} (T_T - T) \tag{15}$$

där *RFZ* [mm d^{-1}] är mängden flytande vatten som återfryser, C_{FR} [-] är en faktor som beskriver hur fort vattnet återfryser. Vattnet återfryser generellt saktare än vad det smälter då vattnet som kan frysa återfinns i snötäckets porer och är där inte lika lätt påverkad av förändringar av lufttemperaturen (Seibert 2005).

Nederbördsmätare underskattar ofta mängden snö som faller därför behövs en korrigeringsfaktor, S_{FCF} [-], för att korrigera detta (Seibert 2005).

2.2.3 Vattenflöde genom marken

Vattenflöde genom marken antas vara laminärt och därmed följa Darcys lag. Darcys lag beskriver vattens flödeshastighet genom ett poröst medium, formulerad enligt ekvationen

$$q = -K \frac{\partial H}{\partial s} = -K \left(\frac{\partial h}{\partial s} + \frac{\partial z}{\partial s} \right) \tag{16}$$

där q är flödestätheten [m/s], K är den hydrauliska konduktiviteten [m/s] och $\partial H/\partial s$ är den hydrauliska gradienten [m/m] som ”driver” flödet genom marken. Flödestätheten q beskriver den mängd vatten som flödar genom ett plan vinkelrätt mot flödesriktningen s. Den hydrauliska gradienten kan delas upp i två komponenter: en komponent som tillkommer genom tryckpotentialskillnader ($\partial h/\partial s$) och en komponent som tillkommer genom gravitationspotentialskillnader ($\partial z/\partial s$). Om flödesriktningen (s) är i riktning av z (vertikalt nedåt) så blir $\partial z/\partial s=1$ (Messing 2013).

2.2.3.1 Omättat flöde

Där porutrymmen inte är helt fyllda med vatten utan innehåller även luft, kallas markens omättade zon. Den omättade zonen befinner sig mellan grundvattennivån (markens mättade zon) och markytan. Omättat flöde definieras som det sätt med vilket vatten rör sig genom den omättade zonen där porerna inte är helt fyllda med vatten (Szymkiewicz 2013). Gråbergsdeponier har generellt omättade vattenförhållanden, även om det lokalt kan finnas mättade förhållanden (Blackmore et al. 2014).

Vid omättat flöde genomgår vattnet två huvudsakliga processer: infiltration och perkolering. Infiltrationen sker när vatten tränger in i marken från ytan, antingen genom nederbörd, bevattning eller andra källor. Detta vatten rör sig sedan nedåt genom markens porösa struktur genom perkolering (Rodhe 2000), se Figur 3.

Richards tog 1931 fram en ekvation som beskriver förändringen av vattenhalten i marken över tid baserat på Darcys lag och lagen om massans bevarande (kontinuitetsvillkoret). Ekvationen kan användas för att beräkna vattnets flöde och lagring av vatten i en omättad mark, och formuleras

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial s} \left[K(h) \left(\frac{\partial h}{\partial s} + \frac{\partial z}{\partial s} \right) \right] - a_w \tag{17}$$

där $\partial\theta/\partial t$ är förändringen av vattenhalten över tid, $K(h)$ är den hydrauliska konduktivitetsfunktionen som beror av h på en viss plats, a_w är en term relaterad till vattenvolym

tillfört eller bortfört (till exempel genom rotupptag) som i vissa fall kan ignoreras då den är förhållandevis liten jämfört med de andra termerna (Messing 2013).

2.2.3.2 Matrisflöde

När vatten och lösta ämnen flödar genom porsystemet i en homogen mark brukar detta flöde kallas matrisflöde. Detta flöde kan också kallas jämviktsflöde, vilket är ett uniformt flöde som kan beskrivas av Darcys lag (se ekvation [16]). Under omättade förhållanden är *K* i ekvation (16) en funktion av tryckpotentialen (*K[h]*). Flödet drivs av kapillärkrafter och skillnader i vattenpotential mellan olika delar av markprofilen. Porerna inom markmatrisen kan vara av varierande storlek och form, vilket påverkar flödeshastigheten och riktningen för vattenrörelsen (Messing 2013; Nimmo 2021).

En viktig faktor som påverkar matrisflödet är markens textur. Jordar med finare partiklar, såsom leror, har mindre porstorlek och kan därför ha högre kapillärkrafter, vilket leder till ökad vattenretention och långsammare flödeshastigheter jämfört med jordar med grövre partiklar. Utöver markens textur påverkas matrisflödet också av faktorer som markfuktighet, temperatur, och närvaron av organiskt material. Till exempel kan högre markfuktighet leda till minskade kapillärkrafter och därmed en ökad benägenhet för vattenflöde via makroporer (se nedan) (Messing 2013; Nimmo 2021).

2.2.3.3 Preferentiellt flöde

Preferentiellt flöde (*preferential flow*) avser icke-jämviktsflöde av vatten och lösta ämnen genom markprofilen, och kännetecknas av flöde genom preferentiella flödesvägar med minsta motstånd som utgör en liten del av mediet och förbigår därav mycket av mediet (markprofilen). Andra egenskaper som kännetecknar preferentiellt flöde är att flödet sker generellt snabbare än andra markflöden, att flödet inte är kontinuerligt under naturliga förhållanden och drivet av gravitationen med liten inverkan av kapillära krafter (Messing 2013; Nimmo 2021).

Det finns generellt tre accepterade kategorier av preferentiellt flöde. En typ av preferentiellt flöde är ”trattflöde” (*funnel flow*) som uppstår på grund av heterogeniteter i markens hydrauliska egenskaper. Preferentiella flödesvägar uppstår där mediets ledande förmåga är störst. Ökande vattenhalt i dessa föredragna flödesvägar leder till en ökad konduktivitet vilket stärker ”tratteffekten” (Nimmo 2021).

En annan typ av preferentiellt flöde är makroporflöde. Det är vattenflöde genom porer med stor diameter. För att en por ska definieras som en makropor ska den ha en diameter större än 0,3-0,5 mm (definitionen varierar) (Messing 2013). Makroporer kan till exempel bildas genom grävande djur (bioporer) och tomrum mellan aggregat i marken. Då makroporerna är de största porerna kommer dessa porer att vattenfyllas först efter jordmatrisen fyllts med vatten, alternativt vid nederbörd så kan makroporerna till en början fyllas även om jordmatrisen är torr. Om jordmatrisen är torr och vatten tränger in i makroporerna, skapas en potentialgradient som driver flöde från makroporerna in i jordmatrisen. Hastigheten med vilken makroporerna töms på vatten beror på potentialgradienten och matrisens hydrauliska konduktivitet. Ju torrare jordmatrisen är, desto större är potentialgradienten, vilket resulterar i ökat flöde in i jordmatrisen. Om jordmatrisen i stället är våt kommer det leda till ett större vertikalt flöde i makroporerna. Flödeshastigheten i makroporerna måste vara stor i förhållanden till den horisontella utjämning som potentialgradienten driver för att ett makroporflöde ska upprätthållas. Större pordiameter leder till mindre friktion när vattnet flödar genom poren, där det är proportionellt mot porstorleken upphöjt i fyra. Om det förekommer heterogeniteter i marken där det finns flödesvägar med dessa två egenskaper kommer vattnet föredra att flöda genom dessa vägar (Messing 2013).

Den sista typen av preferentiellt flöde är ”fingerflödet”. Detta flöde uppkommer när delar av marken blivit vattenavstötande, på de ställen som inte är vattenavstötande kan vatten infiltrera och det bildas ”fingrar” där vattnet rör sig ner genom marken (Messing 2013).

Att preferentiellt flöde förekommer i omättad poröst medium, som gråbergsdeponin som denna studie tittar på, är mer regel än undantag. Speciellt gäller detta gråberg med hög andel av grus där den rumsliga fördelningen av gruspartiklar skapar makroporer och heterogeniteter i porstorleksfördelningen (Blackmore et al. 2014; Keller et al. 2015). Många modeller använder Richards ekvation (se ekvation [17]), för att beräkna omättat flöde men om förekomsten av preferentiellt flöde är stor kan inte Richards ekvation användas direkt. I sådana fall används exempelvis en modell som kan ta hänsyn till flera olika hydrauliska konduktivitetsfunktioner (K(h)), som beskriver olika delar av porstorleksfördelningen (Keller et al. 2015).

2.2.3.4 Vattenretentionskurvan

Vattnets flöde i marken är nära kopplat till vattenretentionskurvan, som relaterar vattenhalten (θ) till tryckpotentialen (h). Det finns flera olika sätt att beskriva markens vattenretentionskurva, i denna studie kommer vattenretentionskurvan enligt Brooks och Corey (1964) beskrivas.

Effektiv mättnad (S_e) kan beskrivas med ekvation

$$S_e = \left(\frac{h}{h_a}\right)^{-\lambda} \tag{18}$$

där h_a är den tryckpotential som krävs för att luft ska kunna träda in i marken och ersätta vatten i porutrymmet (luftinträdestryck, *air entry pressure*) och λ är en anpassningsparameter relaterade till porstorleksfördelningen i marken. S_e beskriver mättnadsgrad där endast det vatten som faktiskt är tillgängligt för vattenflöde tas i hänsyn som andel av totalporositeten (θ_s) och kan även beskrivas med följande ekvation

$$S_e = \left(\frac{\theta - \theta_r}{\theta_s - \theta_r}\right) \tag{19}$$

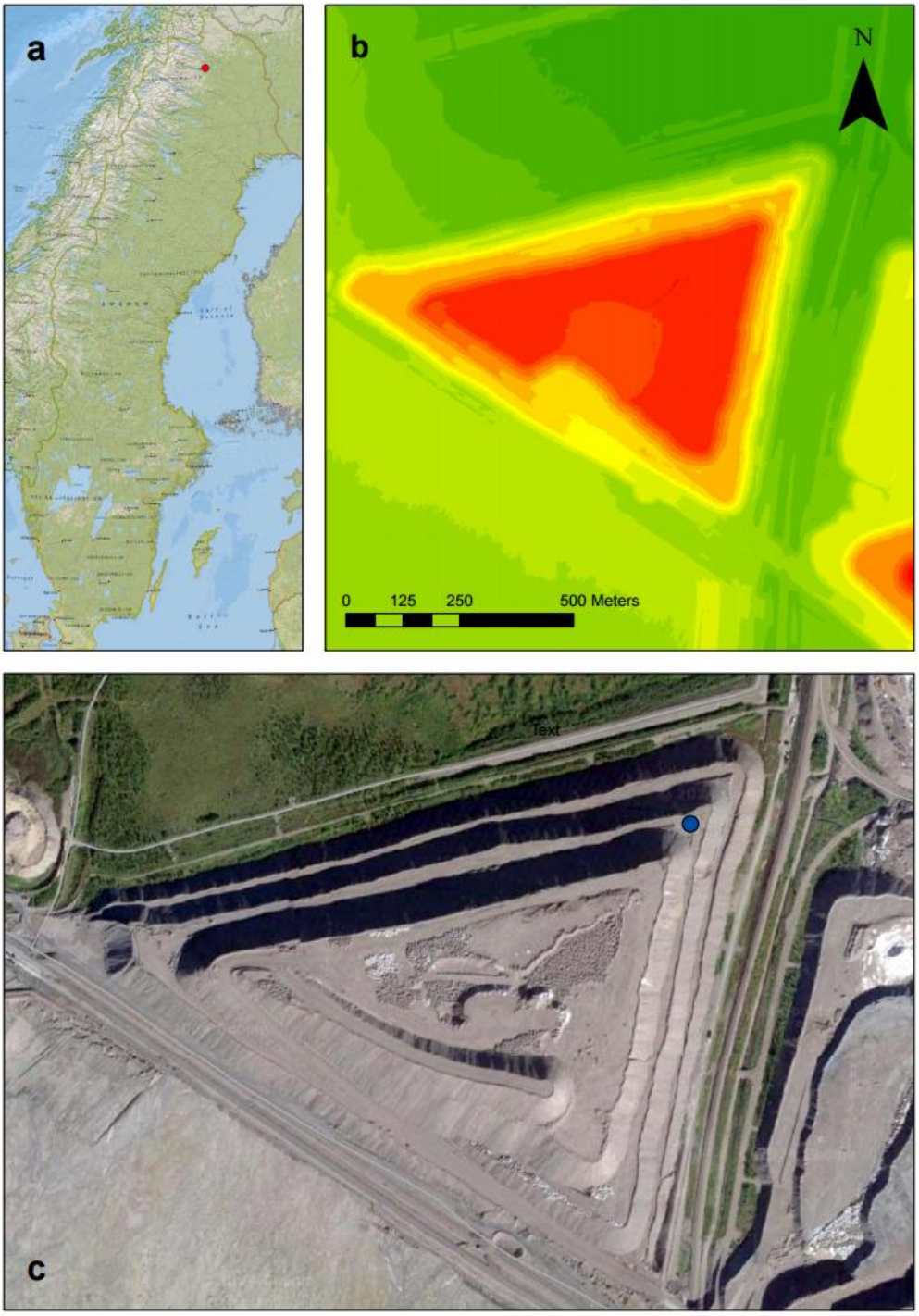
där θ_r är den residuala vattenhalten. Om ekvation (18) och ekvation (19) kombineras och löses för θ, erhålls följande ekvation

$$\theta(h) = \theta_r + (\theta_s - \theta_r) \cdot \left(\frac{h}{h_a}\right)^{-\lambda} \quad för\ h > h_a \tag{20}$$

Om θ som funktion av h plottas, erhålls vattenretentionskurvan för den aktuella marken. Enligt ekvation (20) sker det ingen förändring av vattenhalten när marken är mättad förrän h är större än h_a (Brooks & Corey 1964; Jansson & Karlberg 2011).

3 Platsbeskrivning och tidigare fältresultat

Vid järnmalmsgruvan Kirunagruvan, som bedrivs av Luossavaara-Kiirunavaara Aktiebolag (LKAB), finns flera gråbergsdeponier. Denna studie genomfördes vid en av dessa deponier som kallas ”Triangeldeponin”. Triangeldeponin (67°51'00"N 20°08'00"Ö) ligger i Kiruna vilket är en stad i norra delen av Sverige. Kiruna ligger ovanför polcirkeln och har subarktiskt klimat med korta svala somrar med långa kalla vintrar. I Figur 4 visas Triangeldeponins utformning och geografiska plats under tiden för studien. Triangeldeponin hade en ungefärlig area av 0,5 km² och deponin var en bit under 100 meter hög, där toppen ligger på cirka 595 meter över havsytan.



Figur 4. a) geografisk placering av Triangeldeponin i Sverige, markerad med röd punkt, bakgrundskarta från ESRI. b) höjdmodell över Triangeldeponin framtagen av LKAB (2020). c) satellitfoto (CNES/Airbus, 2024) av Triangeldeponin, blå markering utgör platsen för den meteorologiska stationen, foto taget från Google Maps.

Medeltemperaturen i Kiruna är 0 °C för standardnormalperioden (1991–2020), vilket är en höjning med 1,5 °C jämfört med den tidigare normalperioden (1961–1990). Antal dygn med medeltemperaturer under eller lika med 0 °C under normalperioden har varit 194 dygn/år (SMHI, 2024a). Årsnederbörden för normalperioden har i snitt varit 600 mm/år (SMHI, 2024b).

Från Triangeldeponin lämnar ett neutralt lakvatten, där huvudproblematiken är det kväve som lakas ut. Därför uppfördes en kvävereningsanläggning nära deponin som renar delar av det vatten som rinner genom gråberget. Kvävereningsanläggningen har varit igång sedan 2018 (Hellman et al. 2024).

3.1 Meteorologisk station

På Triangeldeponin har en meteorologisk mätstation uppförts, se placering i Figur 4 c. Mätstationen har en mängd olika sensorer, en översikt över dessa visas i Tabell 1. Sensorerna mäter meteorologiska variabler såsom lufttemperatur, marktemperatur, vindhastighet, nederbörd, nettostrålning och många fler variabler. Den meteorologiska stationen är uppdelad i två delar: en ovan mark (se Figur 19 i bilaga A) och en under mark (se Figur 20 i bilaga A) (Grabs 2024).

Tabell 1. Översikt över den meteorologiska mätstationens sensorer, inkluderat), antalet sensorer, plats (OM: ovan markytan, UM: under markytan), sensorernas viktigaste variabler och i vilket ”sammanhang” parametern används (Typ, EB: energibalans, VB: vattenbalans).

Sensor	Antal	Plats	Huvudmätvariabler	Typ
Pyrometer	1	OM	Nettostrålning	EB
Luftfuktighetsmätare	3	OM	Lufttemperatur och relativ luftfuktighet	EB
Anemometer	1	OM	Vindhastighet och vindriktning	EB
Medeltemperaturgivare	1	UM	Medeltemperatur i marken	EB
Temperaturgivare	2	UM	Lokal temperatur i marken	EB
Värmeflödesmätare	2	UM	Värmeflöde i marken	EB
Nederbördsmätare	1	OM	Nederbördsintensitet och ackumulerad volym	VB
Snövåg	1	OM	Snövatten ekvivalent	VB
Ultraljudssensor	1	OM	Snödjup	VB
TDR	6	UM	Marktemperatur, vattenhalt	VB, EB

3.2 Tidigare studier vid LKAB:s järnmalmsgruva i Kiruna

Utöver Atmosudirdjo (2019) (se avsnitt 1.1) har flera andra tidigare studier och arbeten gjorts vid LKAB:s järngruva i Kiruna, både direkt anknutna till denna studie, och andra studier relaterat till, bland annat, kvävereningsanläggning. Nedan redogörs för de studier och arbeten som har större koppling till denna studie.

Grabs (2024) har innan projektets början gjort en preliminär dataanalys på de data som mäts av sensorerna från 2021-10-01 till 2023-06-01 (senare i rapporten kallad ”mätperioden”), det är dessa data som användes som indata i denna studie. Dataanalysen gjordes i MATLAB där data sammanställdes i olika tidsupplösningar (ursprungsdata loggades var femtonde minut), plottades, nya variabler beräknades, och luckor i några variabler fylldes igen.

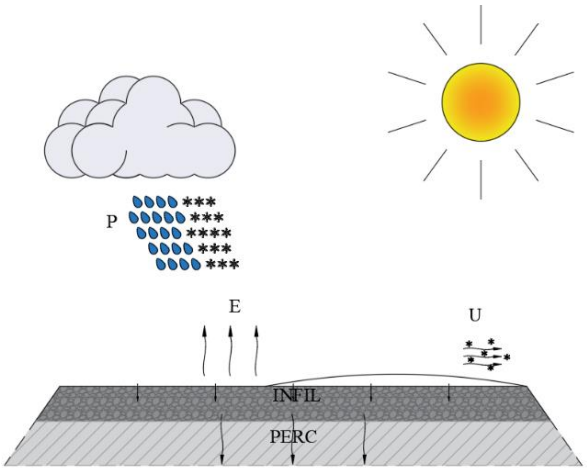
Jørgensens (2022) studie ämnade att kalibrera de TDR-sensorer som användes i denna studie. Utöver detta arbete tog Jørgensen fram Triangeldeponins porositet, vilket uppskattades till 19,13 % för siktade prov och 21,66 % för osiktade prov och tog fram kornstorleksindelning för siktade och osiktade jordprov, vilket visas i Tabell 2.

Tabell 2. Kornstorleksindelning för Triangeldeponin enligt Udden-Wentworth klassifikationsschema. Mätningar av Jørgensen (2022).

Kornstorlek [mm]	Kornklassificering	Siktat jordprov [%]	Osiktat jordprov [%]
>256	Block		
128–256	Sten	24,00	50,53
4–64	Grus		
2–4	Granulat	37,81	24,61
0,063–2	Sand	37,81	24,61
0,004–0,063	Silt	0,38	0,25
<0,004	Ler		

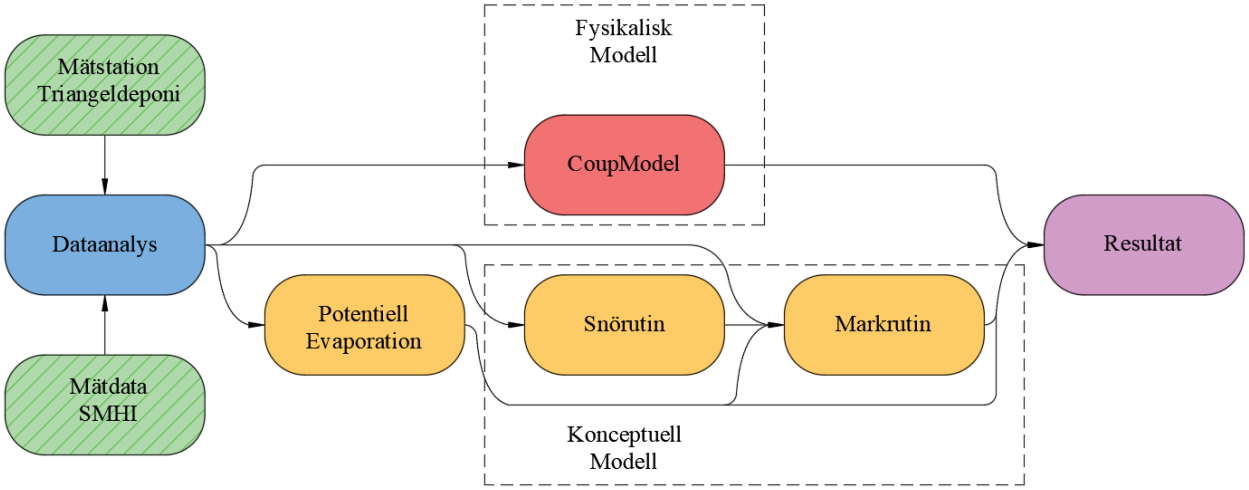
4 Metod

Detta avsnitt avser redogöra för hur modelleringen av vattenbalansen vid ytan av Triangeldeponin, se Figur 5, utfördes. Vid beräkning av vattenbalansen vid ytan är förändringen i lagring (ΔS, se Figur 3) relativt små jämfört med det årliga hydrologiska flödet. Lagringstermen beaktas dock indirekt genom modellering av markfuktigheten, där förändringar i lagring återspeglar antingen ett ökat inflöde eller utflöde av vatten i markprofilen. Det innebär att det är fyra flöden som har betydande påverkan på vattenbalansen: nederbörd, evaporation, markvattenflöde (infiltration som övergår i perkolation) och omfördelning av snö med vinden.



Figur 5. Komponenter av vattenbalansen vid ytan av en gråbergsdeponi.

En dataanalys genomfördes för att fylla igen stora och små luckor i mätdata, detta eftersom modellering av vattenbalansen kräver kontinuerliga data. Vattenbalansen beräknades med två olika modeller: en modell skriven i programmeringsspråket Python och programmet CoupModel. Bakgrund till dessa modeller och hur de fungerar beskrivs under varsin underrubrik. Båda modellerna simulerar ett endimensionellt (1D) vertikalt vattenflöde. Den konceptuella modellen behövde även potentiell evaporation som indata, vilket beräknades med hjälp av Python-paketet PyET. Ett flödesschema över metoden visas i Figur 6. Pilarna i figuren visar flödet av data från en modul till en annan samt flödet av resultat som presenteras i avsnitt 5.



Figur 6. Flödesschema över metod. Pilar visar flödet av data och resultat.

Den meteorologiska stationen mätte en mängd olika variabler som inte var relevant för denna studie. Variabler som valdes att inkludera i dataanalysen nämns under ”Huvudmätvariabler” (bortsett vindriktning) i Tabell 1 samt lufttemperatur mätt av ultraljudssensorn, då variabeln skulle komma till användning senare i modelleringen. Lufttemperatur och relativ luftfuktighet mättes på flera höjder (se Figur 19 i bilaga A), där det valdes att endast använda data från luftfuktighetssensorn som satt 1,65 m över markytan (inom standardiserat spann för temperaturmätare, se Barnett et al. 1998). För de sensorer av samma typ som mäter på samma djup (TDR, temperaturgivare och värmeflödesmätare) hade ett medelvärde tagits mellan sensorerna innan studiens början, vilket representerar den aktuella variabeln.

4.1 Dataanalys

4.1.1 Fylla igen dataluckor

För att beräkna potentiell evaporation och modellera vattenbalansen krävdes kontinuerliga indata. Som beskrivits i avsnitt 3.2, hade en preliminär dataanalys utförts innan studiens början. Alla observerade variabler hade dataluckor i data, luckornas storlek skiljdes åt mellan olika variabler beroende på vilken sensor som mätt variabeln. Sju längre perioder på cirka 20 dagar (vardera) saknade mätdata för flera sensorer. Sensorer av speciell betydelse som påverkades av detta var pyrometern, luftfuktighetsmätarna och värmeflödesmätarna. Det fanns även ett antal kortare perioder (upp till två dagar åt gången som mest) där det inte fanns någon mätdata alls från stationen. Totalt påverkades 149 heldagar under en period av 609 dagar. För att fylla i dataluckorna var det nödvändigt att använda flera olika strategier.

Marktemperatur mättes av tre olika sensorer: medeltemperaturgivare, temperaturgivare och TDR. TDR-sensorn, som också mätte markfuktighet, hade betydligt mindre och färre dataluckor i data än medeltemperaturgivaren och temperaturgivarna. En jämförelse gjordes av observerade värden för de olika sensorerna i översta lagret, vilka mätte på liknande djup (se Figur 20 i bilaga A). Det visade sig att observerade marktemperaturer med de olika sensorerna hade god överenstämmelse. Eftersom ingen ytterligare information om marktemperaturen kunde erhållas från medeltemperaturgivaren eller temperaturgivarna utöver det TDR-sensorn gav, valdes att inte försöka fylla igen dataluckorna för dessa variabler. Då det inte fanns någon externa data som kunde appliceras, eller någon möjlighet att fylla i de mindre dataluckorna (<2 dagar) med hjälp av andra variabler från stationen, användes linjär interpolation mellan de två närmaste datapunkter som observerats (en före och en efter dataluckorna). Marktemperaturen och markfuktigheten uppvisade generellt sett små fluktuationer under korta perioder, vilket gjorde linjär interpolation till en lämplig metod för att fylla i dessa mindre dataluckor.

För värmeflödesmätaren var inte linjär interpolation lämpligt att använda dels för att det var så stora luckor i data, dels för att värmeflödet fluktuerade från dag till dag och även inom dygnet. Inte heller någon data från någon annan mätstation kunde användas för denna variabel, observerad marktemperatur från översta lagret (9 cm djup, se Figur 20 i bilaga A) kunde dock användas. Värmeflödet är ett flöde av energi under en viss tid, därför togs en regression fram mellan förändringen av marktemperaturen (utan dataluckor) mellan varje tidssteg och observerade värmeflödesvärden. Regressionen hade god linjäritet (se Figur 29 i bilaga C), luckorna fylldes därför igen med värden framtagna med denna ekvation.

Mätdata från ultraljudssensor hade bara små dataluckor. Sensorn hade en inbyggd lufttemperaturmätare som visade god linjäritet ($R^2 = 0{,}9994$) med observerad lufttemperatur mätt med luftfuktighetsmätaren. Det ansågs att lufttemperatur mätt med luftfuktighetsmätaren var tillförlitligare och mer relevanta (mätt på ”bättre” höjd, se Figur 19 i bilaga A) där data fanns. Därför fylldes de stora dataluckorna i lufttemperatur mätt med luftfuktighetsmätaren igen där data från ultraljudsensorn fanns. Efter detta kvarstod det endast små dataluckor i lufttemperatursdata.

För att fylla igen luckor för variabler som mäts i luften användes data från SMHI:s meteorologiska mätstationer Kiruna Flygplats och Kiruna Sol. Från Kiruna Flygplats hämtades variablerna lufttemperatur, relativ luftfuktighet, vindriktning och vindhastighet och från Kiruna Sol hämtades variablerna globalstrålning (kortvågsstrålning) och långvågsstrålning. För variablerna lufttemperatur, relativ luftfuktighet och vindhastighet togs regressioner fram mellan data observerad på Triangeldeponin och data från SMHI. Regressionerna visade på god linjäritet för lufttemperatur och relativ luftfuktighet, vilket visas i Figur 26–27 i bilaga C. Med den linjära ekvation som togs fram med regression fylldes dataluckorna igen när den applicerades på de observerade värden från SMHI:s station. För vindhastigheten kunde inte en regression med god linjäritet tas fram (delvis på grund av noggrannhet hos SMHI:s mätningar, se regression i Figur 28 i bilaga C), därför bestämdes det att fylla igen dataluckorna genom att direkt ta de observerade vindhastigheterna på Kiruna flygplats.

Nettostrålningen beror, som ekvation (2) och (3) beskriver, av flera faktorer: markytans temperatur, långvågig strålning, kortvågig strålning, markens albedo och emissivitet. De data som observerades i mätstationen Kiruna Sol var inkommande kortvågig strålning och inkommande långvågig strålning. Regressioner mellan observerad nettostrålning, inkommande kortvågig strålning och inkommande långvågig strålning togs fram i olika kombinationer. Regressioner mellan endast nettostrålning och kortvågig strålning visade god linjäritet (se Figur 24 i bilaga C), men lika bra linjäritet gick inte att hitta mellan nettostrålning och långvågig strålning (se Figur 25 i bilaga C).

Excels dataanalysverktyg *Regression* tillåter användaren att ha flera förklarande variabler ($k_1x_1 + k_2x_2 + \ldots + m = y$). Verktyget användes för att ta fram regressioner mellan nettostrålning och kortvågig strålning och långvågig strålning som förklarande variabler. Detta gjordes även med en tredje förklarande variabel som skulle representera den emitterade energin, enligt ekvation (3). Då markytans temperatur inte mättes användes den överst mätta marktemperaturen (9 cm under markytan) som ett alternativ och lufttemperatur som ett annat alternativ. Observerade data delades in efter om det fanns snö eller ej, då albedo kan skilja sig drastiskt om snö ligger på marken eller inte. För perioderna utan snö visade alla kombinationer på något bättre förklaring än vad den kortvågiga strålningen gjorde, där kombinationen av förklarande variablerna kortvågig strålning, långvågig strålning och lufttemperatur (σT^4, enligt ekvation [3]) visade på den bästa linjäriteten. Samma kombination visade bäst linjäritet även när snö fanns. För gränsfallet när SWE var mellan 0 och 0,5 mm gjordes en separat regression med kombinationen kortvågig strålning, långvågig strålning och marktemperatur (σT^4, enligt ekvation [3]). Ekvationerna för de olika regressionerna användes sedan för att fylla igen dataluckorna, där hänsyn togs till om snö fanns, var inom

”gränsfallet” eller om ingen snö låg på marken. Uppdelningen gjordes med antagande om att det borde bäst representera verkligheten, även om linjäritet för regressionen när det fanns snö inte var lika bra som utan (se Tabell 11–Tabell 13 i bilaga C).

För övriga variabler där dataluckorna var maximalt två dagar användes linjär interpolation mellan värdet direkt innan och värdet direkt efter dataluckan, likt det som gjordes för marktemperaturen och markfuktigheten.

Nederbörd saknade data för de första 25 dagarna. Det valdes att fylla i data med observerad nederbörd från SMHI:s mätstation Rensjön, som var den närmaste stationen som mätte nederbörd per timme.

De ovanbeskrivna strategierna applicerades först på att fylla igen luckorna i observerade data med timbasis, sedan togs data, utan dataluckor, med dygnsbasis fram utifrån data med timbasis (antingen genom medelvärdesbildning eller summering av värdena under ett dygn).

4.1.2 Maximum- och minimumvärden

För lufttemperaturen och relativ luftfuktighet behövdes dagliga maximum- och minimumvärden som indata till att beräkna den potentiella evaporationen (se avsnitt 4.2). Maximum- och minimumvärde för ett dygn togs fram genom att hitta det största respektive minsta timsvärdet som observerats under ett dygn.

4.1.3 Nederbördskorrigering

Innan studiens början hade Grabs (2024) gjort en korrigering av nederbördsdata eftersom nederbördsmätaren ansågs underskatta nederbördsmängden. Under studien gjordes ytterligare en korrigering, där observerad nederbörd för dagarna 2022-04-08 och 2022-04-09 hämtades från SMHI:s mätstation i Kiruna, som mäter nederbörd per dag. En faktor mellan observerad nederbörd på Triangeldeponin och i Kiruna togs fram, vilken timvärdena multiplicerades med. Detta gjordes för att förbättra sambandet mellan nederbörd och observerad snömängd. Baserat på att det blåste mycket dessa dagar, antogs att snön inte fångades upp i lika stor utsträckning som den föll.

4.1.4 Snödjupskorrigering

Observerat snödjup visade under hela mätperioden ett snödjup större än noll. Det framgick tydligt från annan data att snö inte fanns på Triangeldeponin under hela mätperioden. Detta blev särskilt tydligt när snödjupet jämfördes med observerad SWE, där snödjupsmätaren hade registrerat ett snödjup på 2–3 cm när SWE var noll. Snödjupet justerades därför nedåt till att bli noll för alla tillfällen där SWE var noll, förutom dagarna innan SWE var nollskilt.

Under tre dagar (från 2022-10-26 till 2022-10-28) hade snödjupsmätaren registrerat ett snödjup på över 20 cm som kom och försvann snabbt, vilket inte kunde förklaras av SWE eller annan observerade data; dessa tillfällen sattes till noll eftersom SWE var noll.

Det antogs att SWE-mätningarna var mer pålitliga än snödjupssensorn. I CoupModel (den fysikaliska modellen, se avsnitt 4.4) kan endast snödjupet anges. För att få ett rimligt värde på SWE, det vill säga den totala mängden vatten, i modellen halverades snödjupet i april 2022 innan det användes som indata till modellen. Detta har med att göra med vilken densitet som snön antas ha i programmet som används för att konvertera snödjup till SWE (se ekvation [12]).

4.2 Potentiell evaporation

Den konceptuella modellen, se avsnitt 4.3, behövde ha potentiell evaporation (PE) som indata. Den meteorologiska stationen mäter inte direkt potentiell eller faktisk evaporation utan endast variabler vilka kan användas för att beräkna dessa. Det var därför nödvändigt att beräkna PE.

Det finns många olika beräkningsmetoder för att beräkna potentiell evaporation (se några exempel i avsnitt 2.1.1), därför valdes det att använda Python-paket PyET (Vremec et al. 2023). Python är

ett programmeringsspråk med öppen källkod (”open-source”), där paket kan användas för att ge användaren möjlighet att få tillgång till bibliotek med färdigdefinierad funktionalitet. PyET låter användaren enkelt beräkna PE enligt 20 olika beräkningsmetoder. Beräkningsmetoderna skiljer sig åt i hur många variabler som behövs för att beräkna PE där det finns mer simpla metoder som baseras på ett fåtal variabler men också mer komplicerade beräkningsmetoder som kräver fler variabler. Vissa variabler är krav på att ha som indata medan andra variabler kan uppskattas av paketets funktionalitet. Variablerna som användes i beräkningen av den potentiella evaporationen visas i Tabell 3. Om en variabel kunde användas som indata så gjordes detta.

Tabell 3. Variabler som användes när Python paketet PyET användes för att beräkna daglig potentiell evaporation från Triangeldeponin. Enhet som anges är den enhet som indata ska anges i när PyET används för att beräkna den potentiella evaporationen.

Variabel	Beskrivning	Förkortning i skript	Enhet
Medellufttemperatur	Dagligt medelvärde av lufttemperatur, mätt 1,65 m över marken	tmean	°C
Maximal lufttemperatur	Daglig maximal lufttemperatur, mätt 1,65 m över marken	tmax	°C
Minimal lufttemperatur	Daglig minimal lufttemperatur, mätt 1,65 m över marken	tmin	°C
Vindhastighet	Dagligt medelvärde av vindhastighet, korrigerat till 2 m över marken	wind	$m\ s^{-1}$
Relativ luftfuktighet	Dagligt medelvärde av relativ luftfuktighet, mätt 1,65 m över marken	rh	%
Maximal relativ luftfuktighet	Daglig maximal relativ luftfuktighet, mätt 1,65 m över marken	rhmax	%
Minimal relativ luftfuktighet	Daglig minimal relativ luftfuktighet, mätt 1,65 m över marken	rhmin	%
Nettostrålning	Daglig nettostrålning mätt vid 2 meter	rn	$MJ\ dag^{-1}\ m^{-2}$
Värmeflöde i marken	Dagligt nettovärmeflöde	g	$MJ\ dag^{-1}\ m^{-2}$
Altitud	Mätstationens höjd över havet	elevation	m ö. h.
Latitud	Mätstationens (alt. Triangeldeponins) breddgrad	lat	radian

Flertalet beräkningsmetoder kan använda sig av fler variabler, till exempel lufttryck, mättnadsångtryck och soltimmar. Om dessa variabler inte angavs så beräknas dessa av inbyggda funktioner i paketet (tagna framförallt ifrån Allen et al., 1998) eller gavs ett standardiserat värde.

För att välja ut en beräkningsmetod för att beräkna PE togs det fram kriterier för att särskilja mellan metoderna för att slutligen välja en. Beräkningsmetoden skulle:

- Kunna beräkna PE med befintliga observationer.
- Om jämförbara resultat kunde uppnås med färre variabler var detta att föredra.

- Inte över- eller underestimera den potentiella evaporationen.

Python-koden för att beräkna PE samt ta fram figurer enligt de olika metoderna visas i bilaga B.

Utöver de beräkningsmetoder som användes i Python, utfördes ytterligare en beräkning av PE, kallad "totalenergi-metoden". Denna metod är inte allmänt vedertagen utan är baserad på energibalansen (se ekvation [1]) och antagandet att all energi som når marken och inte försvinner genom markvärmeflödet går till att evaporera vatten. Totalenergi-metoden tar endast hänsyn till energiflöden och beräknas enligt följande formel

$$\begin{cases} PE = \dfrac{R_n - G}{L}, & om\ R_n - G > 0 \\ PE = 0, & om\ R_n - G < 0 \end{cases} \tag{21}$$

där L (ångbildningsvärme) är beräknad enligt rekommendationer i FAO-56 (Allen et al. 1998). Metoden använder dygnsvärden och kan ses som en övre gräns för vad den potentiella evaporationen kan vara, eftersom den beräknar den totala energin tillgänglig för evaporation. Metoden fungerar som en riktlinje och tar dock inte hänsyn till det sensibla flödet av energi som också är en del av energibalansen (se ekvation [1]).

4.3 Konceptuell modell

I mening av ”konceptuell modell” är det en modell som använder både av empiriska och fysikaliska ekvationer. Den konceptuella modellen som byggdes i denna studie är indelad i olika delar, där varje del representerar en viss hydrologiska process som sker i naturen. Eftersom det var vattenbalansen vid (mark)ytan som skulle modelleras delades modellen upp i två delmodeller: en snörutin och en markrutin.

4.3.1 Snörutin

Modellen består av två skript: ett som snörutinen är byggd i och ett som kalibrerar de fem parametrar som skickas som indata till snörutinen. Snörutinen baseras på graddagsmetoden så som den är beskriven i avsnitt 2.2.2 och som den är implementerad i HBV (Hydrologiska Byråns Vattenbalansavdelning). (HBV är ett hydrologiskt modellprogram som används för att simulera och förutsäga vattenbalansen i ett avrinningsområde [Bergström, 1992; Seibert, 2005].)

Efter att skriptet initierats med de fem parametrarna, hämtas dagliga värden av nederbörd, lufttemperatur och SWE från sammanställning av data från den meteorologiska mätstationen på Triangeldeponin. Variabler som håller koll på vatten- och snömängder initieras sedan. Efter att variablerna initierats kommer en ”for-loop” som för varje tidssteg i data beräknar flödena av snö och vatten. I for-loopen används T_T (tröskeltemperaturen) för att dels bestämma om nederbörd faller som snö eller regn, dels för att bestämma om snö smälter eller om flytande vatten i snön återfryser och till vilken omfattning detta sker. Om flytande vatten överskrider C_{WH} (snötäckets vattenbärande kapacitet) infiltrerar vatten marken och ”lämnar” därmed delmodellen. Om det inte finns någon snö på marken så infiltrerar all tillgängligt flytande vatten från regn och snösmältning direkt i samma tidssteg. Efter beräkning av mängd vatten som infiltrerar marken och uppdatering av SWE (flytande vatten i snötäcket inkluderas i beräkningen) har en iteration av for-loopen körts.

För att parametrarna i snörutin ska kunna kalibreras (se avsnitt 4.3.3) finns ett anrop till en funktion (som återfinns i ett samlat skript för mindre funktioner, se bilaga D) som beräknar Nash-Sutcliffe effektivitetskoefficient (*Nash-Sutcliffe model efficiency coefficient*, NSE) som beräknas enligt

$$NSE = 1 - \frac{\sum_{i=1}^{n}\left(Q_{obs,i} - Q_{sim,i}\right)^2}{\sum_{i=1}^{n}\left(Q_{obs,i} - \overline{Q}_{obs}\right)^2} \quad (22)$$

där $Q_{obs,i}$ är den observerade värdet av variabeln vid tidpunkt i, $Q_{sim,i}$ är den simulerade värdet av variabeln vid tidpunkt i, $\overline{Q}_{obs}$ är medelvärdet av de observerade värdena, och n är antalet observationer. Det finns möjlighet att ha en viktad beräkning av NSE som bara tar i hänsyn vissa tidssteg. Skriptet returnerar beräknat NSE för den aktuella körningen. Skriptet visas i sin helhet i bilaga D.

Ett antagande som gjorts vid skapandet av snörutinen var att ingen evaporation (sublimation) sker från snötäcket eller vatten vid ytan, utan detta sker endast i marken – i markrutinen.

4.3.2 Markrutin

Likt snörutinen består markrutinen av två skript: ett där markrutinen är byggd i och ett skript som kalibrerar markrutinens parametrar. Till en början skapades en markrutin som baseras på hur markvattenflödet sker enligt en HBV-manual. I HBV så delas det infiltrerade vattnet upp i att antingen gå till markfuktighet (S_{sm}) eller grundvattenbildning (Q, *recharge*) (alternativt benämning: markvattenflöde), enligt följande ekvation

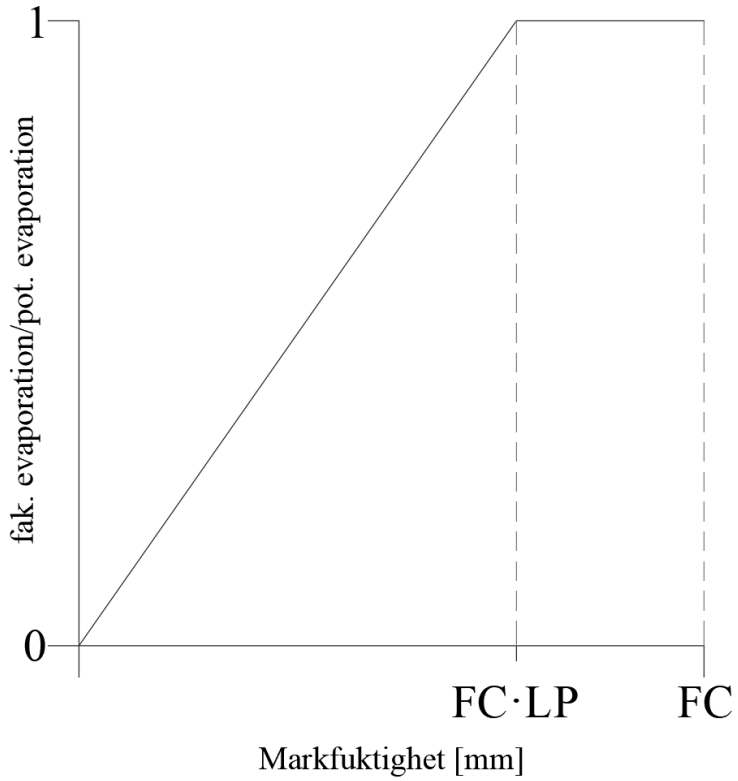
$$\frac{Q}{I} = \left(\frac{S_{sm}}{FC}\right)^{\beta} \quad (23)$$

där I är det infiltrerade vattnet från snösmältning eller regn, FC (fältkapacitet) är det modellerade maximala markfuktigheten, och β är en parameter som beskriver relative fördelningen av infiltrerat vatten till grundvattenbildning och markvattenlagret (se *Soil moisture routine* i Seibert, 2005). För den del av gråbergsdeponin som denna studie undersöker, sker ingen grundvattenbildning; Q tolkades i stället som perkolationsflödet. Det bör noteras att S_{sm} och FC är absoluta värden och skiljer sig därför från markfuktigheten (markvattenhalten) som anges i avsnitt 5, där det anges i procent.

Markrutinen i HBV hanterar även evaporation. Faktisk evaporation (FE) beräknas som en kvot av PE enligt följande ekvation

$$FE = \min\begin{cases} PE \cdot \left(\frac{S_{sm}}{FC \cdot LP}\right) \\ PE \end{cases} \quad (24)$$

där LP är en kalibreringsparameter som antar ett värde mellan 0 och 1. Från ekvationen fås att när $S_{sm} = FC \cdot LP$ uppnår FE sitt maximala värde, PE. Detta förtydligas i Figur 7.



Figur 7. Faktisk evaporation som funktion av markvattenlagret. FC är det maximala markvattenlagret och LP är den fraktion vid vilket den faktiska evaporationen får samma värde som den potentiella evaporationen.

Implementationen av ekvation (23) visade sig vara ineffektiv eftersom utflöde ur markrutinen endast inträffade under de tidssteg där ett inflöde (infiltration) förekom. Dessutom fördelas inflödet mellan marken och perkolation, vilket innebär att vatten ackumuleras i marken över tid om FE inte var större än den del av inflödet som ”stannade” kvar i marken. När modellen byggdes om för att inkludera flera lager, skedde alltid en ackumulering i de undre marklagren där inget utflöde i form av evaporation förekom. Därför beslutades det att implementera en annan ekvation för att hantera utflödet ur ett lager enligt

$$Q = k \cdot (S_{sm} - S_{min}) \tag{25}$$

där *k* är den linjära lagringskoefficienten som avgör storleken av perkolationen och S_{min} är den lägsta nivån för markfuktigheten vid vilket perkolation upphör. HBV:s sätt att hantera evaporation hölls dock intakt.

Markrutinskriptet initieras med modellens kalibreringsparametrar och några olika val, där användaren får, bland annat, valmöjlighet att välja hur många lager (1 – 3) som ska modelleras. Som första steg i modellen laddas PE enligt Hargreaves metod (se avsnitt 285.1 och 6.1 för förklaring varför denna metod valdes) och infiltrationsdata från snörutinen in i modellen. Beroende på antalet lager som användaren valt att modellera, laddas relevant markfuktighetsdata in för det/de valda lagren, observerad av den meteorologiska mätstationen. Modellvariabler initieras som lagrar modellerad markfuktighet, grundvattenbildning och FE. Efter att variablerna initierats kommer en for-loop som för varje tidssteg i angiven datumlista beräknar markvattenlagrets storlek, grundvattenbildning och FE. En ”under”-for-loop delar upp varje tidssteg i mindre inkrement beroende på storlek på variabeln Δt, detta för att öka upplösningen och göra modellen mer stabil. Om flera lager modelleras används ekvation (25) för att modellera flödet mellan lagren såväl som ut från det lägsta lagret, men kalibreras var för sig. Vattenflödet, så som det är implementerat i modellen, är endast riktat vertikalt nedåt och kan inte flöda i

motsatt riktning. Variabelnamnen är lika för de olika lagren men det är flödet ut från det djupaste lagret som presenteras som perkolationsflöde i avsnitt 5.4. Det är endast från det översta lagret som ett utflöde ur modellen sker genom evaporation. För att kunna ta hänsyn till fler lager än ett i kalibrering av modellen, beräknas Normaliserad Nash-Sutcliffe effektivitetskoefficient (*Normalized Nash-Sutcliffe model efficiency coefficient*, NNSE) för lager x enligt

$$NNSE_x = \frac{1}{2 - NSE_x} \tag{26}$$

Efter beräkning av NNSE enligt ekvation (26), antingen genom en viktad eller oviktad beräkning, tas ett geometriskt medelvärde av NNSE från varje lager enligt

$$NNSE = \sqrt[n]{NNSE_1 \cdot NNSE_2 \cdot \ldots \cdot NNSE_n} \tag{27}$$

där n är antalet lager. Från skriptet returneras NNSE, beräknad enligt ekvation (27). Skriptet visas i sin helhet i bilaga D.

4.3.3 Kalibrering

För att kalibrera de båda modellerna användes två skript som funktionellt sett är likadana men skiljer sig i vilka parametrar (vilken rutin) som ska kalibreras, vilka startvärden parametrarna har och inom vilka intervall som de tillåts variera. Skripten inleds med att importera funktionen ”differential_evolution” (sv. differentiell utveckling), vilket låter användaren optimera ett problem genom en iterativ process tills ett minimum är funnet. En objektiv funktion (*objectiv function*) är sedan skapad, den innehåller parametrarna som ska optimeras och ett anrop till aktuellt skript vilket returnerar NSE (NNSE för markrutinen). Eftersom differentiell_evolution optimerar för ett minimum, returnerar objektiv funktionen -NSE, vilket då resulterar i att NSE maximeras. Efter objektiv funktionen definieras startvärden för parametrarna och inom vilka intervall, parametrarna får variera. Sedan körs funktionen differential_evolution med objektiv funktionen och parametervärden som input. Från detta kalibreras parametrarna och bästa resultat skrivs ut.

I Tabell 4–5 visas undre och övre gräns för det intervall som parametrarna får variera inom, här anges även startvärdet för varje parameter. C_{WH} och C_{FR} har inte blivit kalibrerade utan deras värden blev satta till vara konstanta värden, 0,1 respektive 0,05 (se Seibert 2005), då båda parametrarna fick orimliga värden vid kalibrering.

Tabell 4. Undre och övre gräns samt startvärde för kalibrering av snörutinsparametrar.

Parameter	Undre gräns	Övre gräns	Startvärde
C_{fmax}	1	8	3
T_T	-2	2	0
C_{WH}	0,1	0,1	0,1
C_{FR}	0,05	0,05	0,05
S_{FCF}	0,1	2	1

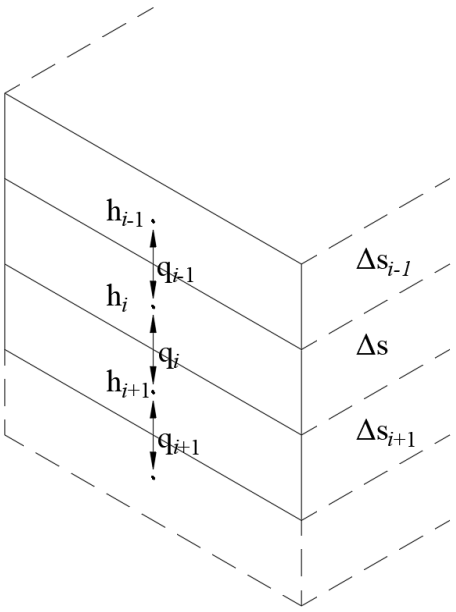
Tabell 5. Undre och övre gräns samt startvärde för kalibrering av markrutinsparametrar.

Parameter	Undre gräns	Övre gräns	Startvärde
FC	0	2	1
LP	0	1	0,5
k	0	1	0,1
k_2	0	1	0,1

Redan innan studiens början stod det klart att Triangeldeponin påverkades av bortförelse av snö med vinden. Ingen implementering av omfördelning av snö gjordes, vilket innebar att det fanns två sätt att kalibrera snörutinen: antingen skulle modellen kalibreras för att anpassas till ”snötopparna” och efterkommande snösmältning, eller skulle modellen kalibreras för hela mätperioden. Med snötoppar menas det tydliga toppar i data som både kommer fort och försvinner fort, som skiljer sig från snöperioder som ackumuleras sakta och ligger kvar länge. Det valdes att göra det förstnämnda då det var svårt att hitta bra parametrar som passade över hela perioden och för att det ansågs viktigare att få infiltration från snösmältning korrekt innan perioder med högre potentiell evaporation och perkolation än snödynamiken under den kalla delen av vintern. I data fanns det tre tydliga snötoppar under mätperioden som det kalibrerades för: 2021-10-28–2021-11-03, 2022-04-10–2022-04-25 och 2023-04-30–2023-05-09.

4.4 Fysikalisk modell

Vad som avses med en ”fysikalisk modell” i denna rapport är en modell som består av kontrollvolymer (lager) där beräkningar för in- och utflöden utförs för varje kontrollvolym (se Figur 8). Den fysikaliska modellen anses allmänt mer komplex än den konceptuella modellen, där komplexare och fler ekvationer beräknas. Att sätta upp en fysikalisk modell är därför mer komplicerat och en körning med modellen är mer krävande.



Figur 8. Ett exempel på hur en fysikalisk modell kan se ut. Uppdelning av markprofilen i olika diskreta lager (i) med olika tjocklek (Δs), tryckpotential (h) och flöde (q) för ett visst tidssteg.

Som fysikalisk modell byggdes en modell i programmet CoupModel (se Jansson och Karlberg, 2011).

4.4.1 Programbeskrivning

CoupModel är ett avancerat modelleringsprogram som används för att simulera hydrologiska och termiska processer i ”mark-växt-atmosfärs systemet”. CoupModel utvecklades från SOIL-modellen med fokus på skogsekosystem, men har sedan anpassats för att modellera olika typer av markmiljöer med inkluderade kol- och kväveprocesser. Prefixet ”Coup” är hämtat från engelskans *coupled* (sv. kopplade), där modellen består av flera kopplade delmodeller (Jansson & Karlberg 2011).

Centralt för Coupmodellen är de två delmodellerna som beskriver vattenflödet och värmeflödet genom markprofilen, som kopplar Richards ekvation för vattenflödet och Fouriers ekvation för värmeflöde i 1D (Jansson & Karlberg 2011).

Programmet ger användaren möjlighet att ändra förutsättningar för en simulering genom kategorierna ”Switches”, ”Parameters”, ”Parameter Tables”, ”Model Files”, ”Output Variables” och ”Validation”: Switches låter användaren välja vilka delmodeller som ska inkluderas i modellen, hur data ska laddas in eller uppskattas i modellen och randvillkor. Parameters låter användaren ge enskilda parametrar, som är kopplade till de olika delmodellerna, värden. Parameter Table låter användaren fylla i parametervärden för modellerade eller observerade lager, till exempel lagerdjup, hydraulisk konduktivitet och retentionskurva. I Model Files får användaren välja datafil med indata till modellen och en valideringsfil som används i Validation. Output Variables låter användaren välja vilka variabler som modellen ska ge som output. I Validation får användaren välja att koppla mätdata från angiven fil i Model Files och simulerad variabel från modellen. Programmet har inbygg funktionalitet för jämförelse av observerade och simulerade värden (se Jansson & Karlberg 2011).

Programmet ger användaren möjlighet att modellera ett makroporflöde (se avsnitt 2.2.3.3) utöver ett vattenflöde genom matrisen. Ett makroporflöde simuleras när vattenhalten är större än skillnaden av totalporositeten och makroporositeten ($\theta > \theta_s - \theta_m$). Användaren kan ange en hydraulisk konduktivitet för matrisen och en för hela markprofilen när vattenhalten överstiger matrisporositeten (”total konduktivitet”). Det finns även möjlighet att modellera ett bypass flöde (*bypassflow*) (se Jansson & Karlberg 2011).

Programmet kan köras med ”singelrun” eller ”multirun”. En singelrun innebär att programmet körs en gång med den uppsättning av parametrar som är angivna. En multirun tillåter användaren att köra programmet många gånger i rad, där vissa förvalda parametrar ändrar värde mellan körningarna enligt olika kalibreringsmetoder såsom GLUE eller Bayesian (se Jansson & Karlberg 2011).

Det bör noteras att CoupModel kan modellera mark-växt-atmosfärssystemet med en maximal partikelstorlek på 2 mm (sand). Inställningar kan göras för att detta ska ha mindre påverkan på resultatet genom att till exempel välja vattenretentionsparametrar manuellt. En aspekt som dock inte kan påverkas är kapillärstigning, som är korrelerad med partikelsammansättningen (se Beskow 1929). Detta kan leda till att ett vertikalt flöde uppåt modelleras som inte återfinns i verkligheten.

4.4.2 Arbetsgång

Ett nytt dokument öppnades i CoupModel med förinställda inställningar för en simulering av värmeflöde genom en godtycklig markprofil. För att modellera vattenbalansen på Triangeldeponin behövdes många inställningar och parametervärden ändras i CoupModel.

I fliken ”Run info” ställdes simulationsperioden in med startdatum 2021-10-01 00:00 och slutdatum 2023-06-01 00:00. "PrePeriodDate" och "PostPeriodDate" ställdes in på samma datum som start- och slutdatumet. Modellens tidsupplösning valdes till att vara per timme med 96 iterationer per dag.

Därefter gjordes inställningar under fliken Switches. Det är viktigt att ställa in dessa innan andra inställningar eftersom de kan påverka vilka alternativ som är tillgängliga under andra kategorier. En översikt över viktiga inställningar i Switches visas i Tabell 6. Modul (*Module*) anger till vilken modul inställningen tillhör, namn (*Name*) är namnet på inställningen, alternativ (*Option*) är där själva inställningen görs, och standardinställning (*Default*) visar standardinställningen i

programmet. Alternativ och standardinställningar har översatts till svenska. Inga andra inställningar ändrades från standardinställningarna förutom de som anges i tabellen.

Tabell 6. Uttömmande lista på inställningar ändrade från standardinställning under fliken "Switches" i CoupModel. Ett fåtal utvalda inställningar som inte har förändrats finns även med.

Modul	**Namn**	**Alternativ**	**Standardinställning**
Additional Variables	Soil Temp Sensors	På	Av
	Soil Theta Sensors	På	Av
Drainage and Deep Percolation	LBoundSaturated	Enhet Grad flöde	Konstant tryckpotential
Meteorological Data	CloudInput	Estimerad	Läs från fil
	HumRelInput	Läs från fil	Genererad av parametrar
	PercInput	Läs från fil	Genererad av parametrar
	RadGlobInput	Estimerad	Läs från fil
	RadNetInput	Läs från fil	Estimerad
	TAirGlobRad	Användes inte	Användes
	TempAirCycle	Årligen	Dagligen
	TempAirInput	Läs från fil	Genererad av parametrar
	VapourAirInput	Som relativ luftfuktighet	Läs som tryck
	WSpeedInput	Läs från fil	Läs från fil
Model Structure	Evaporation	Strålningsinput stil	Av
	Heat Equation	På	På
	Snowpack	På	Av
	WaterEq	På med fullständig markprofil	Av
	Plant Type	Ingen vegetation	Ingen vegetation
SnowPack	SnowAdjustment	Tvingad att matcha kontinuerlig	Ingen korrektion
Soil evaporation	Evaporation Method	Iterativ energibalans	Inte estimerad
	Surface Temperature	f(E-balans lösning)	Lufttemperatur
Soil heat flows	Initial Heat Conditions	Temp(z)-Estimerad	Uniform temperatur
Soil water flows	Initial Water Conditions	Vatteninnehåll(z)	Uniform tryckhöjd
Technical	TimeResolution	Per timme	Inom dagen (tidssteg)
	TypeOfDrivingFile	Standard körfil	Ingen standard klimatfil
	ValidationMode	En	Av

Det fanns ingen inställning som kunde göras för att ta hänsyn till vindens bortförande av snö i CoupModel. Det fanns dock en inställning som tvingar modellerat att följa en datalista med snödjup som användaren ger (se *Snowadjustment* i Tabell 6). Programmet hanterade tillförsel och borttagning av snö genom en snönederbördskorrektion så att modellerad snödjup anpassade sig till den kontinuerliga datalistan. Snökorrigeringen sker under vilket tidssteg som helst för att anpassa

modellerat snödjup och kan då vara negativ, vilket representerar ett flöde från marken. Den datalista som angavs var observerat dagligt snödjup på Triangeldeponin.

En empirisk ekvation (som liknar graddagsmetoden i stor utsträckning, såsom den presenteras i avsnitt 2.2.2) användes för att beräkna snösmältning trots möjligheten att använda en mer fysikalisk ekvation. Detta val gjordes eftersom anpassningen av marktemperaturen blev betydligt sämre med den mer fysikaliska ekvationen jämfört med den empiriska. CoupModels fysikaliska ekvation är baserad på energibalansen och liknar den ekvation som presenteras i ekvation (13) (se Jansson & Karlberg 2011).

Inställningarna av de andra kategorierna gjordes i en godtycklig ordning, men de presenteras i ordningen som de ges i programmet. I Tabell 7 visas de parametervärden som avviker från dess standardvärden. ”PrecA0Corr” och ”PercA1corr” anger korrektioner på nederbörd som regn respektive snö. Ingen korrektion gjordes i körningen på grund av att det redan skett en korrektion av nederbörd innan den användes i CoupModel. Med ”SurfCoef ” anges hur mycket vatten som rinner bort genom ytavrinning, i denna studie har ett antagande gjorts att ingen ytavrinning sker, därför har SurfCoef fått värdet noll.

Tabell 7. Parametrar med andra värden än standardvärden.

Modul	Namn	Värde	Standardvärde	Enhet
Meteorological Data	AltMetStation	555	0	m
	AltSimPosition	555	0	m
	PrecA0Corr	1	1,07	-
	PrecA1Corr	0	0,08	-
	TempAirMean	4,02	10	°C
Soil evaporation	EquilAdjustPsi	1,9	1	-
Soil frost	FreezepointFWi	2	0,5	-
Soil water flows	InitialWaterContent	10	20	%
Surface water	SurfCoef	0	0,8	-
Radiation properties	Latitude	67,8	58,5	-

I programmet skapades en markprofil i ”Soil - Profile” baserat på partikelsammansättning som visas i Tabell 2. Alla partiklar större än sand tilldelades den största tillgängliga partikelstorleken, vilket i detta fall var sand. Detta innebar att nästan hela partikelsammansättningen (99,6 %) bestod i den modellerade markprofilen av sand. Profilen skapades med tre ”uppmätta” lager med samma hydrauliska egenskaper och partikelsammansättning, där djupet var samma som lagrens djup i den konceptuella modellen. Det bör nämnas att de ”uppmätta” lagren inte representerar verkliga lager och inga riktiga mätningar som motsvarar dessa lager har gjorts. Antalet lager och deras djup var relaterade till antalet nivåer av TDR-sensorer och deras placering i marken (se lagrens djup i Tabell 8).

Till Parameter Tables importerades ovan nämnda markprofilen. Profilen innehöll information om det antal ”uppmätta” lager, deras djup och parametrar som beskriver retentionskurvan enligt Brooks-Corey (se avsnitt 2.2.3.4). De uppskattade hydrologiska egenskaperna för den importerade markprofilen kunde inte återge markvattenhalter och marktemperaturer som observerats, därför gjordes en kalibrering av Brooks-Coreys parametrar med en multirun, där kalibrering gjordes efter

bästa NSE och R^2. I Tabell 8 visas de värden som togs fram genom kalibrering som användes för att ta fram modellresultatet.

Tabell 8. ”Brooks-Corey, water retention, measured horizon”. Lagerdjup och parametrar för retentionskurva enligt Brooks-Corey.

Index	**Övre djup**	**Nedre djup**	**λ**	**h_a**	**θ_s**	**Vissningsgräns**	**θ_r**	**θ_m**	**Övre gräns**
	[m]	[m]	[-]	[cm]	[%]	[%]	[%]	[%]	[cm]
1	0	0,346	0,95	45	30	1,9	0	10	10 000
2	0,36	0,876	0,95	45	30	1,4	0	10	10 000
3	0,876	1,2	0,95	45	30	1,4	0	10	10 000

I programmet är en annan tabell, ”Brooks-Corey, water retention, model layers”, direkt beroende av värdena i Tabell 8. Användaren får välja antal beräkningslager (med tillhörande djup för varje lager) där Brooks-Corey-retentionsparametrar uppdateras baserat på vilket ”uppmätt” lager det modellerade lagret befinner sig i. Till exempel, om det modellerade beräkningslagret med index 2 har ett övre djup på 0,1 m och ett nedre djup på 0,2 m, befinner det sig fullständigt inom det "uppmätta" lagret med index 1 och får då samma retentionsparametrar som rad 1 i Tabell 8. Totalt modellerades 20 beräkningslager, där de första åtta hade ett djup på 0,1 m, de nästkommande fyra lager hade ett djup på 0,2 m och resterande, åtta lager hade ett djup på 0,4 m. Det totala djupet på den modellerade markprofilen var alltså 4,8 m. Anledningen till att ett större djup modellerades i CoupModel än i den konceptuella modellen var för att komma under tjälfritt djup. Modellerade beräkningslager djupare än ”uppmätta” lager fick samma värde som det lagret med index 3.

Från markprofilen laddades också värden på hydraulisk konduktivitet in för de tre ”uppmätta” lagren, som uppskattats av CoupModel från partikelsammansättningen. Efter kalibrering uppdaterades värdet på matris konduktiviteten till 300 mm/dag för lager 1 och 600 mm/dag för lager 2 och 3 (se Tabell 8) och total konduktivitet till 10 000 mm/dag för samtliga lager. Tortuositet som anges i samma tabell i programmet, ändrades inte från sitt standardvärde (1 för samtliga lager).

I Parameter Tables ”placerades” tre fiktiva temperatursensorer och tre fiktiva markfuktighetsmätare vid djupen 9, 62 och 115 cm, de representerar de sensorer som fanns i marken på Triangeldeponin (se Figur 20 i bilaga A). Detta gjordes för att underlätta jämförelse av modellerade med observerade temperaturer och vattenhalter i marken.

Under Model Files behövdes tre filer anges: en innehållande meteorologiska data som används som indata till modellen, en med endast snödjup och en med valideringsdata. Den meteorologiska datafilen innehöll nederbörd [mm], lufttemperatur [°C], vindhastighet [m/s], relativ luftfuktighet [%] och nettostrålning [J/m²/dag]. Filen med valideringsdata innehöll värmeflöde i marken [J/m²/dag], snödjup [m], marktemperatur [°C] vid tre olika djup och markfuktighet [%] vid tre olika djup.

Under Output fliken valdes alla typer av vattenflöden för alla lager (inklusive vattenlagring och vattenbalanskontroll), alla värmeflöde- och temperaturrelaterade outputs.

Under Validation kopplades de fiktiva sensorerna ihop med data från de verkliga sensorerna (som angavs i filen under Model Files) så att modellen kunde valideras. Dessutom kopplades observerat med simulerat värmeflöde i marken och observerat med simulerat snödjup.

För kalibrering användes MultiRun. För att köra en multirun behövdes först parametrar väljas, antingen från Parameters eller Parameter Tables. För varje parameter valdes ett lämpligt intervall som den ska variera inom under körningen. Vidare valdes metodalternativet ”Stochastic Linear”

för de parametrar som har ett litet intervall och ”Stochastic Log” för de parametrar som har intervall som sträcker sig flera tiopotenser. Desto fler parametrar som skulle variera desto större n (antal körningar under en multirun). Bästa parametrar hittades genom att använda inbyggd funktionalitet för att identifiera bästa körning utifrån prestandamåtten R^2 och NSE när jämförelser gjordes mellan observerad och modellerad markfuktighet och marktemperatur.

5 Resultat

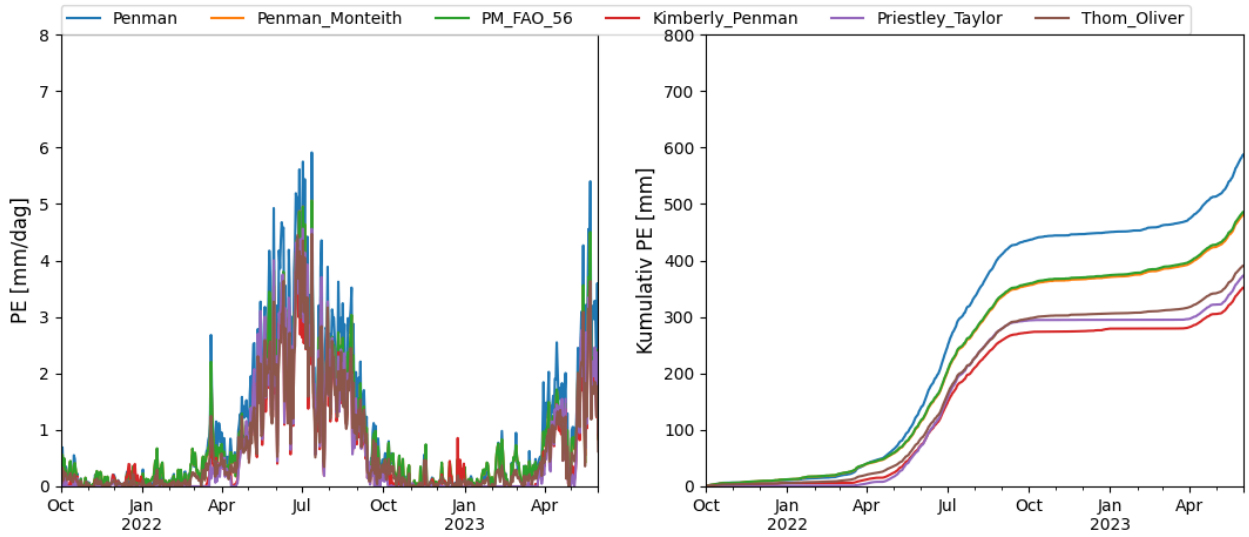
I detta avsnitt kommer det presenteras resultat från beräkningar av potentiell evaporation och från simuleringar av de båda modellerna. Avsnittet om beräkning av potentiell evaporation presenterar resultat från alla beräkningsmetoder, diskussion kring vilken beräkningsmetod som valdes att användes finns i avsnitt 6.1. Simuleringar av infiltration visar på mängden vatten som kommer in i marken. I avsnittet om markfuktighet och marktemperatur presenteras de kalibreringar som gjorts mot dessa variabler och därigenom hur väl de båda modellerna kunde simulera observerade variabler. Simulerad jämförelse av perkolation och evaporation från de båda modellerna presenteras sedan. Resultat anges både för ”hela mätperioden”, som avser period från 2021-10-01 till 2023-06-01, och för ”ett år”, vilket avser det första hydrologiska året som det mätts, från 2021-10-01 till 2022-10-01.

5.1 Beräkning av potentiell evaporation

Den potentiella evaporationen beräknades med 14 olika metoder: sex ”kombinationsmetoder”, tre ”strålningsmetoder” och fem ”temperaturmetoder”. Kombinationsmetoder kombinerar energibalansen med aerodynamiska faktorer för att beräkna PE., till exempel Penmans beräkningsmetod (ekvation [4]). Strålningsmetoder beräknas av strålningen men kan bero av andra parametrar, främst temperatur, till exempel Hargreaves beräkningsmetod (ekvation [5]). Temperaturmetoder är enkla metoder som främst baseras på lufttemperaturen och utgör ofta förenklade versioner av kombinationsmetoder.

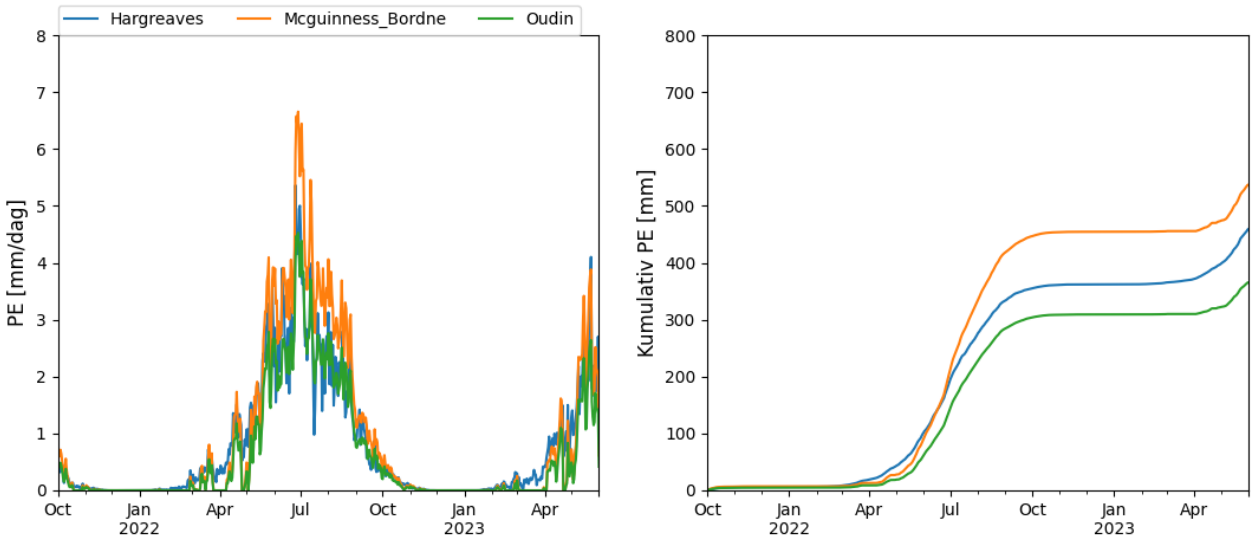
I Figur 9–11 visas daglig PE och kumulativ PE för perioden 2021-10-01–2023-06-01 (609 dagar), figurerna är indelade efter dessa grupper. Under vintern är PE generellt låg, nära noll, och stiger betydligt under sommaren. Potentiellt negativa värden (kondensation) beräknade av en metod är borttagna och ersatt med noll. I Tabell 9 visas en sammanställning av kumulativ PE för samtliga metoder.

Beräknad PE enligt kombinationsmetoder visas i Figur 9. Alla kombinationsmetoder är baserad på Penmans metod, och vidareutvecklingen av metoden har främst varit riktad mot att inkludera påverkan av en referensgröda på PE och standardisera vissa parametrar (Xiang et al. 2020). Penmans metod avviker tydligt från de andra metoderna där daglig PE är generellt högre under stora delar av året, vilket resulterar i en kumulativ PE på uppemot 600 mm sett över hela perioden. Penman-Monteith (orange linje) och PM-FAO56 (grön linje) är svåra att urskilja från varandra även i den kumulativa PE diagrammet. Att det är svårt att urskilja de båda metoderna är för att PM-FAO-56 är en utveckling av Penman-Monteith (i publikationen FAO-56, se Allen et al., 1998), och inga parametervärden har angivits som skulle betydligt skilja de båda metoderna åt. Kimberly-Penman, Priestley-Taylor och Thom-Oliver har alla tre liknande kurvor och beräknar liknande PE sett över hela perioden.



Figur 9. Kombinationsmetoder för bestämning av potentiell evaporation. Dagliga värden av potentiell evaporation och kumulativ potentiell evaporation för perioden 2021-10-01 – 2023-06-01.

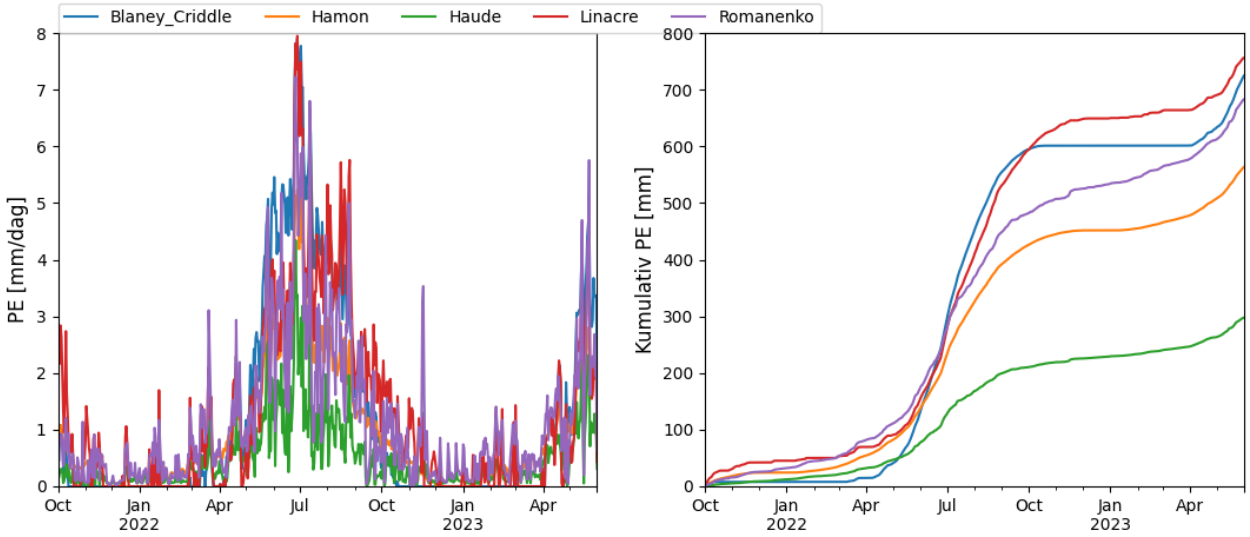
Beräknad PE enligt strålningsmetoder visas i Figur 10. Endast tre metoder i denna grupp finns beräknade då flera metoder kräver inkommande kortvågig strålning som indata vilket inte mättes av den meteorologiska stationen. De tre strålningsmetoderna är tydligt separerade: Oudin har lägst PE, följt av Hargreaves och McGuinnes-Bordne har högst PE. Mönstret för alla tre metoder liknar varandra mycket, både den dagliga och kumulativa PE diagrammen, vilket tyder på att en omkalibrering av standardparametrar skulle kunna resultera i liknande resultat med de olika metoderna.



Figur 10. Strålningsmetoder för bestämning av potentiell evaporation. Dagliga värden av potentiell evaporation och kumulativ potentiell evaporation för perioden 2021-10-01 – 2023-06-01.

Beräknad PE enligt temperaturmetoder visas i Figur 11. Det finns en stor variation i storleken på PE hos temperaturmetoderna: Blaney-Criddle, Linacre och Romanenko har alla ett högt kumulativt PE, men de skiljer sig när kommer till under vilken del av året som PE är störst relativt de andra. Blaney-Criddle beräknar PE nära noll i perioden oktober-april men beräknar relativt högt under resterande del av året. Romanenko visar en annorlunda trend där PE är relativt högt på vintern men inte alls lika högt under sommaren. Linacre beräknar ett relativt högt PE både under vintern och sommaren. Det finns flera variationer Hamon, den som valdes här är den metod som

bäst överensstämmer med övriga PE-metoder ovan (se ekvation [12] i Beran och Ansorge, 2019), de andra varianterna beräknar ett mycket högre PE. Haude-metoden, å andra sidan, har ett mycket lägre PE sett över hela perioden jämfört med de andra temperaturmetoderna.



Figur 11. Temperaturmetoder för bestämning av potentiell evaporation. Dagliga värden av potentiell evaporation och kumulativ potentiell evaporation för perioden 2021-10-01 – 2023-06-01.

En sammanställning av kumulativ PE efter hela mätperioden och efter ett år visas i Tabell 9. Jämförelsen efter ett år är särskilt intressant eftersom vissa metoder beräknar ett relativt högt PE på vintern och relativt lågt PE på sommaren, och vice versa. Temperaturmetoderna avviker från de andra metoderna, där Blaney-Criddle och Linacre har betydligt högre beräknat PE värde än alla andra, även Romanenko har högt PE. Haude är metoden har det lägsta beräknade PE av alla metoderna. Hargreaves metod har beräknat PE väldigt lika Penman-Monteith och PM-FAO-56. Penman, McGuinnes och Hamon har beräknat PE ganska lika varandra, medan Kimberly-Penman, Oudin, Priestley-Taylor och Thom-Oliver har beräknat ganska lika varandra.

I tabellen har ytterligare en metod, ”Totalenergi” (se ekvation [21]), lagts till. Blaney-Criddle, Linacre och Romanenko beräknar PE långt över totalenergi-metoden, vilket kan ses som att dessa metoder tydligt överestimerar PE. Andra metoder (Penman, McGuinnes, Hamon) är nära men överskrider inte totalenergi-metoden.

Tabell 9. Kumulativ PE för 15 PE-metoder efter ett hydrologiskt år och efter ett år och åtta månader (slutet av tillgängliga data). Totalenergi representerar potentiell evaporation om all tillgänglig energi (bortsett från värmeflöde i marken) skulle gå till att evaporera vatten.

PE-metod	Kumulativ PE 2022-10-01	Kumulativ PE 2023-06-01
Kimberly-Penman	273	352
Penman	436	588
Penman-Monteith	356	481
PM-FAO-56	359	486
Priestley-Taylor	294	373
Thom-Oliver	297	391
McGuinness-Bordne	447	537
Hargreaves	355	459
Oudin	304	365
Blaney-Criddle	594	725
Hamon	426	564
Haude	210	298
Linacre	594	756
Romanenko	482	683
Totalenergi	467	608

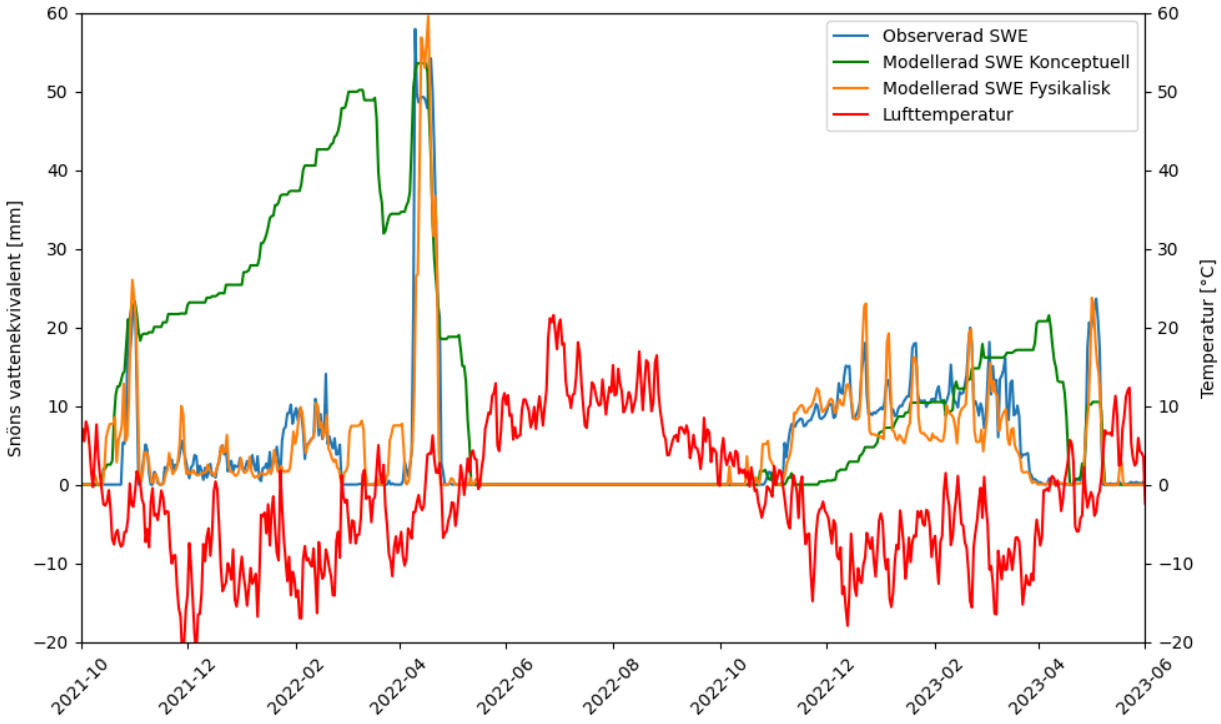
Följt av diskussion i avsnitt 6.1 valdes Hargreaves som beräkningsmetod som skulle uppskatta PE, som användes som indata i den konceptuella modellen.

5.2 Nederbörd och infiltration

Nederbörd och infiltration hör tätt ihop, speciellt när det inte finns snö och marken inte är frusen. Eftersom Triangeldeponin är snötäckt under stora delar av året påverkar det såväl tidpunkt för infiltration som mängden vatten som infiltrerar. Hur snödynamiken hanterades i de båda modellerna skiljdes åt, hur det påverkade snömängden och infiltrationen visas i detta avsnitt.

Från kalibrering av snörutinen (del av den konceptuella modellen, se avsnitt 4.3) fick parametrarna värdena $C_{FMAX} = 1{,}0$ mm/dag/°C, $T_T = -0{,}96$ °C, och $S_{FCF} = 0{,}33$. I Figur 12 visas modellerad SWE av snörutinen med dessa parametervärden. Det är tydligt att den modellerade snömängden skiljer sig stort från den observerade snömängden som mättes med snövåg. Två större snösmältningar sker innan topparna i april 2022 och maj 2023, detta när det inte finns någon snö observerad vid den meteorologiska stationen. Stora delar av snötoppen i april 2022 försvinner inte förrän temperaturen återigen stiger över tröskeltemperaturen. NSE, när endast de korta perioderna vid topparna tas i hänsyn, var 0,91 och för hela mätperioden -3.0.

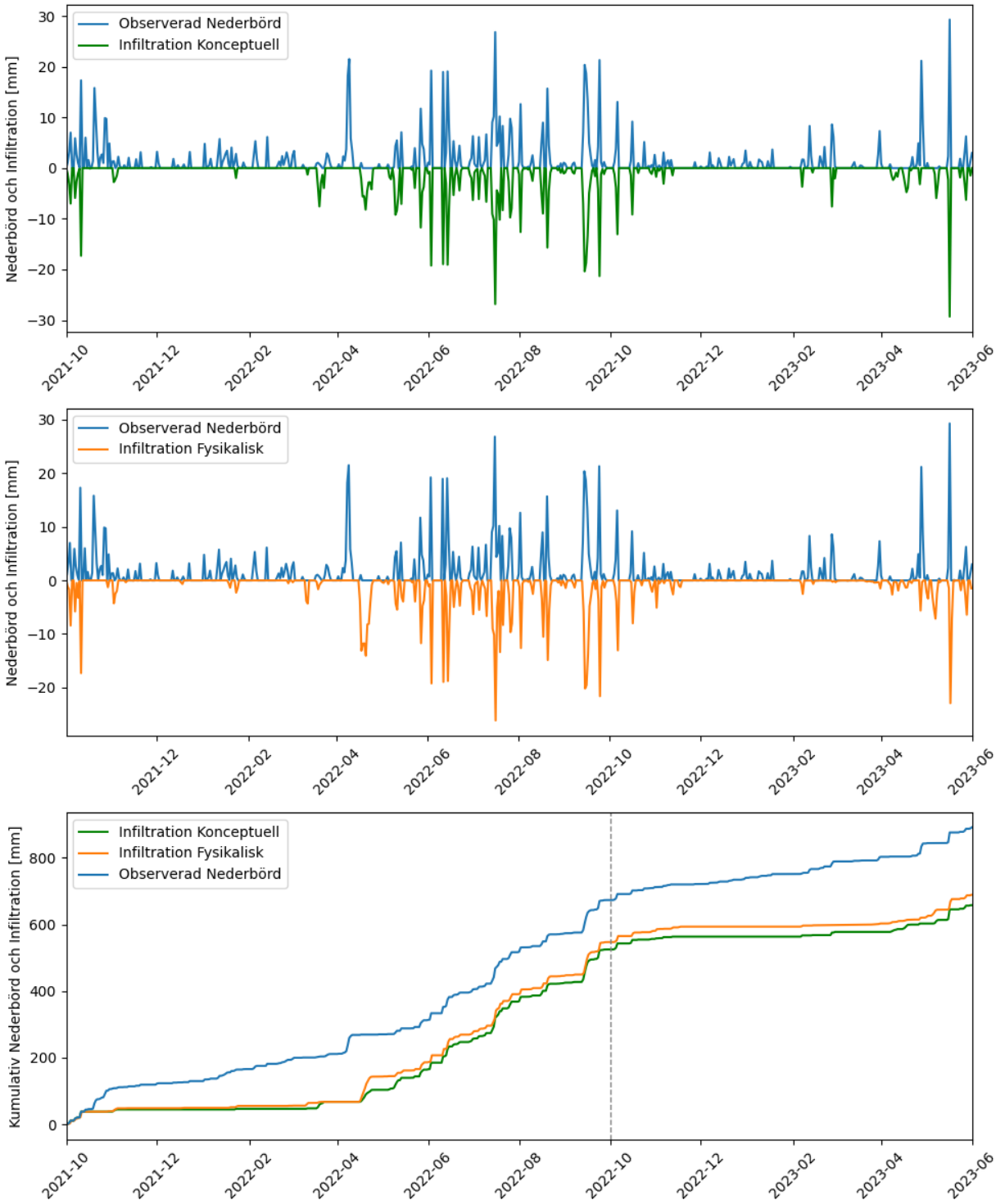
I figuren visas även modellerad SWE från den fysikaliska modellen. Det var betydande skillnader i hur snötäcket modellerades i CoupModel jämfört med den konceptuella modellen. Med de inställningar och förändringar som gjordes i data resulterade i ett NSE på 0,81 för hela mätperioden. Det finns, trots tvingandet att följa snödjup för varje tidssteg, ett antal små toppar modellerat som inte återfinns i observerade data. Den snönederbördskorrigering som modellen gjorde resulterade i totalt 228 mm snö togs bort under hela mätperioden och 126 mm efter ett år.



Figur 12. Modellerad snömängd (SWE) från den konceptuella modellen och den fysikaliska modellen. Modellerad snö jämförs med observerad snö samt lufttemperatur.

Från nederbörd och snösmältning modellerade de båda modellerna infiltration, vilket visas i Figur 13. Infiltration som den var modellerad av båda modellerna var inte påverkad av evaporation, utan evaporation påverkade vatten som redan fanns i markprofilen efter infiltration. Observerad nederbörd (indata till modellen) och modellerad infiltration följer varandra väl under snöfria delar av året för båda modeller. När snö finns och nederbörd faller som snö, sker nästintill ingen infiltration förrän temperaturen stiger och snösmältning sker. Infiltrationstoppar efter snösmältning motsvarar inte observerad nederbörd som fallit under hela snösäsongen, vilket är ett resultat av de parametrar och inställningar som valts i de båda modellerna.

För den konceptuella modellen uppgick infiltration till 658 mm och observerad nederbörd uppgick till 892 mm, detta är en skillnad på 234 mm vilket ska motsvara mängden vatten, i form av snö, som blåst bort. Efter ett år var modellerad infiltration 525 och observerad nederbörd var 673 mm, vilket ger en uppskattning på bortblåst snö till 148 mm. För den fysikaliska modellen var modellerad infiltration 664 mm sett över hela mätperioden och efter ett år var infiltrationen 547 mm. Ackumulering av infiltration av de båda modeller visas i den nedersta delfiguren, vilket visar på liknande dynamik och absolut mängd för båda modeller. Det framgår också tydligt att den största mängden nederbörd föll under perioden maj till oktober.



Figur 13. Modellerad infiltration för de båda modeller jämförs med observerad nederbörd. Den streckade vertikala linjen markerar där det hydrologiska året slutar. Året började 2021-10-01.

5.3 Markfuktighet och marktemperatur

Markfuktighetsobservationer användes för att kalibrera parameter för båda modellerna. I den konceptuella modellens markrutinen var det endast markfuktigheten som användes vid kalibrering av delmodellen. För den fysikaliska modellen kalibrerades det även för marktemperaturen - något den konceptuella modellen inte tog hänsyn till. Resultat från kalibreringar visas i detta avsnitt. TDR-sensorerna från vilken dessa variabler var observerade med fanns på djupen 9 cm, 62 cm och 115 cm (se Figur 20 i bilaga A).

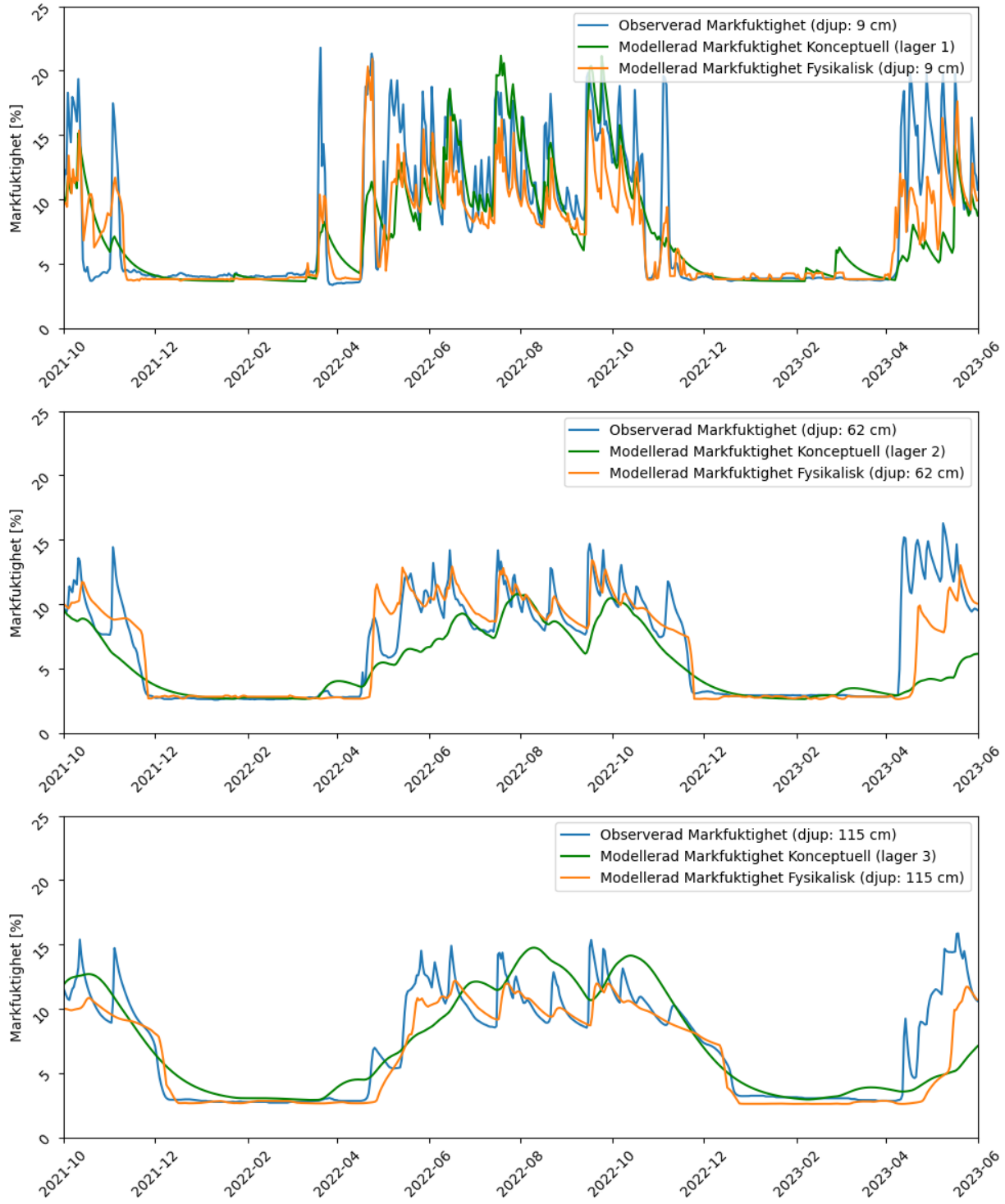
I den konceptuella modellen valdes det att modellera tre lager för att efterliknade tillgängliga data från de tre djup av markfuktighetsmätare, där lagerdjupen var lika som de lager som anges i Tabell 8. Från kalibreringen fick parametrarna värdena $FC = 1{,}0$, $LP = 0{,}58$, $k = 0{,}08$ och $k_2 = 0{,}0761$, där k_2 är den linjära lagringskoefficienten för lager 2 och 3. Med dessa parametervärden modellerades markfuktighet (se Figur 14).

I första lagret tycks modellen återspegla vattenhalten väl från december till mars och någorlunda väl under sommaren 2022. (Det framgår dock senare i rapporten att modellen endast har återgivit den flytande vattenhalten och inte den totala vattenhalten [se avsnitt 6.2.2].) Det tar ett tag för modellen att få samma markfuktighet i övergången mellan vinter och sommar som observerad data, vilket är tydligt både 2022 och 2023, detta kan även ses i övergången mellan sommar och vinter. I lager två och tre finns liknande årstidsvariationer som observerats men modellerade kurvor är mycket mer avrundade än observerade data. Det går att se att topparna förskjuts med ökat djup jämfört med observerad markfuktighet.

Även i den fysikaliska modellen användes observerad markfuktighet för att kalibrera och validera (se Figur 14). I stället för lager finns det möjlighet i CoupModel att direkt jämföra på ett visst djup där en sensor återfinns, här på 9 cm, 62 cm och 115 cm djup. Vid 9 cm djup följer modellerad markfuktighet ett liknande mönster som den observerade, där förändringar i observerad markfuktighet motsvaras av liknande förändringar i modellerad markfuktighet. Dock är modellerad markfuktighet generellt lägre under sommarhalvåret än vad som observerats.

Vid 62 cm djup följer modellerad och observerad markfuktighet varandra väl under stora delar av året. Det finns några tillfällen där det finns större skillnader, där vissa toppar i observerad markfuktighet inte motsvarats av toppar i den modellerade markfuktigheten, till exempel november 2021, november 2022 och april 2023. Topparna i modellerad markfuktighet är något avrundade jämfört med observerad markfuktighet.

Vid 115 cm djup finns ett generellt mönster där modellerad och observerad markfuktighet följer varandra. Modellerad och observerad markfuktighet skiljer sig speciellt i början av mätperioden och i slutet av mätperioden. Där skillnader på grundare nivåer har fortplantat sig nedåt i markprofilen. En tydlig skillnad är hur spetsiga markfuktighetstopparna är, där modellerad markfuktighet har ett mer avtrubbat utseende.



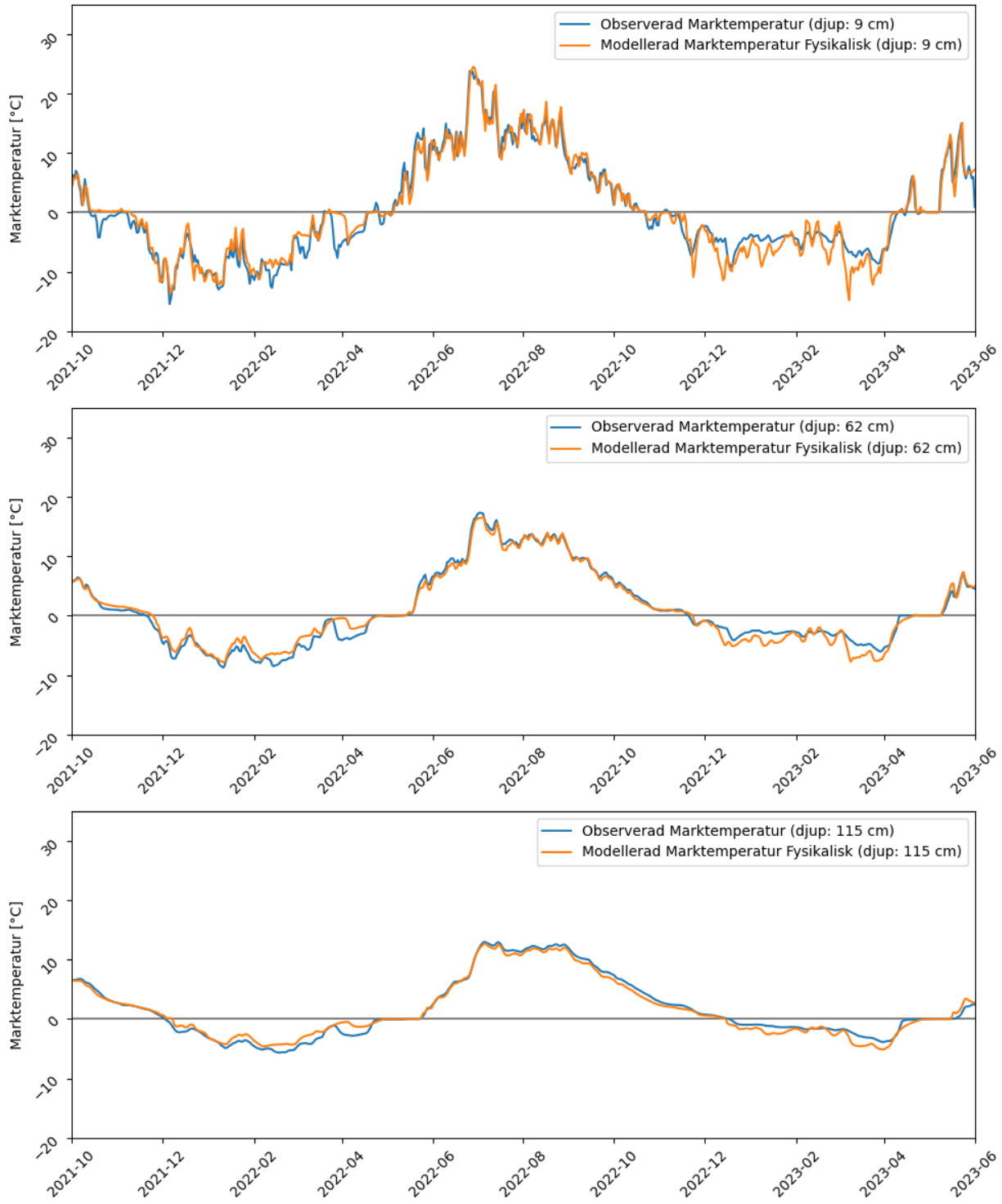
Figur 14. Modellerad markfuktighet i konceptuell modell i tre lager jämförs med modellerad markfuktighet i den fysikaliska modellen och observerad markfuktighet (9, 62 och 115 cm under markytan).

Marktemperaturen var en variabel som den konceptuella modellen inte modellerade eller tog hänsyn till, men som den fysikaliska modellen inkluderade. I den fysikaliska modellen användes observerad marktemperatur som en kalibreringsparameter. Även om marktemperaturen inte direkt påverkar vattenbalansen kan den vara en viktig faktor att beakta vid modelleringen.

Resultaten av marktemperatur från en körning av den fysikaliska modellen visas i Figur 15 , där modellerad och observerad marktemperatur jämförs vid 9, 62 och 115 cm djup. Vid 9 cm djup följer den modellerade marktemperaturen och den observerade marktemperaturen väl, även om det finns mindre skillnader mellan dem. Under vintern 2023 verkar marktemperaturen

underskattas, och modellen verkar inte kunna återspegla en sänkning i marktemperaturen som observerades i november 2021.

Vid 62 cm djup är den modellerade marktemperaturen väl anpassad till den observerade marktemperaturen. Den tidigare underskattningen under vintern 2023 vid 9 cm djup är mindre tydlig här. Tidpunkterna då marktemperaturen stiger över eller går under nollstrecket överensstämmer väl mellan de två. Samma sak kan sägas för marktemperaturen vid 115 cm djup.



Figur 15. Modellerad och observerad marktemperatur 9, 62 och 115 cm under markytan i den fysikaliska modellen.

NSE har beräknats enligt ekvation (22) för att utvärdera hur väl modellerna återger observerad markfuktighet och marktemperatur. De beräknade NSE-värdena presenteras i Tabell 10. NSE-värdena visar tydligt att den fysikaliska modellen återgivit observerad markfuktighet bättre än den konceptuella modellen. I tabellen presenteras även det geometriska medelvärdet av NNSE (se ekvation [26] och ekvation [27]) för de tre lager som optimerades under kalibreringen av den konceptuella modellen. För den fysikaliska modellen har även R^2-värden inkluderats, vilket användes vid utvärdering av en körning.

Tabell 10. Prestandamått på hur väl de båda modellerna återgivit observerad markfuktighet. För den fysikaliska modellen presenteras även prestandamått på hur väl modellen återgivit observerad marktemperatur.

Modell	Variabel	Lager/djup	R^2	NSE	NNSE
Konceptuell	Markfuktighet	Lager 1	-	0,55	0,67*
		Lager 2	-	0,39	0,67*
		Lager 3	-	0,56	0,67*
Fysikalisk	Markfuktighet	9 cm	0,81	0,72	-
		62 cm	0,75	0,74	-
		115 cm	0,75	0,74	-
	Marktemperatur	9 cm	0,95	0,92	-
		62 cm	0,97	0,97	-
		115 cm	0,98	0,98	-

*Geometriskt medelvärde för alla lager

5.4 Perkolation och faktisk evaporation

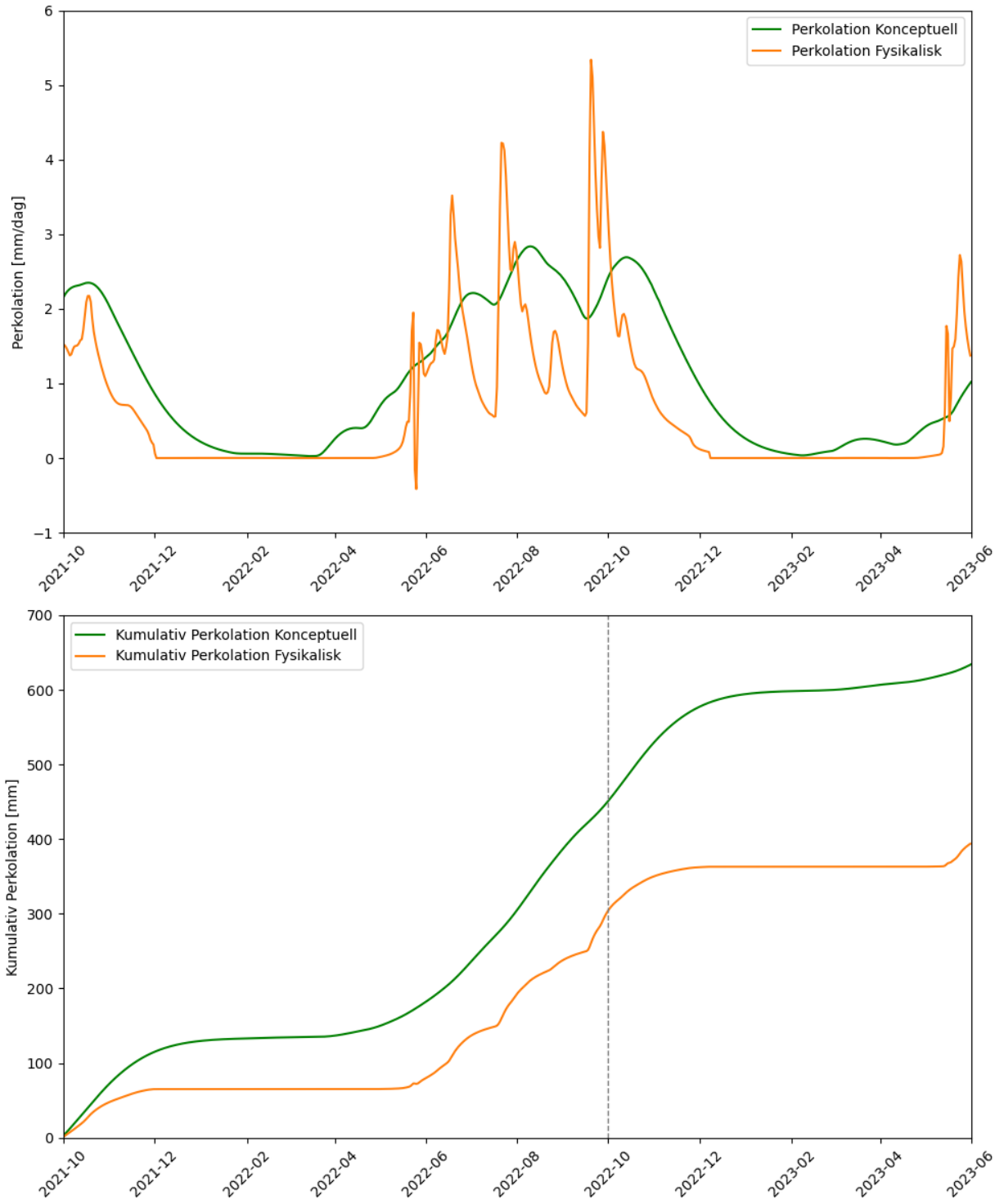
Perkolationsflödet och dess årstidsvariationer av stort intresse när lakvattenbildning undersöks. I detta avsnitt presenteras perkolationsflödet, här presenteras även faktisk evaporation (FE) som direkt motverkar storleken på perkolationsflödet.

Som utdata från den fysikaliska modellen fanns markvattenflödet från alla modellerade lager samt djup perkolation (*deep percolation*) som representerar mängden vatten som lämnar modellen och perkolerar ner i gråbergsdeponin. Eftersom det modellerades ett större djup i den fysikaliska modellen än i den konceptuella modellen valdes det att presentera vattenmängden som perkolerar från det tionde lagret, där de nedersta TDR-sensorerna återfanns, för att en mer rättvis jämförelse skulle kunna göras mellan modellerna.

Daglig summa av perkolation (positivt vertikalt nedåt) från det tiondelagret i den fysikaliska modellen och från det lägsta lagret i den konceptuella modellen visas i Figur 16. Vattenflödet är nära noll från december till början av maj, när marktemperaturen är under noll (se Figur 15) för den fysikaliska modellen. För den konceptuella modellen sker flödet under en längre men når nära noll under februari och mars månad. Under sommaren ökar flödet betydligt för båda modellerna, där flödet är som störst från juni till september för den fysikaliska modellen. Medan för den konceptuella modellen är det förskjutet en månad framåt och har sitt största flöde ungefär från juli till oktober.

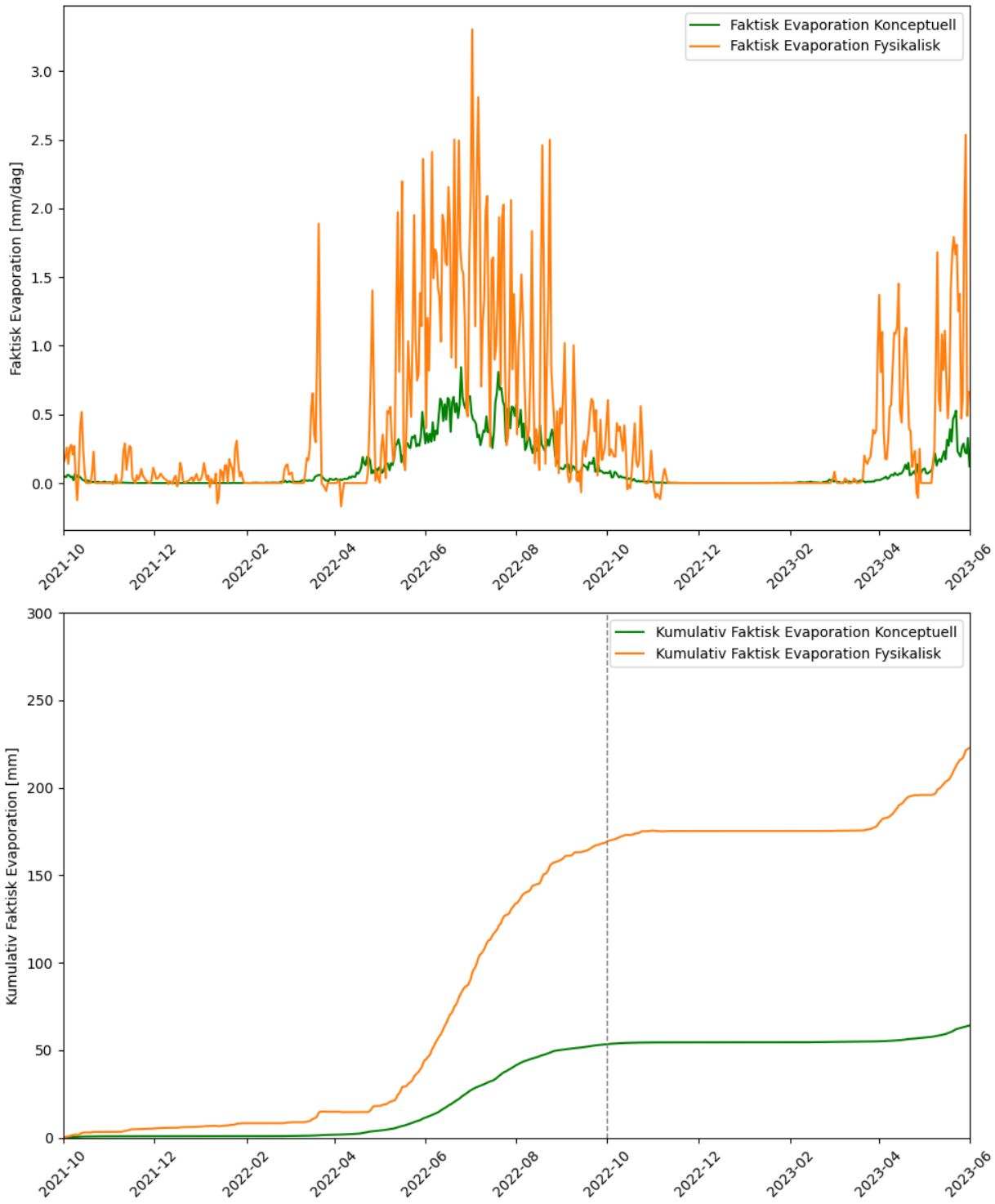
För den fysikaliska modellen har perkolationen större amplitud och varierar mer, medan den konceptuella modellen håller ett mer stadigt flöde med mindre och plattare toppar. Det finns tre stora toppar under sommaren 2022 för båda modellerna som sammanfaller med stora nederbörd och infiltrationstillfällen (se Figur 13). Det kan ses att det finns en stor förskjutning i topparna för den konceptuella modellen jämfört med den fysikaliska modellen. Det finns ett nettonegativt (<0) flöde för den fysikaliska modellen under en kort period i slutet av maj, vilket innebär ett vertikalt uppåt flöde.

I figuren visas även kumulativ perkolation. Från den kumulativa grafen kan utläsas att efter hela mätperioden hade den fysikaliska modellen ett simulerat perkolationsflöde på 394 mm och 306 mm efter ett år. Den konceptuella modellen hade simulerat ett perkolationsflöde på 633 mm efter hela mätperioden och efter ett år 450 mm.



Figur 16. Dagliga och kumulativa värden av perkolation från den konceptuella modellen och den fysikaliska modellen. Den streckade vertikala linjen markerar där det hydrologiska året slutar. Året började 2021-10-01.

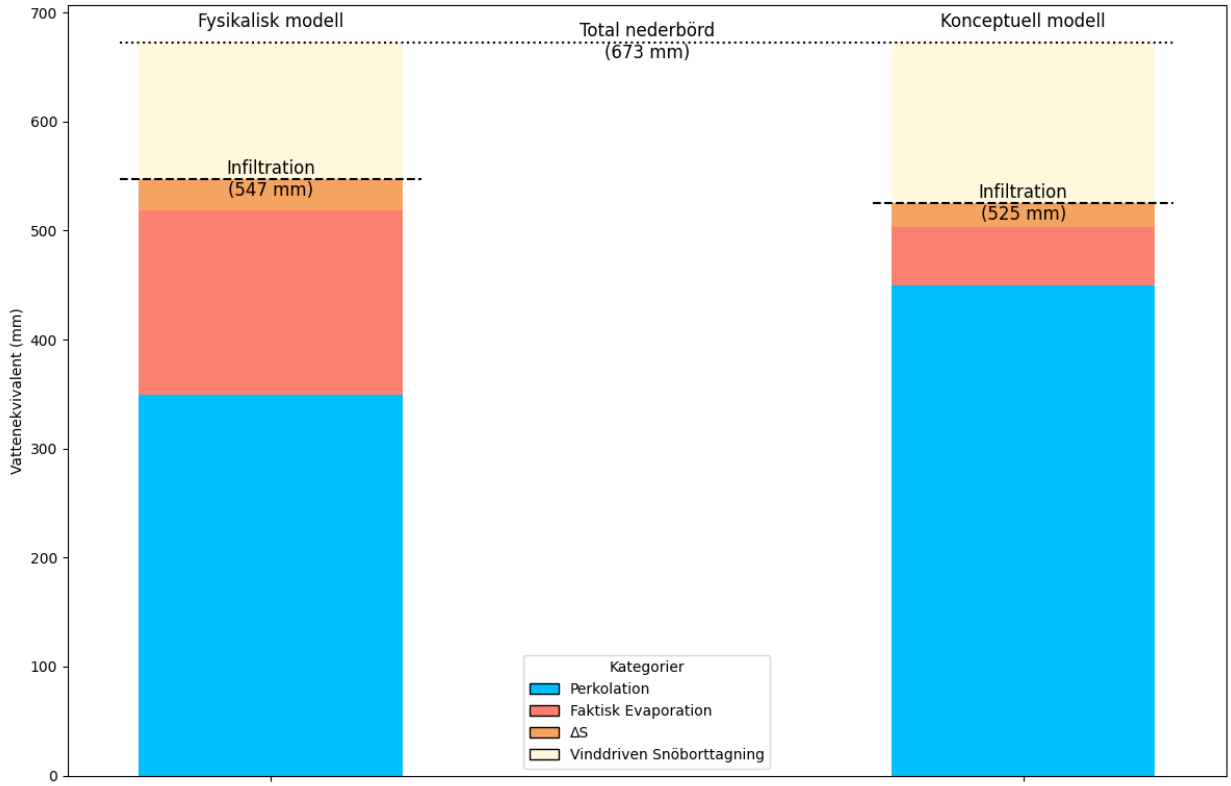
FE är ett flöde som motverkar perkolationsstorleken till stor grad. Daglig summa av FE (positivt vertikalt uppåt) visas i Figur 17 för båda modellerna. För den konceptuella modellen är FE som störst under sommarmånaderna när PE (se PE enligt Hargreaves i Figur 10) och markfuktigheten (se Figur 14) är som högst. Samma årstidsvariationer kan ses i den fysikaliska modellen, men fluktuationerna och amplituden är mycket större för den fysikaliska modellen. Det finns ett nettonegativt (<0) FE under några dagar, detta innebär att för dygnen var kondensationen större än FE. Under hela mätperioden beräknade den konceptuella modellen ett FE på 64 mm och 53 mm efter ett år. För den fysikaliska modellen beräknades ett flöde på 223 mm för hela perioden och efter ett år 170 mm, vilket visas i den kumulativa grafen.



Figur 17. Dagliga och kumulativa värden av faktisk evaporation från den konceptuella modellen och den fysikaliska modellen. Den streckade vertikala linjen markerar där det hydrologiska året slutar. Året började 2021-10-01.

För att sammanfatta vattenbalansen på Triangeldeponin presenteras en fördelning av de olika hydrologiska komponenterna, inklusive infiltration, vinddriven snöborttagning, evaporation, perkolation och förändringar i vattenlagring för de båda modellerna. Denna fördelning, som illustreras i Figur 18, ger en översikt av hur nederbörden fördelas mellan olika hydrologiska processer under det hydrologiska året 2021-10-01–2022-10-01. Det framgår tydligt i figuren att

den konceptuella modellens större perkolationsflöde beror till stor del av det betydligt mindre evaporationsflödet.



Figur 18. Sammanställning av modellerad vattenbalans på Triangeldeponin för det hydrologiska året 2021-10-01–2022-10-01.

6 Diskussion

I detta avsnitt diskuteras det resultat som presenterats, osäkerheter i studien och framtidsutsikter. Resultaten från beräkningar av potentiell evaporation och motivering för valet av beräkningsmetod för potentiell evaporation som användes som indata till den konceptuella modellen redovisas. Detta syftar till att besvara frågeställningen: ”Vilka enkla beräkningsmetoder kan användas för att estimera potentiell evaporation i deponin?”.

En jämförelse mellan de båda modellerna görs för att besvara frågeställningen: ”Jämfört med en avancerad fysikalisk modell, hur väl står sig en simplare konceptuell modell när det gäller att beskriva vattenbalansen i gråbergsdeponin?”. Parallellt med jämförelsen av modellerna lyfts resultat om årstidsvariationer i perkolation och evaporation och dess innebörd för att besvara frågeställningen: ”Hur varierar mängden vatten som perkolerar och evaporerar under årets olika perioder?”

6.1 Potentiell evaporation

Utifrån data i Tabell 9 framgår det att elva beräkningsmetoder inte överskattade den potentiella evaporationen (PE), medan Blaney-Criddle, Linacre och Romanenko troligtvis gör detta och anses därför inte trovärdiga. Haudes beräkningsmetod verkar underskatta PE i hög grad jämfört med andra beräkningsmetoderna och anses även den mindre trovärdig.

Enligt kriterierna i avsnitt 2.1.1 bör en beräkningsmetod som använder färre parametrar föredras framför en mer avancerad beräkningsmetod om liknande resultat kan uppnås. Alla sex kombinationsmetoder har åtminstone en, enklare, beräkningsmetod som ger liknande uppskattningar av PE. Därför valdes dessa sex kombinationsmetoder bort också. Oudins, Kimberly-Penmans och Thom-Olivers beräkningsmetoder är inte *benchmarked* i PyET, vilket

innebär att de inte är kontrollerade mot tidigare beräkningar av potentiell evaporation med samma beräkningsmetod (se Vremec, Collenteur och Birk, 2023) och anses därför mindre trovärdiga.

Valet av beräkningsmetod bland de kvarvarande tre metoderna är inte tydligt utifrån kriterierna. Men Hargreaves beräkningsmetod är den som uppskattar PE närmast PM-FAO-56. Publikationen FAO-56 (Allen et al. 1998) är en av de mest citerade inom sitt område (Pereira et al. 2015). Vidare använde gråbergsprojektet i Diavik, som har liknande klimatförutsättningar som Triangeldeponin, PM-FAO-56 för att uppskatta evaporationen (Neuner et al. 2013). Detta, kombinerat med att Hargreaves explicit nämns i FAO-56 som en beräkningsmetod med global validitet, ledde till valet av Hargreaves som beräkningsmetod för den potentiella evaporationen som skulle användas som indata till den konceptuella modellen.

FAO-56 rekommenderar att för bar mark att en faktor läggs på PE. De första dagarna efter regn bör en faktor på 1,15 tillämpas som sedan avtar med tiden om inget nytt regn faller (Allen et al. 1998; Torres & Calera 2010). Denna korrektion gjordes inte på grund av det inte tydligt framgick att det var applicerbart på en plats som Triangeldeponin. Vidare var det inte tydligt om beräkningen övergick i en uppskattning av faktisk evaporation om detta applicerades.

Mer avancerade beräkningsmetoder som PM-FAO-56 kan med fördel användas för att beräkna PE där data finns tillgänglig. Det finns dock flera anledningar till att en enklare beräkningsmetod var önskvärd för att beräkna PE vid Triangeldeponin. För Triangeldeponin fanns det stora dataluckor, vilket beskrivs i avsnitt 4.1.1. Nettostrålning var en av de variabler som påverkades mest av dessa dataluckor, och det var svårt att fylla igen dessa luckor eftersom nettostrålningsdata inte fanns tillgänglig från närliggande mätstation. För att fylla igen dessa luckor krävdes det användning av flera olika variabler, vilket påverkade trovärdigheten för nettostrålningsdata negativt och var tidskrävande. Att använda en beräkningsmetod som har lufttemperatur som drivande variabel, där dataluckor är enklare och mer pålitliga att fylla i, kan därför vara fördelaktigt. Detta skulle kräva mindre arbete för att fylla i eventuella dataluckor vid framtida beräkningar av PE. En enkel beräkningsmetod kan också underlätta uppskattningar av hur PE förändras med klimatförändringarna, eftersom lufttemperatur ofta finns tillgänglig för många olika klimatscenarier.

En av fördelarna med den konceptuella modellen är att den använder färre nödvändiga variabler och parametrar för att köras. Att använda en beräkningsmetod för PE som kräver många variabler skulle motverka denna fördel. Genom att hålla modellen enkel och minska antalet nödvändiga variabler kan beräkningsprocessen bli smidigare och mer tillgänglig för framtida studier och tillämpningar. Om nya mätstationer sätts upp för att öka datatätheten skulle kravet på antal sensorer även minska.

6.2 Jämförelse av den konceptuella modellen mot den fysikaliska modellen

6.2.1 Nederbörd och infiltration

När observerad SWE (se Figur 22 i bilaga B), lufttemperaturdata (se Figur 21 i bilaga B) och vinddata (se Figur 23 i bilaga B) undersöks tillsammans är det tydligt att vinden för bort delar av den snö som faller. Hur den här dynamiken tas i hänsyn skiljer modellerna åt. I den konceptuella modellen används parametern S_{FCF} (som korrigerar mängden snönederbörd som modelleras från observerad snönederbörd) för att ta hänsyn till bortförandet av snö. Den fysikaliska modellen tvingar däremot det modellerade snödjupet att följa det observerade snödjupet i varje tidssteg.

Det finns större skillnader i SWE för de båda modellerna. Detta är ganska förståeligt då den konceptuella modellen endast kalibrerades för att passa de största snötopparna och inte anpassar sig till andra delar av snösäsongen. När den konceptuella modellen tar hänsyn till parametern S_{FCF} (värde på 0,33) innebär det att endast cirka en tredjedel av observerat snöfall faller i modellen. För att snömängden ska nå upp till topparna måste det ske en större ackumulering av snö innan

topparna, vilket resulterar i stora skillnader jämfört med observerad SWE. Detta jämfört med den fysikaliska modellen där snödjup anpassats för att modellen ska uppskatta SWE så bra som möjligt.

Skillnader i hantering verkar dock inte haft allt för stor påverkan på infiltrationen vilket framgår av Figur 13. Ett syfte med kalibreringen av snörutinen var att snösmältningsinfiltrationen skulle bli så nära verkligheten som möjligt, vilket då lyckades i någon utsträckning.

Under vintern är infiltrationen likartad mellan modellerna, vilket är förväntat eftersom nederbörden då mestadels faller som snö och temperaturerna är så låga att ingen smältning sker. Detta innebär att inget flytande vatten finns tillgängligt att infiltrera marken. Det finns dock perioder i den konceptuella modellen där snö på marken modelleras trots att den i verkligheten har smält eller blåst bort tidigare. När temperaturen tillåter det, smälter delar av denna snö, vilket leder till att en större mängd infiltrerat vatten simuleras än vad den fysikaliska modellen gör för samma period. Samtidigt finns det snösmältningstillfällen med större mängd infiltrerat vatten i den fysikaliska modellen än i den konceptuella modellen.

Under sommaren, när temperaturen är betydligt över 0 °C, visar båda modellerna liknande resultat. I den konceptuella modellen antas infiltrationskapaciteten vara obegränsad, vilket innebär att allt tillgängligt flytande vatten vid ytan infiltrerar marken inom samma tidssteg. CoupModel beräknar däremot infiltrationskapaciteten för varje tidssteg, som bland annat är kopplad till om marken är frusen eller inte. Viss infiltration kan ske vid en marktemperatur mindre än 0 °C. Endast under några få korta perioder kunde inte allt vatten infiltrera marken i CoupModel (se Figur 33 i bilaga E), vilket skedde efter stor snösmältning när marken fortfarande var helt eller delvis frusen. Antagande om fri infiltration på Triangeldeponin verkar alltså rimligt, förutsatt att vatten endast finns i flytande form när markens yttemperatur är över 0 °C. Ett scenario där detta inte stämmer vore om regn faller på frusen mark.

6.2.2 Markfuktighet och marktemperatur

Figur 14 kan ge intrycket av att årstidsvariationerna i modellerad markfuktighet är liknande för båda modellerna, men figuren återger inte hela sanningen. Modellerna skiljer sig markant när is inkluderas. I CoupModel finns det en skillnad på flytande vattenhalt och total vattenhalt där is inkluderas (se Figur 34 i Bilaga E). De branta dalar som syns i markfuktighetsdata under vinterhalvåret är dalar i mängden flytande vatten (som TDR-sensorerna mäter) på grund av att vattnet fryser – vatten försvinner alltså inte från det aktuella lagret. När isen smälter under olika perioder ökar markfuktigheten snabbt, även om ingen större infiltration sker under den aktuella perioden. Detta kan studeras i markfuktighetsfigurerna (Figur 14) och marktemperaturfigurerna (Figur 15), där avvikelser mellan modellerad och observerad marktemperatur också leder till skillnader i markfuktighet.

Den konceptuella modellen är kalibrerad utifrån markfuktighetsdata och tar ingen hänsyn till marktemperaturen, vilket innebär att den saknar dynamik för frysning och upptining. I modell rinner vattnet ut tills markvattenhalten når S_{min} (minsta markfuktighet varvid perkolation upphör) i stället för att frysa och smälta när temperaturen ökar igen. När marken sedan värms upp under våren finns det inget vatten kvar som skulle smälta och öka markfuktigheten enligt vad som observerats. En markfuktighet som stämmer någorlunda väl med observerad markfuktighet uppnås någon månad efter att marken tinats upp efter ett större inflöde än utflöde har skett till lagret. Det bör nämnas att det antagligen inte är markens minsta vattenhalt som egentligen modellerats som nås under vintern, utan det flytande vatten som finns kvar i markprofilen när den är frusen. Som det modelleras i Coupmodel är den flytande vattenhalt som är kvar när marken fryser (<0 °C) en funktion av temperatur (se *FreezepointFWI* i Jansson och Karlberg, 2011).

Den fysikaliska modellen tar däremot hänsyn till frysning och upptining, vilket är en stor anledning till att den är bättre på att anpassa en liknande markvattenhalt i perioder när marken fryser eller tinar upp. Det fanns dock svårigheter vid kalibrering av modellen att få till en god anpassning i alla lager. Det framgick att en hög hydraulisk konduktivitet krävdes för att markhaltstoppar i de undre lagren inte skulle komma förskjutas mot observerade data. Samtidigt tycktes en hög hydraulisk konduktivitet innebära att för mycket vatten lämnade översta lagren och modellerad vattenhalt blev mycket lägre än observerad markvattenhalt. En förändring av vattenretentionsparametrarna skulle också kunna leda till en snabbare fortplantning av markfuktighetstoppar, men detta skulle ha liknande effekt på anpassningen som en förändring av den hydrauliska konduktiviteten. Vattenbalansens olika komponenter påverkas av markvattenhalten och dess förändring på olika sätt, varför en kompromiss gjordes för att försöka få en någorlunda anpassning i alla lager.

Den konceptuella modellen har inte alls lika snabbt svar på toppar och dalar som den fysikaliska modellen i de djupare lagren, där toppar och dalar blir mer och mer avtrubbade med ökat djup (lager). Detta beror delvis på frysning, men också på sättet flödet är modellerat ut från lagren. Den konceptuella modellen använder en empirisk ekvation som beskriver flödet och endast approximera verkliga processer. Vidare verkar det som att k (linjära lagringskoefficienten) har kalibrerats till ett mindre värde än vad den borde vara för att totalt sett få en bättre anpassning över hela perioden. Detta påverkar hur snabbt vattnet lämnar lagren. Resonemanget stöds av hur perkolationsflödet ser ut, där flödestoppar är förskjutna framåt i tiden, har mindre amplitud och varierar mindre (se Figur 16). k i sig är inte en funktion av vattenhalten som den hydrauliska konduktiviteten är (se ekvation 17), vilket kan göra det svårt att hitta ett värde som ger ett representativt flöde för olika markvattenhalter.

6.2.3 Perkolation och faktisk evaporation

Det finns tydliga likheter mellan modellerad markfuktighet och modellerad perkolation, vilket är förståeligt eftersom en förändring i vattenhalt (inklusive is) innebär att ett vattenflöde har skett. Markfuktigheten sjunker snabbare i den fysikaliska modellens tionde lager (lagret där perkolation beräknas) än i den konceptuella modellen, vilket ger ett snabbare och större svar i perkolationen. Perkolationen i den fysikaliska modellen är dock begränsad till kortare del av året än i den konceptuella modellen. Detta beror sannolikt på om modellen tog hänsyn till marktemperaturen eller inte.

När marktemperaturen (se Figur 15) är under noll, finns det inget perkolationsflöde (se Figur 16) i den fysikaliska modellen, vilket visar på tydliga interaktioner mellan vatten- och värmeflöden. Även fast den konceptuella modellen inte tar hänsyn till marktemperaturen, visar den tydliga skillnader mellan årstider. Detta beror på att infiltrationsmängden (se Figur 13) är så låg under vintern men hög under sommaren.

Det framgår inte tydligt i jämförelsen mellan modellerna att den fysikaliska modellen modellerar ett makroporflöde (preferentiellt flöde). Även om den fysikaliska modellen endast modellerat vattenflöde i matrisen, hade skillnaderna mellan modellerna förmodligen varit likartade. Det finns ingen observerade data om flöde från Triangeldeponin (vilket är vanligt i många andra studier att det finns ett observerat flöde ut från den studerade gråbergsdeponin) som perkolationsflödet skulle kunna jämföras med; endast markfuktighetsdata är tillgänglig. Markfuktigheten kan bara ge en uppfattning om möjliga vattenflöden. Med de modelleringsparametrar och inställningar som används kan inget säkert påstående göras om preferentiellt flöde är signifikant eller inte. Observerade data indikerar dock på att det finns markheterogeniteter även på liten skala, vilket tyder på att det kan finnas möjlighet för preferentiellt flöde.

Eftersom infiltrationen sett över längre perioder är jämförbar i de båda modellerna skulle det kunna förväntas att modellerna skulle ha liknande absoluta värden av perkolation om det inte vore för skillnader i faktisk evaporation.

Att årstidsvariationerna är liknande för den faktiska evaporationen är förväntat, men att de två modellerna skiljer sig mycket i amplitud och fluktuationer är kanske inte lika självklart. Den fysikaliska modellen har mycket större daglig variation, vilket kan bero på att en av de drivande variablerna, nettostrålning, varierar mycket dag för dag. Potentiell evaporation enligt Hargreaves (se Figur 10), som används i den konceptuella modellen, har inte alls samma variationer och är därmed stabilare från dag till dag, eftersom den inte beror av nettostrålningen. Det finns en möjlighet att använda, till exempel, PM-FAO-56 som också beror av nettostrålningen för att få en mer jämförbar dag-för-dag-jämförelse. Det har troligen dock inte stor påverkan på FE sett över längre perioder. Skillnader i storleken av evaporation sett över längre perioder kan även den ha att göra med om modellen tar hänsyn till marktemperatur eller inte.

Markfuktigheten i ett lager påverkas av vattenflödena in och ut, där de utgående flödena består av faktisk evaporation och perkolation. När markfuktigheten underskattas i det översta lagret, som det gör i det här fallet (se Figur 14), skulle en slutsats kunna dras att evaporationen har överskattats. Detta behöver dock inte nödvändigtvis vara den rätta tolkningen; det kan också bero på att ett för stort perkolationsflöde har modellerats, där det aktuella lagret inte hållit kvar vatten i samma utsträckning som i verkligheten. Detta kan i sin tur indikera en underskattning av evaporationen.

Om den modellerade markvattenhalten är lägre än den observerade markvattenhalten, betyder det att mängden tillgängligt vatten för evaporation är mindre. Den lägre modellerade vattenhalten är särskilt tydlig under den första delen av sommaren, när den potentiella evaporationen var som störst (se Figur 9–11). Detta är speciellt ett problem för den konceptuella modellen eftersom parametrarna som bestämmer den faktiska evaporationen kalibreras utifrån markfuktigheten. Eftersom markfuktigheten i modellen underskattas rejält under första delen av sommaren (då en betydande mängd vatten egentligen borde ha bildat is under vintern och sedan smält), tar det tid innan markfuktigheten återgår till nivåer som liknar den observerade markfuktigheten. Om då ett mindre bortflöde av vatten sker i form av evaporation, anpassas modellen snabbare och därmed bättre till observerade data. Detta leder sannolikt till att parametrar som reglerar faktisk evaporation kalibreras så att den faktiska evaporationen modelleras till en lägre nivå än den faktiskt är. För att förbättra evaporationsparametrarna kunde markrutinen ha körts endast under sommaren 2022 med initialt samma vattenhalt som observerats, och därefter kalibrerats evaporationen. Detta skulle högst troligt leda till en högre evaporation men sämre NSE för hela mätperioden.

För den fysikaliska modellen kalibreras evaporationen framför allt med marktemperaturen. Att kalibrera mot marktemperaturen gjordes eftersom CoupModel beräknar en energibalans där markytans temperatur och evaporation ingår; de är alltså direkt korrelerade (se även ekvation [1]). Detta resulterade i att evaporationen blev mycket större än om den hade kalibrerats mot markfuktigheten. En ökad evaporation ledde till en något sämre anpassning i markfuktigheten, vilket är förståeligt då markfuktigheten redan var underskattad.

Samtidigt som ovanstående argumentation tyder på en underskattning av faktisk evaporation, är metoder för att beräkna potentiell evaporation och evapotranspiration framtagna för marker med betydligt mindre partiklar och porstorlekar än den mark som studeras i denna studie. Markens kapillära stighöjd är omvänt proportionell mot porositeten (Beskow 1929). Detta innebär att när evaporation sker från markens översta skikt, kommer inte det översta lagret att återfuktas i lika stor utsträckning eftersom den kapillära stighöjden är mindre. Detta kan resultera i en mindre faktisk evaporation då inte tillräckligt mycket vatten finns tillgängligt.

Kondensation av vattenånga är ett vattenflöde som motverkar evaporation och är något som skiljer modellerna åt. Kondensationens påverkan på den faktiska evaporationen kan ses genom att studera Figur 35 i bilaga E som visar evaporationen i en körning i CoupModel utan kondensation. Om figuren jämförs med en körning med kondensation (Figur 17) kan det ses enligt vad den fysikaliska modellen beräknat, efter hela perioden en skillnad på 12 mm och efter ett år 9 mm. Men ett utlåtande om kondensationens betydelse bör inte göras här. Detta för att en empirisk ekvation för snödynamiken har använts och inte en som baseras på energibalansen. Detta innebär att kondensation och sublimation från snötäcket inte tagits med, när tester gjordes med en energibaserad styrande ekvation så var kondensationsflödet betydligt större.

6.2.4 Allmän jämförelse och tidigare studier

En av anledningarna som Atmosudirdjo (2019) angav som orsak till skillnader i modellerad och observerad data var tidsupplösningen, vilket är något som skiljer sig mellan modellerna i denna studie. Den fysikaliska modellen har en tidsupplösning på 96 iterationer per dygn (drivande variabler med timvis upplösning) medan den konceptuella modellen har en tidsupplösning på ett dygn. En högre upplösning i den fysikaliska modellen ger möjlighet till mer detaljerade och dynamiska simuleringar. Till exempel, om temperaturen överstiger tröskeltemperaturen under delar av dagen men medelvärdet för hela dagen är under tröskeltemperatur, kan snö smälta i en modell med hög upplösning, medan det inte skulle ske i en modell med lägre upplösning.

Teoretiskt sett skulle den konceptuella modellen kunna köras med högre upplösning. Snörutinen kan anpassas för timvisa beräkningar med vissa skriptjusteringar. I markrutinen är varje modellerad dag redan uppdelad i mindre tidssteg, men potentiell evaporation kan endast beräknas per dag med de beräkningsmetoder som används i denna studie. Detta gör det svårt att göra något annat för att öka tidsupplösningen än vad som redan görs i skriptet. Att kalibrera den konceptuella modellen med högre upplösning skulle kräva ökad beräkningskraft och tid, och det är osäkert om detta skulle resultera i en tydlig förbättring av modellens prestanda. De potentiella fördelarna med en högre upplösning måste vägas mot de ökade kostnaderna i beräkningskraft och tid. Med de tidsupplösningar som konceptuella modellen har tar markrutinen knappt 2 sekunder, medan snörutinen kan köras flera per sekund. Detta är inte lång tid om endast en körning ska göras men om en kalibrering görs, där skriptet körs flera hundra gånger, har tiden för varje enskild körning stor påverkan.

Att den konceptuella modellen använder färre variabler än den fysikaliska modellen kan anses önskvärt. Detta då mindre arbete i form av dataanalys kommer att krävas för att ta fram data till modellen. Vidare gör det modellen enklare att distribuera (i rummet), eftersom en ny mätstation inte behöver mäta lika många variabler. Detta innebär färre sensorer och därmed mindre kostnad och arbete.

Perkolationsflödet som tränger ner i gråbergsdeponin motsvarar 67 % (450 mm/673 mm) av nederbörden för den konceptuella modellen och 45 % (306 mm/673 mm) för den fysikaliska modellen under ett hydrologiskt år, vilket är då en uppskattning av den årliga avrinningskoefficienten (alternativ benämning nettoperkolation, nettoinfiltration eller nettonederbörd). Detta kan jämföras med 37 % av nederbörd som Neuner et al. (2013) fann i deras studie vid diamantgruvan i Diavik. Årsmedelnederbörden för Diavik var 280 mm och årsmedeltemperaturen var -5,2 °C. Potentiell evaporation var cirka 270 mm/år, där evaporation sker från juli till november. Det är värt att notera att Diavik har permafrost och signifikant mindre nederbörd, vilket skiljer sig från förhållandena vid Triangeldeponin. En annan studie av Villeneuve et al. (2017) i Elk Valley, British Columbia, fann att 48–72 % av nederbörden (beräknad nettoperkolation 446 ± 50 och observerad nederbörd på 689–823 mm) hade runnit genom den studerade gråbergsdeponin. Detta under förutsättningen att allt perkolerande vatten rinner ut ur

bergsdräneringen. Beräknad årsmedeltemperatur inom intervallet -0,4–2,7 °C på gråbergsdeponin. Klimatförhållanden är jämförbara med klimatet i Kiruna.

6.3 Osäkerheter

Som i alla studier finns det osäkerheter. I detta avsnitt diskuteras utvalda osäkerheter och deras innebörd för resultatet. Osäkerheter i de specifika modellerna anges under respektive avsnitt. Allmänna osäkerheter som inte faller under dessa kategorier får ett eget avsnitt, där framför allt osäkerheter i mätdata tas upp.

6.3.1 Konceptuell modell

En stor anledning till att snörutinen hade svårt att kalibreras över hela perioden var att modellen inte tog hänsyn till vinddriven snöborttagning. Att använda parametern S_{FCF} kan inte ses som optimalt då parametern egentligen används till att korrigera snönederbörd då nederbördsmätare ofta underskattar nederbörd som faller som snö. Snö som finns ”på marken” i modellen blir kvar tills det smälter, vilket kan resultera i snösmältningsperioder som inte återfinns i verkligheten. Snörutinsmodellen är bättre anpassad att fungera på en plats där det blåser mindre, men går att användas ändå men då behöver användaren känna till begränsningarna.

Allen et.al (1998) rekommenderar i FAO-56 att ha ett lager på 0,1 – 0,15 m som evaporation sker ifrån. Det översta lagret i modellen kan antingen vara 1,2 m eller 0,346 m djup Detta innebär att evaporationen kan ske från ett mycket större djup i modellen än vad som är realistiskt. Vidare antogs det i denna studie att inget vattenflöde sker vertikalt uppåt mot ytan, vilket stämmer väl överens med vad Tokunaga, Wan och Olson (2002) fann i deras undersökning. Det kan därför argumenteras för att evaporationen sker från ett ännu mindre djup i verkligheten, vilket potentiellt skulle leda till mindre faktisk evaporation. Dock är det inte klart i vilken utsträckning detta påverkar, men det kan ses som ett motargument till tidigare påståenden om att den konceptuella modellen underskattar evaporationen.

6.3.2 Fysikalisk modell

Att den fysikaliska modellen saknade möjlighet att påverka snödynamiken direkt genom vindpåverkan framgick först under senare delar av studien. I stället valdes det att använda en funktion i programmet som tvingar det modellerade snödjupet att anpassas till det observerade snödjupet. Denna metod kräver dock korrekta mätdata för snödjup under hela mätperioden. Under studien framkom det att snödjupssensorn (ultraljudssensorn) vid flera tillfällen hade mätt orimliga snödjup, med ett kontinuerligt fel i mätningen som varierade mellan 20–30 mm under sommaren när ingen snö fanns på Triangeldeponin. Detta skiftande fel gjorde det svårt att bedöma snödjupet korrekt och ökade osäkerheterna kring mätningarna. Att tvinga snödjupet till ett visst värde kan också leda till att mer smältning simuleras än vad som verkligen sker, eftersom snö "skapas" som inte kom från nederbörden, vilket i sin tur kan leda till överdriven infiltration i marken.

I den fysikaliska modellen observerades vid några tillfällen ett nettonegativt vertikalt vattenflöde, det vill säga en vattenrörelse uppåt. Ett signifikant flöde vertikalt uppåt modellerats under två dygn sett till hela mätperioden. Detta berodde på att programmet egentligen bara hanterar partikelstorlekar upp till sand, där de kapillära krafterna är starkare än för grus och block och kan tillåta en viss grad av vattenflöde uppåt.

De flesta parametervärden som har ändrats från sina standardvärden i CoupModel har justerats genom kalibrering och har inte alltid en direkt koppling till litteraturen. Brooks & Coreys parametrar (se Tabell 8) ska tas med försiktighet. De är framtagna i en kalibrering för att anpassa modellerad markfuktighet till observerad markfuktighet. h_a (luftinträdestrycket) frångår speciellt vad andra studier funnit. Keller et al. (2015) fann ett värde på 8 cm (omvandling från van Genuchtens α), Neuner et al. (2013) fann ett värde på 2 cm för matrisen och 4 cm för bulken, Blackmore et al. (2014) 17–56 cm (samtliga omräknade från van Genuchtens α till h_a). h_a ligger

alltså inom det spann som Blackmore et al. (2014) funnit men det jordprov som hade dessa värden är inte applicerbart på Triangeldeponin. Det testades att ha en lägre h_a men det resulterade i en sämre anpassning. Detta kan ha att göra med den antagande som gjort för att förenkla modellering att residualvattenhalten är noll. Samtliga studier har en nollskild residualvattenhalt som de fann vara 1–4,8 % (se Neuner et al. 2013; Blackmore et al. 2014; Keller et al. 2015). Den hydrauliska konduktiviteten och λ som kalibrerats fram återfinns dock inom det spann som dessa studier rapporterat.

När det gäller densiteten för snö ändrades denna variabel under körningen, där snö som legat på marken under en längre tid antogs ha en ökad densitet utöver det som modelleras av programmet självt. Detta stöds av Meløysund et al. (2007) fann i deras studie, där det konstateras att snö under vindpinade förhållanden får en högre densitet.

I CoupModel finns det möjlighet att ange två hydrauliska konduktiviteter: en matriskonduktivitet (k_{mat}) och en totalkonduktivitet (k_{sat}). k_{mat} är den hydrauliska konduktiviteten i matrisen, medan k_{sat} anger hydraulisk konduktivitet när markprofilen har en markfuktighet över matrisporositeten, och tar då hänsyn till flöde i makroporer. k_{mat} styr flödeshastigheten medan vattenhalten är mindre än differensen mellan totalporositeten och makroporositeten ($\theta < \theta_s - \theta_m$). I CoupModel är den största θ_m som kan modelleras på 10 % och θ_s ansattes till 30 %. Detta innebär att när markfuktigheten är under 20 % styrs flödet av k_{mat}. Endast under några få tillfällen var (medelvärdet av) markfuktigheten över 20 % vilket gjorde det svårt att kalibrera θ_m, ännu svårare att kalibrera k_{sat}, då det hade så liten påverkan.

Porositeten som till slut valdes i den fysikaliska modellen var 30 % vilket var långt över den Jørgensen (2020) fann i sin försök på jorden som var cirka 19 % för matrisen och 21 % för bulken. Det fanns flera anledningar till att en större porositet valdes. Dels var det för att en av TDR-sensorerna mätte en högsta vattenhalt på 26 % (se Figur 30 i bilaga B), dels för att det blev en förbättrad anpassning med högre porositet.

6.3.3 Allmänna osäkerheter

Det är viktigt att vara medveten om osäkerheterna i mätdata när man analyserar resultaten. En av de största osäkerheterna är relaterad till dataluckor och hur dessa fylls i. Att använda data från andra platser med olika förhållanden och anpassa dem till den specifika platsen kan påverka datakvaliteten och därmed trovärdigheten i resultaten.

När det gäller nederbördsdata är det viktigt att komma ihåg att nederbörd kan vara högst regional. Skillnader kan förekomma även inom samma avrinningsområde. SMHI:s väderstationer som mäter nederbörd med timsupplösning ligger flera mil från Triangeldeponin och observerade värden från dessa stationer kan skilja sig mycket från verkliga värden på Triangeldeponin. Nederbördsdata från SMHI:s väderstation i Kiruna, som mäter nederbörd per dygn, skiljer sig en hel del från observerad nederbörd på Triangeldeponin, vilket vidare styrker att nederbörd kan skilja sig även på kortare avstånd. Samtidigt finns det tydliga indikationer på att nederbördsstationen på Triangeldeponin underskattar nederbörden, då antagligen på grund av vind, vilket gör det svårt att säga att den faktiska skillnaden är stor (Grabs, pres. medd.).

Observerad SWE, nederbörd och snödjupet stämmer bitvis ganska dåligt med varandra, vilket försvårar modellering och kalibrering. Det är svårt att veta vilken av variablerna som är mest betrodd. På grund av tydliga problem i snödjupsmätaren gjordes förändringar av data för att få bättre korrelation mellan variablerna. Att justera data är generellt inte önskvärt, men det var nödvändigt för att snödjupet skulle kunna användas som en tvingande variabel i den fysikaliska modellen.

Att det finns en del problem i data har inte allt för stor påverkan på själva jämförelsen av de två modeller som är huvudpunkten i denna studie. Men självklart skulle förbättrad datakvalitet naturligtvis ha underlättat anpassningen av modellerna till de observerade datavärdena.

Ingen känslighetsanalys har gjorts i denna studie på de parametrar som tagits fram. En känslighetsanalys hade kunnat belysa hur snarlika anpassningsgrad (*goodness of fit*) kan erhållas av helt olika parameteruppsättningar. I den konceptuella modellen tröskeltemperaturen och graddagsfaktorn i hög grad beroende av varandra vilket då skulle innebära att olika kombinationer av dessa skulle likande anpassningsgrad erhållas. Samma gäller för h_a och λ (se Tabell 8).

Det finns tydliga skillnader mellan vad TDR-sensorerna mätt på samma djup vilket framgår av Figur 30–32 i bilaga B. Det kan ha funnits heterogeniteter i marken vilket skulle kunna förklara de spatiala skillnaderna, där markfuktighetssensorerna mätt en korrekt vattenhalt för den specifika plats som sensorn fanns på. Det är inte klart om vilken av sensorerna eller om medelvärdet återspeglar representerar verkligheten bäst. Att den fysikaliska modellen då underskattat markfuktigheten är egentligen inte helt tydlig utan kan ha återspeglat TDR-sensorerna (4, 5, 6, se Figur 20 i bilaga A) som kontinuerligt observerat en lägre markfuktighet under sommarhalvåret.

6.4 Framtidsutsikter och utvecklingsmöjligheter

En tydlig utvecklingsmöjlighet, speciellt för den konceptuella modellen, vore att implementera funktionalitet som kan ta hänsyn till vindpåverkan på snötäcket. Vidare skulle markrutinen kunna förbättras genom att inkludera observerad marktemperatur för att kunna ta hänsyn till frysnings- och upptiningsdynamik. Detta kräver dock att marktemperaturen mäts på nya platser om modellen ska kunna användas där.

Samtliga studier där vattenretentionsparametrar har angivits i samband med modellering av vattenflöde har gjort det med van Genuchten och inte med Brooks-Corey (se Neuner et al. 2013; Blackmore et al. 2014; Keller et al. 2015). För att enklare göra jämförelser i framtiden föreslås att använda van Genuchtens vattenretentionsparametrar, vilket också kan anges i CoupModel som alternativ till Brooks-Corey. Det rekommenderas också att ta fram en vattenretentionskurva för Triangeldeponin i laboratorium. Framtagandet görs då med fördel med flera prov och används sedan som utgångspunkt vid kalibrering av modellen.

Direkta mätningar av perkolation och avdunstning med hjälp av en markvägningslysimeter (se Meissner et al. 2020) skulle underlätta modelleringen och minska de osäkerheter som finns vid kalibrering av modellerna. Faktisk evaporation skulle också kunna uppskattas med hjälp av energibalansen.

Data skulle behöva samlas in under längre tid så att flera år skulle kunna jämföras. Detta skulle också möjliggöra att modellerna kan köras med en uppvärmningsperiod innan kalibreringsperioden. En uppvärmningsperiod skulle kunna användas för att få en bättre uppskattning av markfuktigheten i de lager där ingen markfuktighetssensor finns. Ett alternativ till detta vore att skapa en uppvärmningsperiod genom att duplicera observerade drivande variabler och använda den första perioden som uppvärmningsperiod, vilket inte gjordes i denna studie.

Nästa steg i studien vore att försöka distribuera modellerna så att en bättre uppskattning av perkolationen för hela Triangeldeponin erhålls. Därefter kan den framtagna perkolationen användas som indata till en modell som beskriver lakvattenbildning i gråbergsdeponin.

7 Slutsats

Syftet med denna studie var att kvantifiera vattenbalansen vid ytan på gråbergsdeponin Triangeldeponin genom hydrologisk modellering, med fokus på mängden vatten som infiltrerar och perkolerar ner i deponin samt under vilka delar av året detta sker. Studien undersökte även vilka enkla beräkningsmetoder som kan användas för att estimera potentiell evaporation och jämförde hur väl en konceptuell modell stod sig mot en avancerad fysikalisk modell när det gäller att återspegla observerade variabler, beräkna faktisk evaporation och perkolation.

För att besvara frågeställningen om enkel beräkningsmetod för potentiell evaporation jämfördes 14 olika beräkningsmetoder. Hargreaves metod visade sig vara den mest lämpliga på grund av dess enkelhet och minimala krav på data, då den endast kräver att maximal och minimal daglig temperatur mäts.

Resultatet visade att mängden vatten som perkolerar och evaporerar varierar markant under året. Under vintermånaderna är evaporationen minimal på grund av låg nettostrålning och snötäcke, samtidigt som den frusna marken leder till ett perkolationsflöde nära noll. Under maj månad ökade evaporationen på grund av ökad energitillförsel, och marken tinar vilket möjliggör vattentransport nedåt i markprofilen. Sommaren kännetecknas av relativt hög evaporation och hög perkolation på grund av stora mängder regn som infiltrerar marken. Hösten markerar en återgång till lägre evaporation och perkolation när nederbörden och nettostrålning avtar drastiskt. Under senhösten återfryser marken och perkolationen avtar helt igen.

Jämförelsen av den fysikaliska modellen och konceptuella modellen visade att trots att den konceptuella modellen inte tar hänsyn till alla detaljerade hydrologiska processer som den fysikaliska modellen gör, kan den fortfarande ge en användbar uppskattning av vattenbalansen över tid. Den konceptuella modellen har fördelen att använda färre variabler, vilket minskar behovet av omfattande mätningar och gör den lättare att implementera. Resultaten visade dock att den fysikaliska modellen, med sina detaljerade representationer av hydrologiska processer, ger en mer exakt beskrivning av vattenbalansen.

Perkolationsflödet uppgick till 67 % av observerad nederbörd för den konceptuella modellen och 45 % av observerad nederbörd för den fysikaliska modellen, med huvudsakliga skillnader som uppstår från evaporationsförluster och förmågan att simulera frysnings- och upptiningsdynamik. Perkolationsflödet ger i sig en fingervisning om hur mycket vatten som rinner ner i berget och bildar lakvatten som sedan rinner ut i närliggande recipienter. Perkolationsflödet skulle kunna användas som indata till andra modeller som beskriver vattenflödet och ämnestransporten i gråbergsdeponin, vilket ger en god uppfattning om hur mycket vatten som lakas ut från gråbergsdeponin och under vilken del av året detta sker.

8 Källförteckning

Allen, R.G., Pereira, L.S., Raes, D. & Smith, M. (1998). Crop evapotranspiration - Guidelines for computing crop water requirements - FAO Irrigation and drainage paper 56. https://www.fao.org/3/x0490e/x0490e00.htm#Contents [2024-04-19]

Amos, R.T., Blowes, D.W., Bailey, B.L., Sego, D.C., Smith, L. & Ritchie, A.I.M. (2015). Waste-rock hydrogeology and geochemistry. *Applied Geochemistry*, 57, 140–156. https://doi.org/10.1016/j.apgeochem.2014.06.020

Ansorge, L. & Beran, A. (2019). Performance of simple temperature-based evaporation methods compared with a time series of pan evaporation measures from a standard 20 m2 tank. *Journal of Water and Land Development; 2019; No 41; 1-11,*. https://journals.pan.pl/dlibra/publication/129295/edition/112841 [2024-04-27]

Atmosudirdjo, A. (2019). *Simulation of Leachate Generation from a Waste Rock Dump in Kiruna Using HYDRUS-1D*. https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-379225 [2024-02-01]

Barnett, A., Hatton, D.B. & Jones, D.W. (1998). *RECENT CHANGES IN THERMOMETER SCREEN DESIGN AND THIER IMPACT*. (INSTRUMENTS AND OBSERVING METHODS, 66). WORLD METEOROLOGICAL ORGANIZATION.

Bergström, S. (1992). THE HBV MODEL - its structure and applications. *SMHI Reports Hydrology*,

Beskow, G. (1929). EN NY METOD FÖR BESTÄMNING AV KAPILLÄR- KRAFTEN (ELLER KAPILLÄRA STIGHÖJDEN).

Blackmore, S., Smith, L., Ulrich Mayer, K. & Beckie, R.D. (2014). Comparison of unsaturated flow and solute transport through waste rock at two experimental scales using temporal moments and numerical modeling. *Journal of Contaminant Hydrology*, 171, 49–65. https://doi.org/10.1016/j.jconhyd.2014.10.009

Brooks, R.H. & Corey, A.T. (1964). *HYDRAULIC PROPERTIES OF POROUS MEDIA*.

Byman, K. (2024). *Utmaningar för att möta ökade behov av metaller och mineral*. Kungl. Ingenjörsvetenskapsakademien (IVA).

Demers, I., Molson, J., Bussière, B. & Laflamme, D. (2013). Numerical modeling of contaminated neutral drainage from a waste-rock field test cell. *Applied Geochemistry*, 33, 346–356. https://doi.org/10.1016/j.apgeochem.2013.02.025

Domenico, P.A. & Schwartz, F.W. (1998). *Physical and Chemical Hydrogeology*. 2. uppl. John Wiley & Sons, Inc.

Fernandez, A. (2023). *Gruvnäringen välkomnar ny EU-lag - men stora hinder kvarstår*. *Svemin*. https://www.svemin.se/aktuellt/nyhet/gruvnaringen-valkomnar-ny-eu-lag-men-stora-hinder-kvarstar/ [2024-02-08]

Grabs, T. (2024). *LKAB-Triangelstation-Waterbalance*. [Opublicerat material]

Hargreaves, G. & Samani, Z. (1985). Reference Crop Evapotranspiration From Temperature. *Applied Engineering in Agriculture*, 1. https://doi.org/10.13031/2013.26773

Hellman, M., Juhanson, J., Wallnäs, F., Herbert, R.B. & Hallin, S. (2024). Microbial succession and denitrifying woodchip bioreactor performance at low water temperatures. *Journal of Environmental Management*, 356, 120607. https://doi.org/10.1016/j.jenvman.2024.120607

Hendry, M.J., Wassenaar, L.I., Barbour, S.L., Schabert, M.S., Birkham, T.K., Fedec, T. & Schmeling, E.E. (2018). Assessing the fate of explosives derived nitrate in mine waste rock dumps using the stable isotopes of oxygen and nitrogen. *Science of The Total Environment*, 640–641, 127–137. https://doi.org/10.1016/j.scitotenv.2018.05.275

Hiemstra, C.A., Liston, G.E. & Reiners, W.A. (2002). Snow Redistribution by Wind and Interactions with Vegetation at Upper Treeline in the Medicine Bow Mountains, Wyoming, U.S.A. *Arctic, Antarctic, and Alpine Research*, 34 (3), 262–273. https://doi.org/10.1080/15230430.2002.12003493

Jansson, P.-E. & Karlberg, L. (2011). *REFERENCE MANUAL*.

Jørgensen, R. (2022). Soil Moisture Sensing in Mining Waste Rock: Comparing Calibration Curves of Multiple Low-Cost Capacitance Sensors and a Single TDR Sensor. *Examensarbete vid Institutionen för geovetenskaper*, (571)

Keller, J., Milczarek, M. & Zhan, G. (2015). Water balance modeling of preferential flow in waste rock materials. *Proceedings of 10th International Conference on Acid Rock Drainage & IMWA Annual Conference*, april 21 2015.

Linklater, C.M., Sinclair, D.J. & Brown, P.L. (2005). Coupled chemistry and transport modelling of sulphidic waste rock dumps at the Aitik mine site, Sweden. *Applied Geochemistry*, 20 (2), 275–293. https://doi.org/10.1016/j.apgeochem.2004.08.003

Ma, L., Huang, C., Liu, Z.-S., Morin, K.A., Dy, E., Tufa, K., Fisher, E., Zhou, J., Aziz, M. & Meints, C. (2020). A Full-Scale Case Study on the Leaching Process of Acid Rock Drainage in Waste Rock Piles and the Net Infiltration Through Cover Systems. *Water, Air, & Soil Pollution*, 231 (6), 305. https://doi.org/10.1007/s11270-020-04660-5

Meissner, R., Rupp, H. & Haselow, L. (2020). Chapter 7 - Use of lysimeters for monitoring soil water balance parameters and nutrient leaching. I: Prasad, M.N.V. & Pietrzykowski, M. (red.) *Climate Change and Soil Interactions*. Elsevier. 171–205. https://doi.org/10.1016/B978-0-12-818032-7.00007-2

Meløysund, V., Leira, B., Høiseth, K.V. & Lisø, K.R. (2007). Predicting snow density using meteorological data. *Meteorological Applications*, 14 (4), 413–423. https://doi.org/10.1002/met.40

Messing, I. (2013). *MARKFYSIKENS GRUNDER*. Instutionen för mark och miljö, SLU.

Nationalencyklopedin (u.å.a). *energibalans*. https://www-ne-se.ezproxy.its.uu.se/uppslagsverk/encyklopedi/l%C3%A5ng/energibalans-(klimatologi) [2024-03-26]

Nationalencyklopedin (u.å.b). *Stefan–Boltzmanns lag - Uppslagsverk - NE.se*. https://www-ne-se.ezproxy.its.uu.se/uppslagsverk/encyklopedi/l%C3%A5ng/stefan-boltzmanns-lag [2024-03-31]

Nationalencyklopedin (u.å.c). *växthuseffekten - Uppslagsverk - NE.se*. https://www-ne-se.ezproxy.its.uu.se/uppslagsverk/encyklopedi/l%C3%A5ng/v%C3%A4xthuseffekten [2024-03-31]

Neuner, M., Smith, L., Blowes, D.W., Sego, D.C., Smith, L.J.D., Fretz, N. & Gupton, M. (2013). The Diavik waste rock project: Water flow through mine waste rock in a permafrost terrain. *Applied Geochemistry*, 36, 222–233. https://doi.org/10.1016/j.apgeochem.2012.03.011

Nimmo, J.R. (2021). The processes of preferential flow in the unsaturated zone. *Soil Science Society of America Journal*, 85 (1), 1–27. https://doi.org/10.1002/saj2.20143

Oudin, L., Michel, C. & Anctil, F. (2005). Which potential evapotranspiration input for a lumped rainfall-runoff model?: Part 1—Can rainfall-runoff models effectively handle detailed potential evapotranspiration inputs? *Journal of Hydrology*, 303 (1), 275–289. https://doi.org/10.1016/j.jhydrol.2004.08.025

Penman, H.L. (1948). Natural Evaporation from Open Water, Bare Soil and Grass. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 193 (1032), 120–145

Pereira, L.S., Allen, R.G., Smith, M. & Raes, D. (2015). Crop evapotranspiration estimation with FAO56: Past and future. *Agricultural Water Management*, 147, 4–20. https://doi.org/10.1016/j.agwat.2014.07.031

Priestley, C.H.B. & Taylor, R.J. (1972). On the Assessment of Surface Heat Flux and Evaporation Using Large-Scale Parameters. *Monthly Weather Review*, 100 (2), 81–92. https://doi.org/10.1175/1520-0493(1972)100<0081:OTAOSH>2.3.CO;2

Ramasamy, M. & Power, C. (2019). Evolution of Acid Mine Drainage from a Coal Waste Rock Pile Reclaimed with a Simple Soil Cover. *Hydrology*, 6 (4), 83. https://doi.org/10.3390/hydrology6040083

Richards, L.A. (1931). CAPILLARY CONDUCTION OF LIQUIDS THROUGH POROUS MEDIUMS. *Physics*, 1 (5), 318–333. https://doi.org/10.1063/1.1745010

Rodhe, A. (2000). THE HYDROLOGICAL CYCLE AND THE ENERGY BALANCE. I: *Sustainable Water Management in the Baltic Sea Basin : 1. The Waterscape*. (Sustainable Water Management in the Baltic Sea Basin; 1). 55–68.

Seibert, J. (2005). *HBV light User's Manual*. Department of Physical Geography and Quaternary Geology.

Seigneur, N., Vriens, B., Beckie, R.D. & Mayer, K.U. (2021). Reactive transport modelling to investigate multi-scale waste rock weathering processes. *Journal of Contaminant Hydrology*, 236, 103752. https://doi.org/10.1016/j.jconhyd.2020.103752

SGU (2020). *Gruvor och miljöpåverkan*. https://www.sgu.se/mineralnaring/gruvor-och-miljopaverkan/ [2024-06-25]

St-Arnault, M., Vriens, B., Klein, B., Mayer, K.U. & Beckie, R.D. (2019). Mineralogical controls on drainage quality during the weathering of waste rock. *Applied Geochemistry*, 108, 104376. https://doi.org/10.1016/j.apgeochem.2019.104376

Szymkiewicz, A. (2013). Modeling Water Flow in Unsaturated Porous Media. I: Szymkiewicz, A. (red.) *Modelling Water Flow in Unsaturated Porous Media: Accounting for Nonlinear Permeability and Material Heterogeneity*. Springer. 1–8. https://doi.org/10.1007/978-3-642-23559-7_1

Tokunaga, T.K., Wan, J. & Olson, K.R. (2002). Saturation-matric potential relations in gravel. *Water Resources Research*, 38 (10), 32-1-32–7. https://doi.org/10.1029/2001WR001242

Torres, E.A. & Calera, A. (2010). Bare soil evaporation under high evaporation demand: a proposed modification to the FAO-56 model. *Hydrological Sciences Journal*, 55 (3), 303–315. https://doi.org/10.1080/02626661003683249

Villeneuve, S.A., Barbour, S.L., Hendry, M.J. & Carey, S.K. (2017). Estimates of water and solute release from a coal waste rock dump in the Elk Valley, British Columbia, Canada. *Science of The Total Environment*, 601–602, 543–555. https://doi.org/10.1016/j.scitotenv.2017.05.040

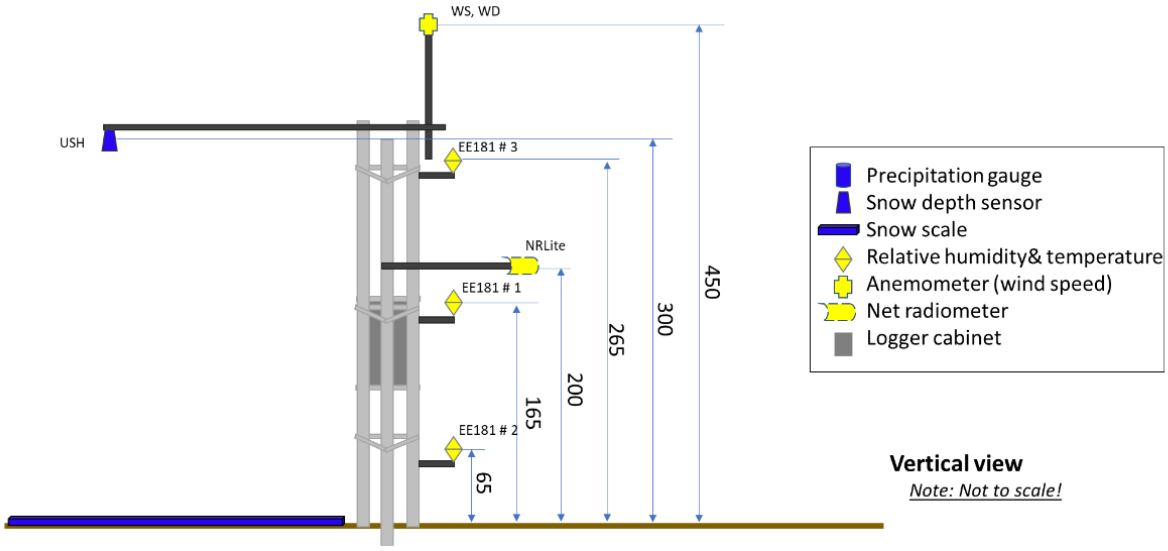
Vremec, M., Collenteur, R.A. & Birk, S. (2023). *Technical note: Improved handling of potential evapotranspiration in hydrological studies with PyEt*. Catchment hydrology/Modelling approaches. https://doi.org/10.5194/hess-2022-417

Xiang, K., Li, Y., Horton, R. & Feng, H. (2020). Similarity and difference of potential evapotranspiration and reference crop evapotranspiration – a review. *Agricultural Water Management*, 232, 106043. https://doi.org/10.1016/j.agwat.2020.106043

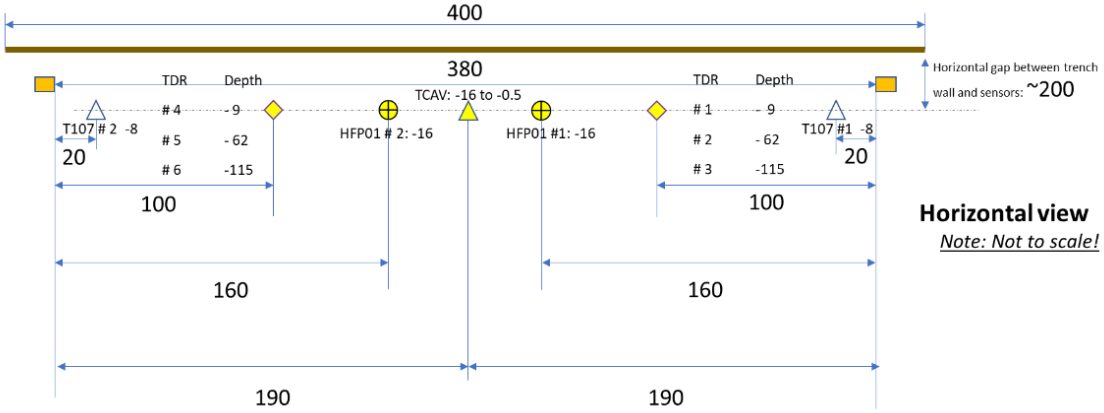
9 Bilagor

Bilaga A. Meteorologisk station

Den meteorologiska stationens utseende presenteras nedan.



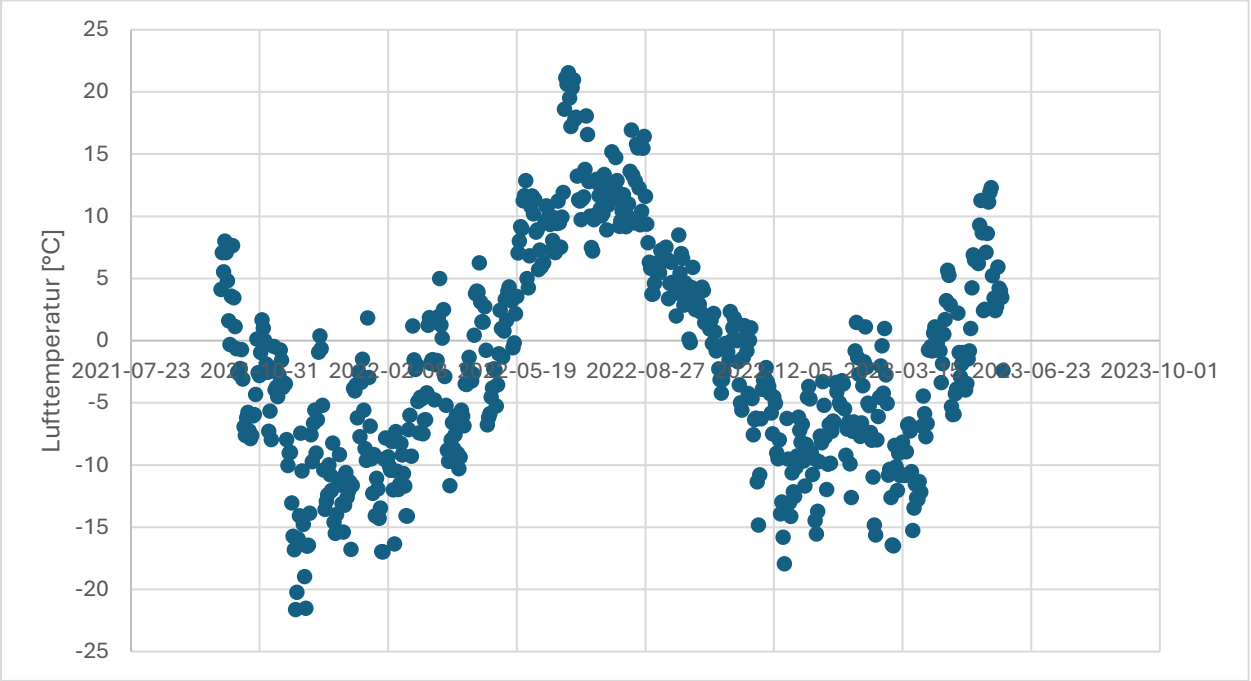
Figur 19. Meteorologisk mätstation ovan markytan på Triangeldeponin. Figur av Grabs (2023).



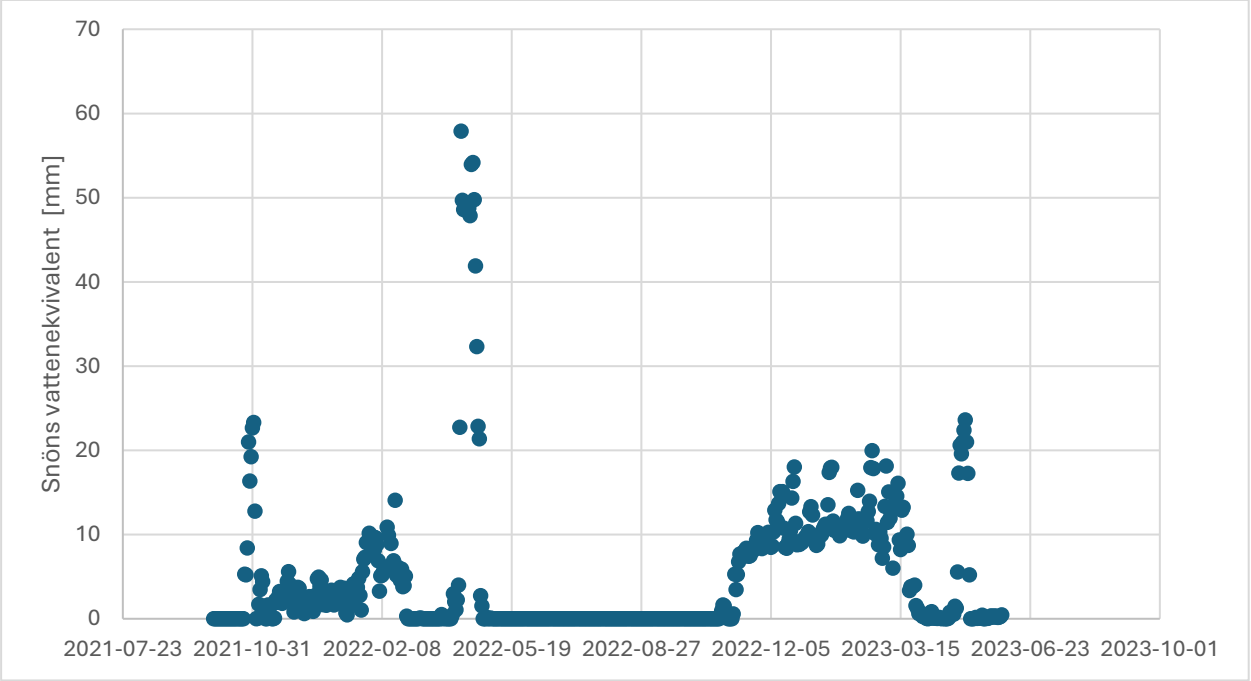
Figur 20. Meteorologisk mätstation under markytan på Triangeldeponin. Figur av Grabs (2023).

Bilaga B. Observerad data

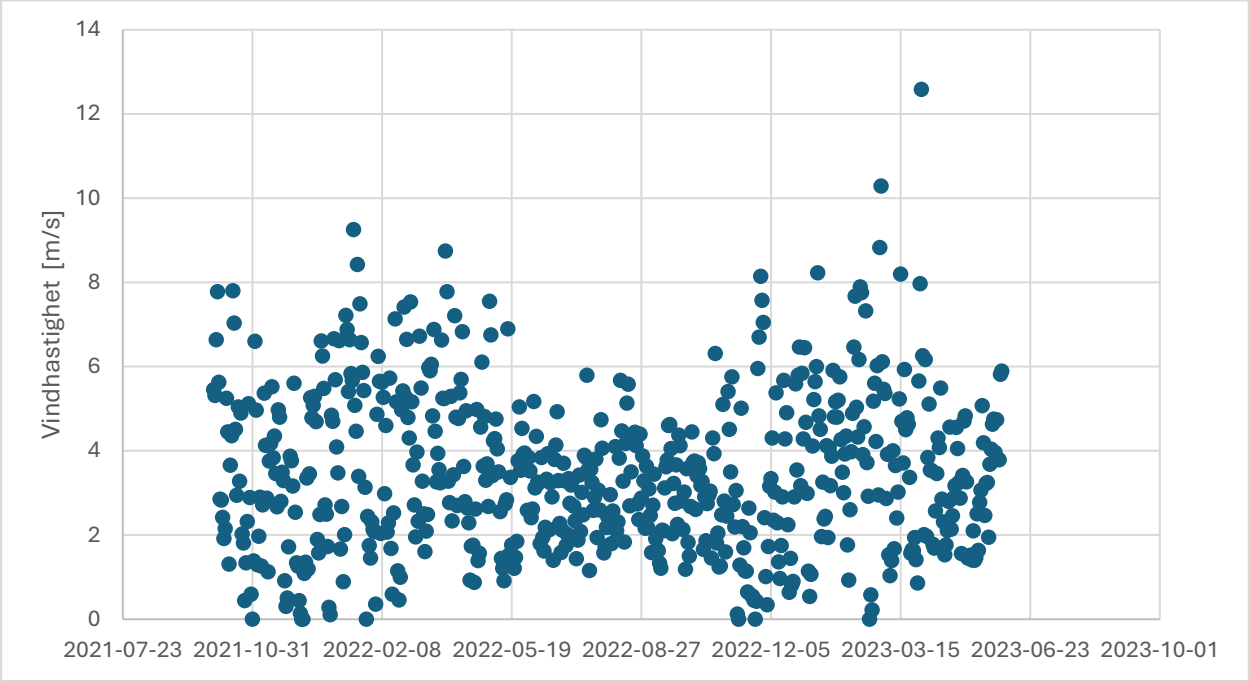
Utvalda observerade variabler presenteras i denna bilaga. Här finns även en figur som visar under vilka datum som meteorologiska stationen saknar mätdata ifrån.



Figur 21. Dagligt medelvärde Lufttemperatur (utan dataluckor) observerad på Triangeldeponin.



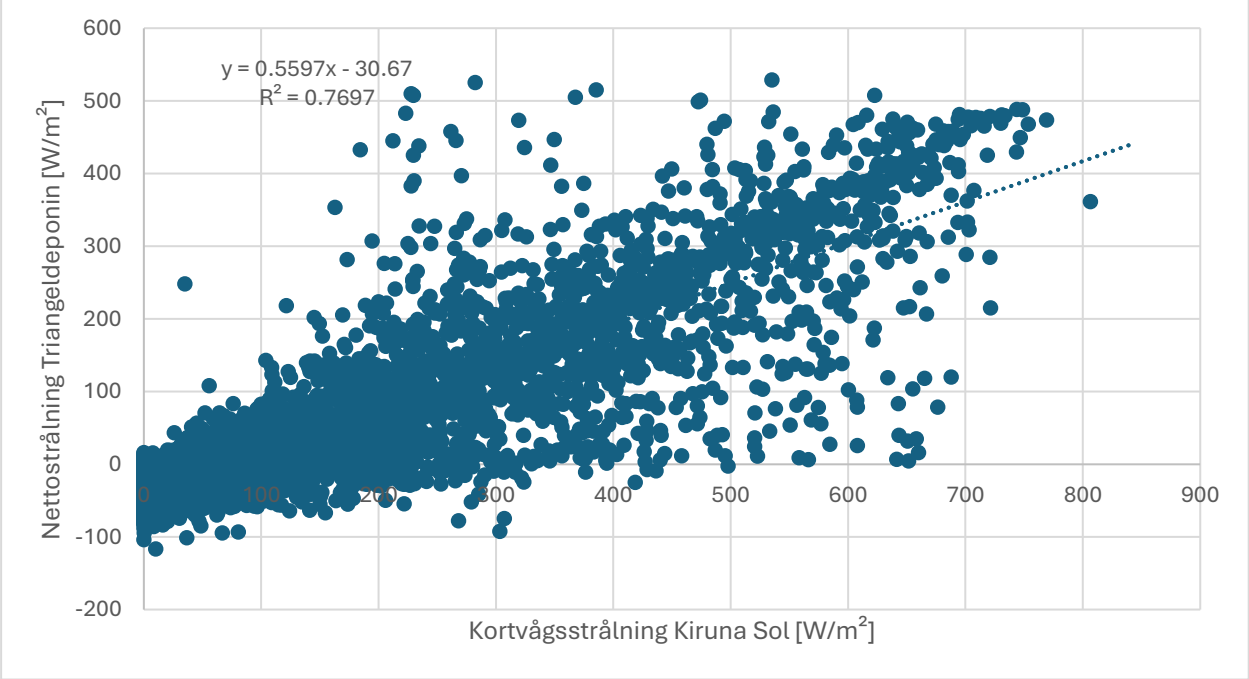
Figur 22. Dagligt medelvärde av snöns vattenekvivalent (utan dataluckor) observerad på Triangeldeponin.



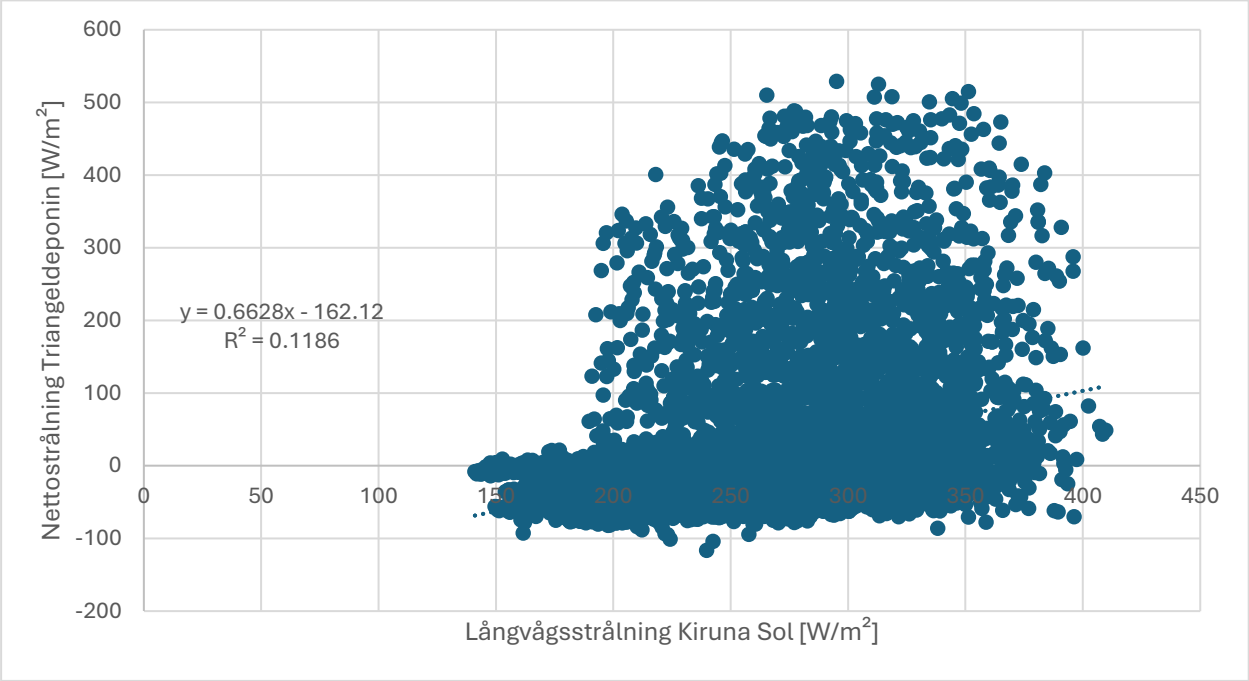
Figur 23. Dagligt medelvärde av vindhastighet (utan dataluckor) observerad på Triangeldeponin.

Bilaga C. Regressioner

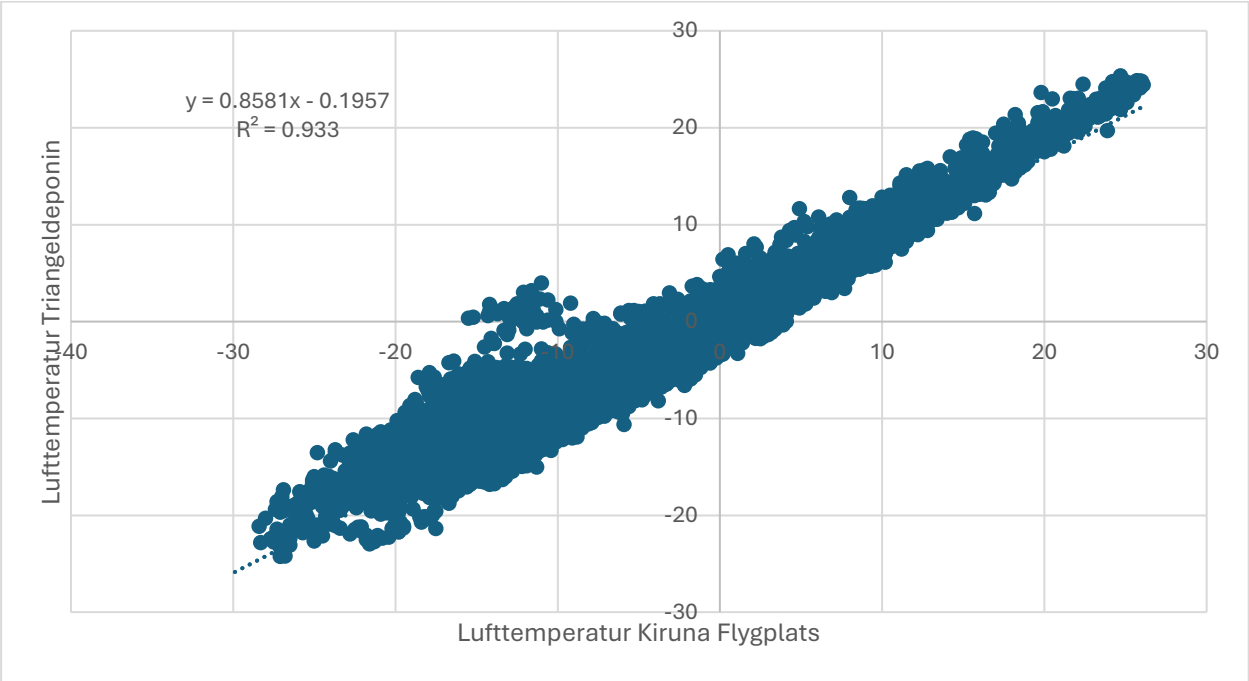
I denna bilaga presenteras regressioner från dataanalysen.



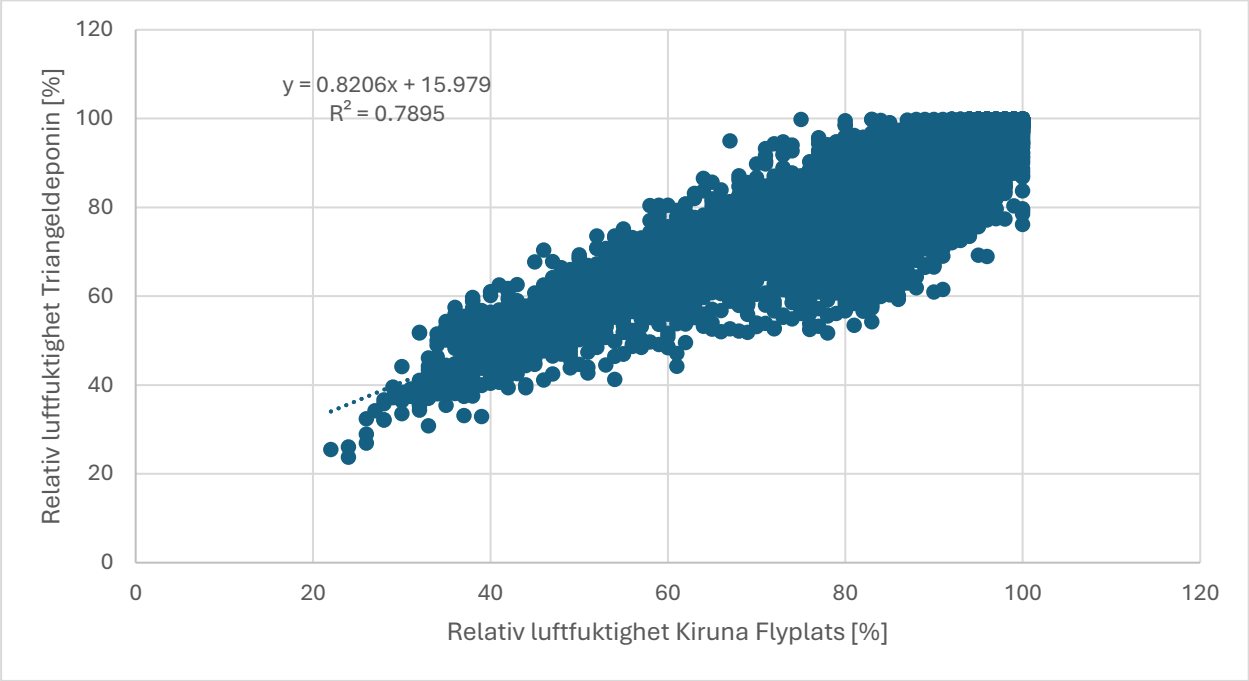
Figur 24. Regression mellan kortvågsstrålning fån SMHI:s mätstation ”Kiruna Sol” och observerad nettostrålning på Triangeldeponin.



Figur 25. Regression mellan långvågsstrålning fån SMHI:s mätstation "Kiruna Sol" och observerad nettostrålning på Triangeldeponin.



Figur 26. Regression mellan lufttemperatur från SMHI:s mätstation "Kiruna Flygplats" och observerad lufttemperatur på Triangeldeponin.



Figur 27. Regression mellan relativ luftfuktighet från SMHI:s mätstation ”Kiruna Flygplats” och observerad relativ luftfuktighet på Triangeldeponin.

Tabell 11. Regressionsstatisk och koefficienter framtagen med Excels dataanalysverktyg ”Regression”. Snö större än 0,5 och lufttemperatur för beräkning av emitterad värmestrålning.

Regressionsstatistik		**Koefficienter**	
Multipel-R	0,826541531	Konstant	30,97255816
R-kvadrat	0,683170902	Kortvågsstrålning	0,237380143
Justerad R-kvadrat	0,682979657	Långvågsstrålning	0,532049722
Standardfel	17,98135167	σT_a^4	-0,654382087
Observationer	4974		

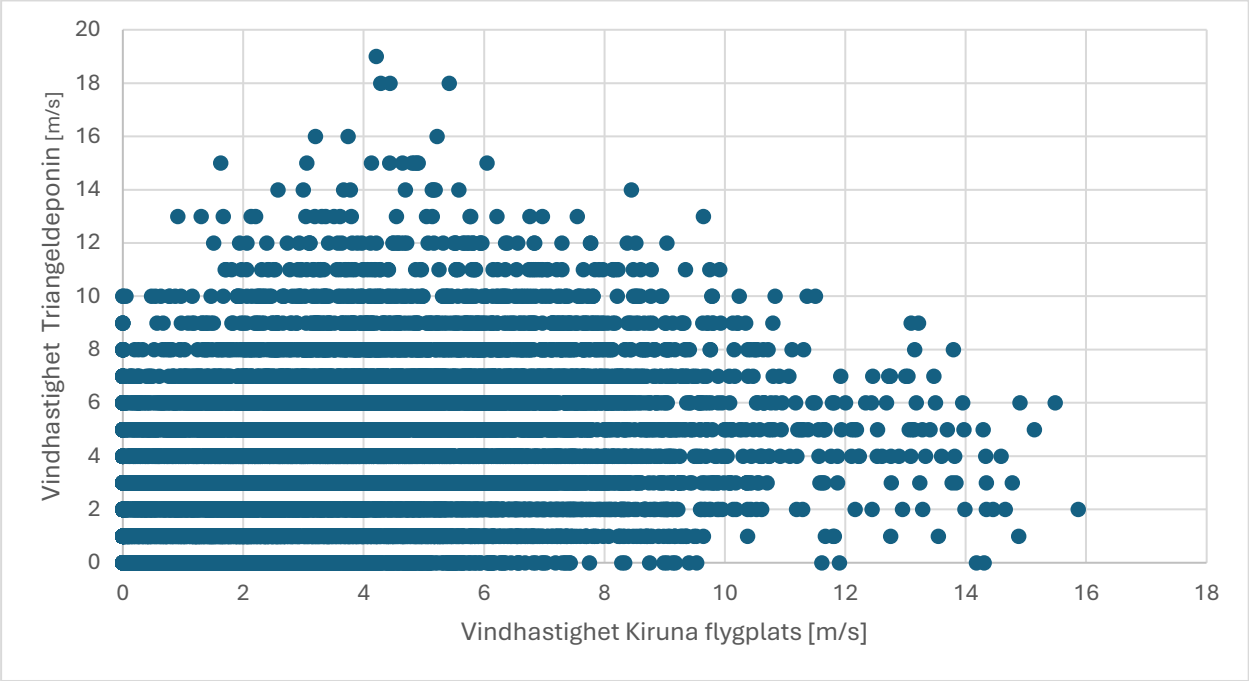
Tabell 12. Regressionsstatisk och koefficienter framtagen med Excels dataanalysverktyg ”Regression”. Ingen snö och lufttemperatur för beräkning av emitterad värmestrålning.

Regressionsstatistik		**Koefficienter**	
Multipel-R	0,929976968	Konstant	-87,06360943
R-kvadrat	0,864857161	Kortvågsstrålning	0,642013523
Justerad R-kvadrat	0,864778983	Långvågsstrålning	0,471727431
Standardfel	44,5931036	σT_a^4	-0,251161667
Observationer	5190		

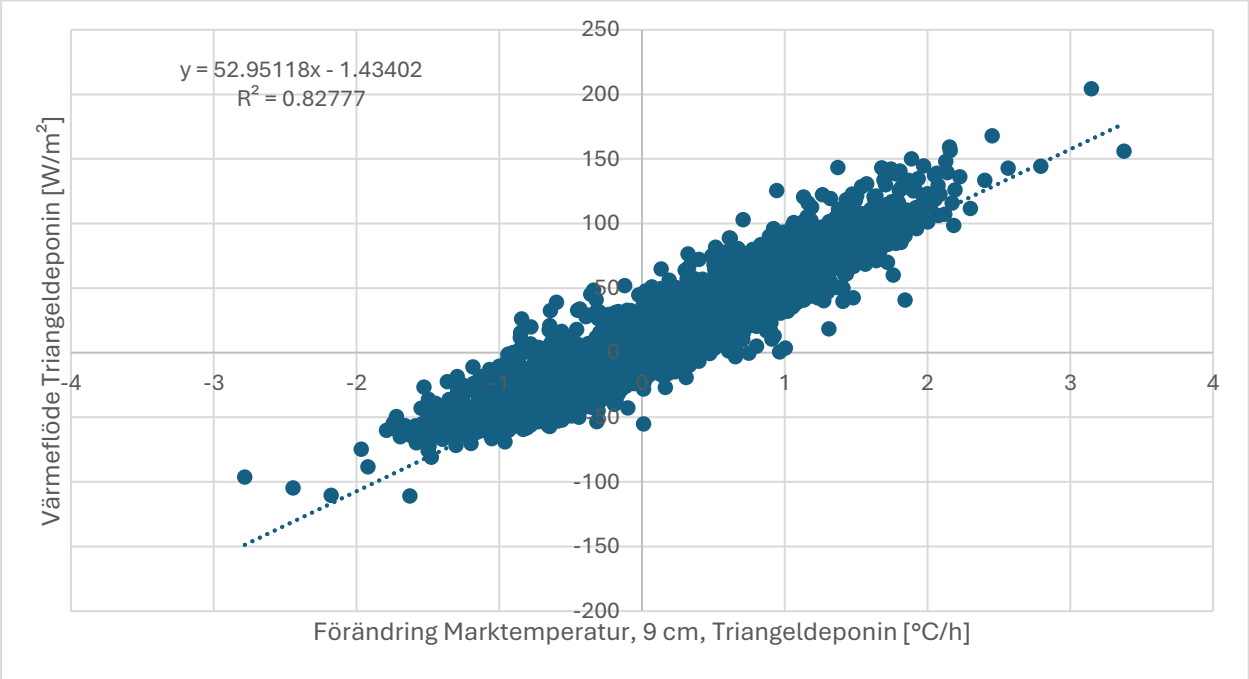
Tabell 13. Regressionsstatisk och koefficienter framtagen med Excels dataanalysverktyg ”Regression”. Snö större än 0 men mindre än 0,5 och marktemperatur 9 cm under markytan för beräkning av emitterad värmestrålning.

Regressionsstatistik		**Koefficienter**	
Multipel-R	0,888539256	Konstant	-87,051278
R-kvadrat	0,78950201	Kortvågsstrålning	0,5486505
Justerad R-kvadrat	0,787757551	Långvågsstrålning	0,6656497
Standardfel	32,07180549	σT_s^4	-0,3967445

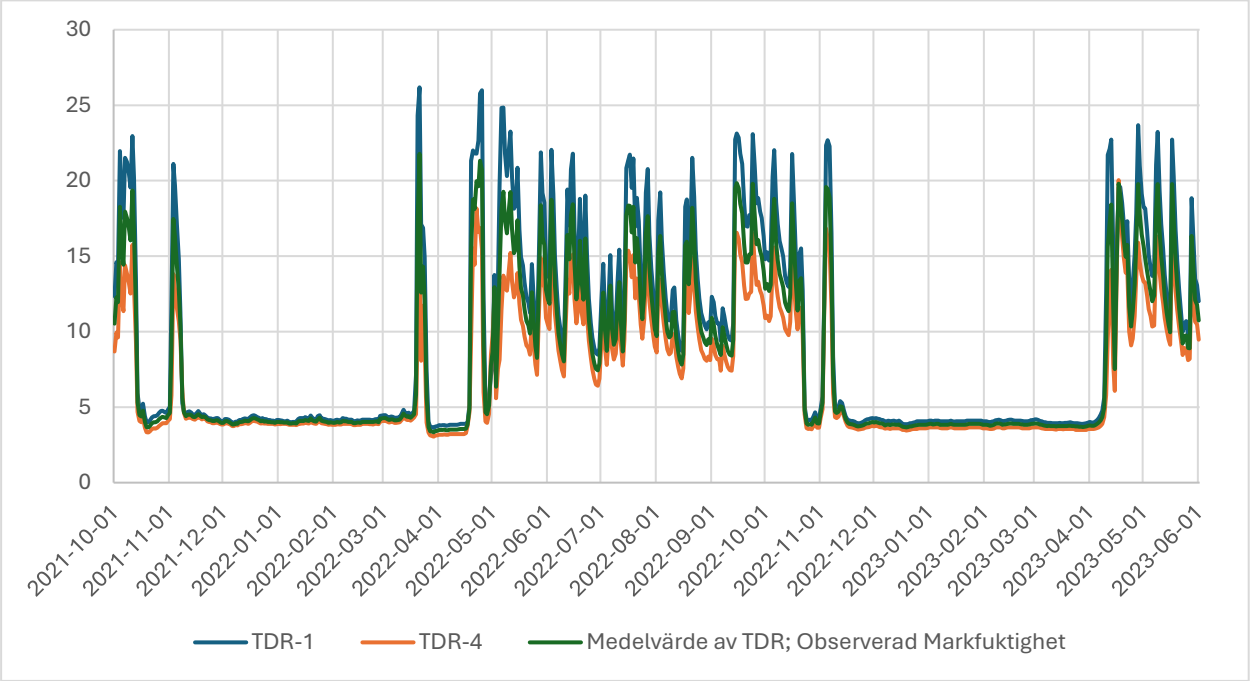
Observationer	366	



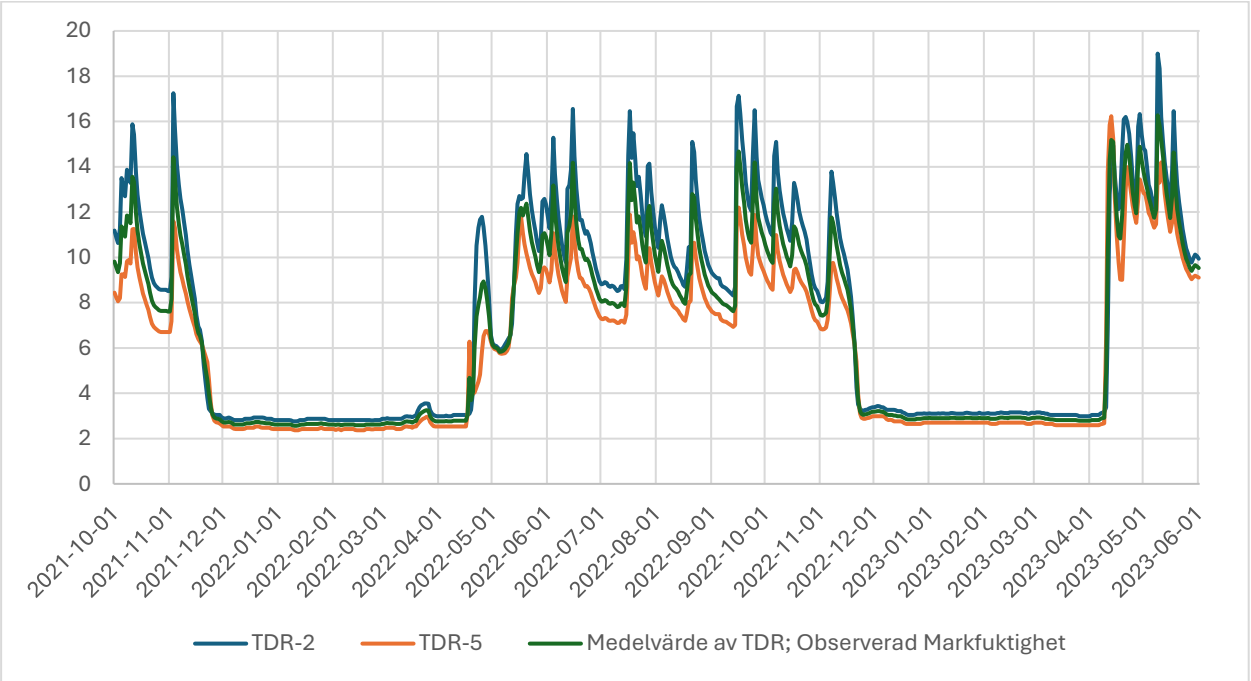
Figur 28. Regression mellan vindhastighet från SMHI:s mätstation ”Kiruna Flygplats” och observerad vindhastighet på Triangeldeponin.



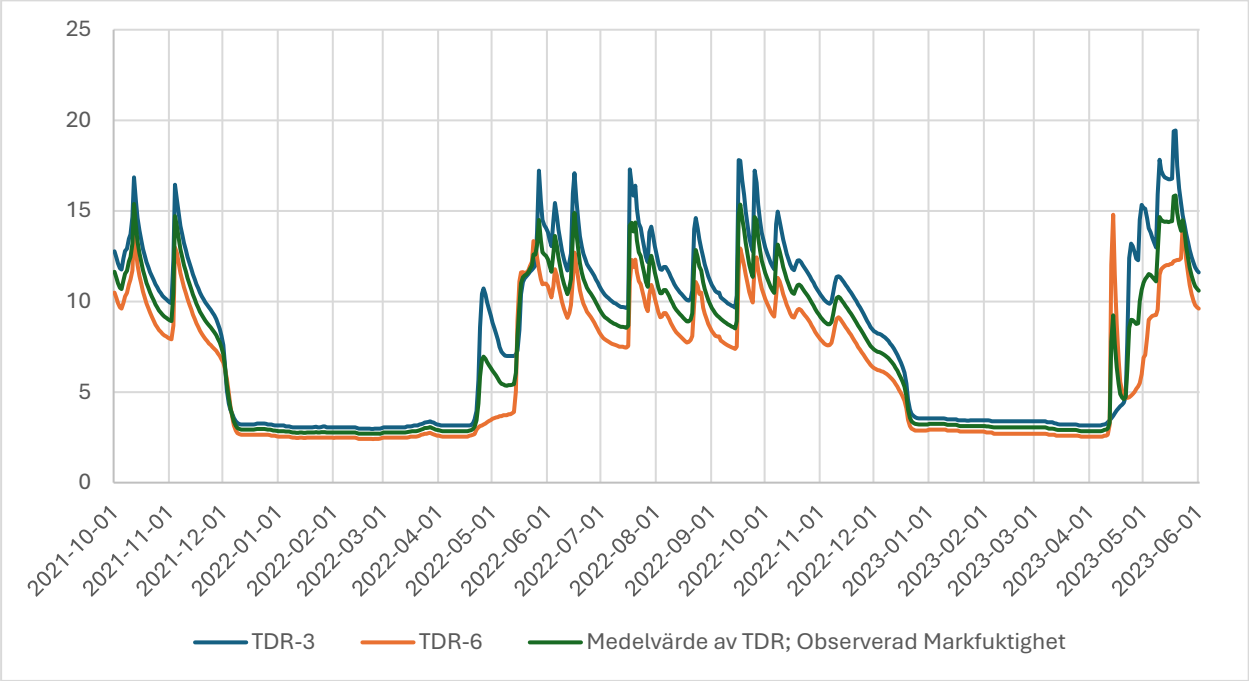
Figur 29. Regression mellan förändring av marktemperatur observerad av TDR-sensorer vid ett djup på 9 cm och observerade värmeflöde på Triangeldeponin.



Figur 30. Observerad markfuktighet för de båda TDR-sensorer vid 9 cm och dess medelvärde vilket använts för kalibrering i studien.



Figur 31. Observerad markfuktighet för de båda TDR-sensorer vid 62 cm och dess medelvärde vilket använts för kalibrering i studien.



Figur 32. Observerad markfuktighet för de båda TDR-sensorer vid 115 cm och dess medelvärde vilket använts för kalibrering i studien.

Bilaga D. Python skript

I denna bilaga skript för potentiell evaporation och den konceptuella modellen.

Potentiell evaporation

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import pyet



data = pd.read_excel("meteodata_triangeldeponi.xlsx", index_col=0, parse_dates=True, sheet_name="meteodata_filled")

meteo = pd.DataFrame({"time":data.index, "tmean":data.T_air_165, "tmax":data.Tmax, "tmin":data.Tmin, "rh":data.RH_air_165, "wind":data.Wind_Speed_2, "rn":data.Net_Radiation*(24*60**2/(10**6)), "rhmax":data.RHmax, "rhmin":data.RHmin, "g":data.GroundHeatFlow*(24*60**2/(10**6))})


time, tmean, tmax, tmin, rh, wind, rn, rhmax, rhmin, g = [meteo[col] for col in meteo.columns]


elevation = 555 # meters above sea-level

lat = 67.85572 * np.pi/180  # Latitude of the meteorological station, converting from degrees to radians

def pet_calc(combination=0, temperature=0, radiation=0, plot=True):
```

```
#Combination
if combination == 1:
    pet_kim_pen = pyet.kimberly_penman(tmean=tmean, wind = wind, tmax=tmax, tmin=tmin, lat=lat, rn=rn, rh=rh, elevation=elevation, g=g)
    pet_penman = pyet.penman(tmean=tmean, wind = wind, tmax=tmax, tmin=tmin, lat=lat, rn=rn, rh=rh, rhmax=rhmax, rhmin=rhmin, elevation=elevation, g=g)
    pet_pm = pyet.pm(tmean=tmean, wind = wind, tmax=tmax, tmin=tmin, lat=lat, rn=rn, rh=rh, rhmax=rhmax, rhmin=rhmin, elevation=elevation, g=g)
    pet_fao_56 = pyet.pm_fao56(tmean=tmean, wind = wind, tmax=tmax, tmin=tmin, lat=lat, rn=rn, rh=rh, rhmax=rhmax, rhmin=rhmin, elevation=elevation, g=g)
    pet_pri_tay = pyet.priestley_taylor(tmean=tmean, tmax=tmax, tmin=tmin, lat=lat, rn=rn, rh=rh, elevation=elevation, g=g)
    pet_thom_oliver = pyet.thom_oliver(tmean=tmean, wind = wind, tmax=tmax, tmin=tmin, lat=lat, rn=rn, rh=rh, rhmax=rhmax, rhmin=rhmin, elevation=elevation, g=g)

    if plot:
        fig, axs = plt.subplots(figsize=(13,5), ncols=2)
        pet_penman.plot(ax=axs[0])
        pet_pm.plot(ax=axs[0])
        pet_fao_56.plot(ax=axs[0])
        pet_kim_pen.plot(ax=axs[0])
        pet_pri_tay.plot(ax=axs[0])
        pet_thom_oliver.plot(ax=axs[0])

        pet_penman.cumsum().plot(ax=axs[1], legend=False)
        pet_pm.cumsum().plot(ax=axs[1], legend=False)
        pet_fao_56.cumsum().plot(ax=axs[1], legend=False)
        pet_kim_pen.cumsum().plot(ax=axs[1], legend=False)
        pet_pri_tay.cumsum().plot(ax=axs[1], legend=False)
        pet_thom_oliver.cumsum().plot(ax=axs[1], legend=False)

        axs[0].set_ylabel("PET [mm/day]", fontsize=12)
        axs[1].set_ylabel("Cumulative PET [mm]", fontsize=12)
        axs[0].legend(ncol=6, loc=[0,1.])
        axs[0].set_ylim(0, 8)
        axs[1].set_ylim(0, 800)
        axs[0].set_xlabel('')
```

```
        axs[1].set_xlabel('')
        plt.show()

    #Temperature
    if temperature == 1:
        pet_blaney_criddle = pyet.blaney_criddle(tmean=tmean, wind=wind, rhmin=rhmin, lat=lat, method = 0)
        pet_hamon = pyet.hamon(tmean=tmean, lat = lat, tmax=tmax, tmin=tmin, method = 2)
        pet_haude = pyet.haude(tmean, rh)
        pet_linacre = pyet.linacre(tmean = tmean, elevation = elevation, lat = lat, tmin=tmin, tmax=tmax)
        pet_rom = pyet.romanenko(tmean = tmean, rh=rh, tmin=tmin, tmax=tmax, rhmax=rhmax, rhmin=rhmin)



        if plot:
            fig, axs = plt.subplots(figsize=(13,5), ncols=2)
            pet_blaney_criddle.plot(ax=axs[0])
            pet_hamon.plot(ax=axs[0])
            pet_haude.plot(ax=axs[0])
            pet_linacre.plot(ax=axs[0])
            pet_rom.plot(ax=axs[0])

            pet_blaney_criddle.cumsum().plot(ax=axs[1], legend=False)
            pet_hamon.cumsum().plot(ax=axs[1], legend=False)
            pet_haude.cumsum().plot(ax=axs[1], legend=False)
            pet_linacre.cumsum().plot(ax=axs[1], legend=False)
            pet_rom.cumsum().plot(ax=axs[1], legend=False)

            axs[0].set_ylabel("PET [mm/day]", fontsize=12)
            axs[1].set_ylabel("Cumulative PET [mm]", fontsize=12)
            axs[0].legend(ncol=6, loc=[0,1.])
            axs[0].set_ylim(0, 8)
            axs[1].set_ylim(0, 800)
            axs[0].set_xlabel('')
            axs[1].set_xlabel('')
```

```
        plt.show()

    #Radiation
    if radiation == 1:
        pet_har = pyet.hargreaves(tmean, tmax, tmin, lat)
        pet_mcguinness_bordne = pyet.mcguinness_bordne(tmean=tmean, lat=lat)
        pet_oudin= pyet.oudin(tmean, lat)

        if plot:
            fig, axs = plt.subplots(figsize=(13,5), ncols=2)
            pet_har.plot(ax=axs[0])
            pet_mcguinness_bordne.plot(ax=axs[0])
            pet_oudin.plot(ax=axs[0])

            pet_har.cumsum().plot(ax=axs[1], legend=False)
            pet_mcguinness_bordne.cumsum().plot(ax=axs[1], legend=False)
            pet_oudin.cumsum().plot(ax=axs[1], legend=False)

            axs[0].set_ylabel("PET [mm/day]", fontsize=12)
            axs[1].set_ylabel("Cumulative PET [mm]", fontsize=12)
            axs[0].legend(ncol=6, loc=[0,1.])
            axs[0].set_ylim(0, 8)
            axs[1].set_ylim(0, 800)
            axs[0].set_xlabel('')
            axs[1].set_xlabel('')

            plt.show()
```

Konceptuell modell

Snörutin

```
import pandas as pd
import matplotlib.pyplot as plt
from utilities import *

def snow_routine(cfmax, tt, cwh, cfr, sfcf, is_weighted = False, create_file = False, plot=True):
```

"""

Perform a snow routine simulation based on input parameters.

Parameters:

cfmax (float): Maximum melt rate [mm/day].

tt (float): Threshold temperature [°C]. Temperature below which snow occurs.

cwh (float): Fraction of water holding capacity [-].

cfr (float): Refreezing coefficient [-].

sfcf (float): Snowfall correction factor [-].

plot (bool, optional): Whether to plot the results. Defaults to True.

is_weighted (bool, optional): Whether the modeled data should be weighted. Defaults to False.

create_file (bool, optional): Whether to create an output file. Defaults to False.

Returns:

float: Nash-Sutcliffe Efficiency (NSE) value.

Notes:3

02

This function performs a snow routine simulation based on the given parameters. It reads meteorological data

from an Excel file and calculates various snow-related variables such as snowfall, rainfall, melt,

refreezing, infiltration (amount of water that infiltrate the soil surface), and total snow mass.

Example:

snow_routine(cfmax=3.5, tt=0, cwh=0.1, cfr=0.05, sfcf=1)

"""

Load data

data=pd.read_excel("meteodata_triangeldeponi.xlsx", index_col=None, sheet_name="meteodata_filled")

meteo = pd.DataFrame({"date":data["TIMESTAMP"], "precip":data["Precip_Test"], "temp":data["T_air_165"], "s":data["Snow_SWE"], "weighted":data["Weighted_rapid"]})

date, precip, temp, s, weighted = [meteo[col] for col in meteo.columns]

Variables set to zero (or initialize with measured value) for the first time-step

```
    snow = [s[0]]  # snow amount, first time step initalize with snow
    wat = [0]   # liquid water
    swe = [s[0]]   # total mass of snow and water, first time step equal to the amount of snow (no liquid water)
    infiltration = [0]  # infiltration
    melt = [0]   # melt
    snowfall = [0]  # snowfall
    rainfall = [0]  # rainfall
    refreezing = [0]  # refreezing



    # Time-loop
    for t in range(1, len(precip)):
        if temp[t] <= tt:
            snowfall.append(sfcf * precip[t])  # calculate snowfall
            rainfall.append(0)  # calculate rainfall
            melt.append(0)  # calculate melt
            refreezing.append(cfr * cfmax * (tt - temp[t]))  # calculate refreezing

            # Refreezing should not exceed water amount
            if refreezing[t] > wat[t - 1]:
                refreezing[t] = wat[t - 1]
        else:
            snowfall.append(0)  # calculate snowfall
            rainfall.append(precip[t])  # calculate rainfall
            melt.append(cfmax * (temp[t] - tt))  # calculate melt

            # Melt should not exceed snow amount
            if melt[t] > snow[t - 1]:
                melt[t] = snow[t - 1]
            refreezing.append(0)  # calculate refreezing

        snow.append(snow[t - 1] + snowfall[t] - melt[t] + refreezing[t])
        # calculate new snow amount

        wat.append(wat[t - 1] + melt[t] + rainfall[t] - refreezing[t])
```

```
        # calculate water amount BEFORE infiltration

        # If water amount is larger than water holding capacity
        if snow[t] == 0:
            infiltration.append(wat[t])
            wat[t] = 0
        else:
            if wat[t]/snow[t] >= (cwh/(1-cwh)):
                infiltration.append(wat[t]- snow[t]*(cwh/(1-cwh)))  # calculate infiltration
                wat[t]= wat[t]-infiltration[t]  # calculate water amount (after infiltration)
            else:
                infiltration.append(0)  # calculate infiltration



        swe.append(snow[t] + wat[t])  # calculate the total snow mass (including water)

    # Check the water balance
    total_precipitation = sum(rainfall) + sum(snowfall)
    total_infiltration = sum(infiltration)

    tolerance = 0.1

    if abs(total_precipitation - total_infiltration + swe[0] - swe[-1]) > tolerance:
        print(f'Warning: Water balance check failed. Precipitation: {total_precipitation},
infiltration: {total_infiltration}, intial snow: {swe[0]}, snow at end: {swe[-1]}')

    # Calculate Nash-Sutcliffe Efficiency (NSE)
    if is_weighted:
        nse = MyUtilities.calculate_nse_weighted(s, swe, weighted)
    else:
        nse = MyUtilities.calculate_nse(s, swe)

    print(f'NSE: {nse}, cfmax: {cfmax}, tt: {tt}, cwh: {cwh}, cfr: {cfr}, sfcf: {sfcf}, infiltration:
{total_infiltration}, precipitation {sum(precip)}  date: {date[0]}-{date.iloc[-1]}')
```

```
    if create_file:
        infil_data = pd.DataFrame({'Date': date, 'infiltration': infiltration})
        infil_data.to_excel('Infiltration_data2.xlsx', index=False)

        model_data = pd.DataFrame({'Date': date, 'Infiltration': infiltration, 'Precipitation': precip, 'SWE': swe, 'Observed SWE': s, 'Temperature': temp})
        model_data.to_excel('model_data_snow_routine.xlsx', index=False)



    if plot:
        # Create the figure and axes
        fig, axes = plt.subplots(2, 1, figsize=(10, 12))

        # First subplot: snow data and temperature
        ax1 = axes[0]
        ax1.plot(date, swe, 'r', label='Modellerad Snö')
        ax1.plot(date, s, 'b', label='Uppmätt Snö')
        ax1.set_ylabel('Snö [mm]')
        ax1.tick_params(axis='x', rotation=45)

        # Create a secondary axis for temperature
        ax2 = ax1.twinx()
        ax2.plot(date, temp, 'y', label='Lufttemperatur')
        ax2.set_ylabel('Temperatur [°C]')

        # Synchronize the y-axes so that the zero levels align
        ax1_ylim = ax1.get_ylim()
        ax2_ylim = ax2.get_ylim()

        # Determine the minimum and maximum values for both axes
        ax1_min, ax1_max = min(ax1_ylim), max(ax1_ylim)
        ax2_min, ax2_max = min(ax2_ylim), max(ax2_ylim)

        # Calculate the new limits to synchronize the zero levels
        new_ax1_min = min(ax1_min, ax2_min)
```

```
new_ax1_max = max(ax1_max, ax2_max)
new_ax2_min = new_ax1_min
new_ax2_max = new_ax1_max

# Set the new limits
ax1.set_ylim(new_ax1_min, new_ax1_max)
ax2.set_ylim(new_ax2_min, new_ax2_max)

# Add legends
lines, labels = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax1.legend(lines + lines2, labels + labels2, loc='upper left')

# Second subplot: precipitation and infiltration
ax3 = axes[1]
infiltration_plot = np.array(infiltration)
ax3.plot(date, precip, 'c', label='Nederbörd')
ax3.plot(date, -infiltration_plot, 'g', label='Infiltration')
ax3.set_ylabel('Nederbörd och Infiltration [mm]')
ax3.legend(loc='upper left')
ax3.tick_params(axis='x', rotation=45)

# Set the x-axis limits to the range of your data
ax1.set_xlim([date.min(), date.max()])
ax3.set_xlim([date.min(), date.max()])

# Create ticks for even months
even_months = pd.date_range(start=date.min(), end=date.max(), freq='2MS')
ax1.set_xticks(even_months)
ax3.set_xticks(even_months)

# Set the date format on x-axis
ax1.xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
ax3.xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
```

```
        # Adjust layout
        plt.tight_layout()
        plt.show()

        # Statistics
        # 2021-10-01 - 2022-10-01
        print(f'Infiltration: {np.cumsum(infiltration)[365]}, precipitation: {np.cumsum(precip)[365]}')

        # 2021-10-01 - 2023-10-01
        infiltration = np.cumsum(infiltration)
        precip = np.cumsum(precip)
        print(f'Infiltration: {infiltration[-1]}, precipitation: {precip.iloc[-1]}')



    return nse
```

Markrutin

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from utilities import MyUtilities


def soil_routine(max_sm_frac, lp, k, k2, layers = 1, is_weighted = False, plot = True, create_file =False):
    """
    Calculate soil water storage, acutal evapoartion and recharge/percolating water.

    Args
    max_sm_frac (float): Maximum soil moisture as a fraction of layer depth.
    lp (float): Factor of maximum soil moisture which discribes when actual evaporation reaches potential evaporation.
    k (float): Linear storage coefficient
    k2 (float): Linear storage coefficient below layer 1
```

```
    Returns:
    float: Nash-Sutcliffe Efficiency (NSE) value
    """

    # Load data
    data = pd.read_excel("meteodata_triangeldeponi.xlsx", index_col=None,
sheet_name="meteodata_filled")
    infiltration_data = pd.read_excel("Infiltration_data2.xlsx", index_col=None)
    epot_data = pd.read_excel("PET_har.xlsx", index_col=None)

    # Extract columns
    date = data["TIMESTAMP"]
    infiltration = infiltration_data["infiltration"]
    epot = epot_data["Hargreaves"]




    if is_weighted:
        weighted = data["Weighted"]


    if layers == 1:
        depth_layer1 = 1200
        sm_measured = data["SoilWaterStorage"] #Calculated water storage for a layer with depth
1200
        max_sm = max_sm_frac * depth_layer1
        min_sm = 0.031 * depth_layer1 # minimum soil moisture.


        # Initialize arrays, sm with the initial soil moisture value (measured value)
        sm = [sm_measured[0]]
        recharge = [0]
        recharge_total = [0]
        eact = [0]
        eact_total = [0]


    elif layers == 2:
```

```
        depth_layer1 = 346
        depth_layer2 = 854
        sm_measured = data["SoilMoisture1"] * depth_layer1 #multiply with layerdepth
        max_sm = max_sm_frac*depth_layer1
        min_sm = 0.0365 * depth_layer1
        sm_measured2 = data["SoilMoisture2"] * depth_layer2 #multiply with layerdepth (1200-
346)
        min_sm2 = 0.026 * depth_layer2 #minimum soil moisture layer 2.

        # Initialize arrays, sm with the initial soil moisture value (measured value)
        sm = [sm_measured[0]]
        recharge = [0]
        recharge_total = [0]
        eact = [0]
        eact_total = [0]
        sm2 = [sm_measured2[0]]
        recharge2 = [0]
        recharge_total2 = [0]

    elif layers == 3:
        depth_layer1 = 346
        depth_layer2 = 530
        depth_layer3 = 324
        sm_measured = data["SoilMoisture1"] * depth_layer1 #multiply with layerdepth
        max_sm = max_sm_frac * depth_layer1
        min_sm = 0.0365 * depth_layer1
        sm_measured2 = data["SoilMoisture2"] * depth_layer2 #multiply with layerdepth (1200-
346)
        min_sm2 = 0.026 * depth_layer2
        sm_measured3 = data["SoilMoisture3"] * depth_layer3 #multiply with layerdepth (1200-
876)
        min_sm3 = 0.028 * depth_layer3 #minimum soil moisture layer 3.

        # Initialize arrays, sm with the initial soil moisture value (measured value)
        sm = [sm_measured[0]]
        sm2 = [sm_measured2[0]]
```

```
    sm3 = [sm_measured3[0]]
    recharge = [0]
    recharge2 = [0]
    recharge3 = [0]
    eact = [0]
    eact_total = [0]
    recharge_total = [0]
    recharge_total2 = [0]
    recharge_total3 = [0]



else:
    raise ValueError("layers is not an integer between 1 and 3 (including).")



# Iterate over each timestep
for i in range(0, len(date)):

    # Ensure sm, eact(_total) and recharge(_total) has an entry for each timestep
    if i != 0:
        sm.append(sm[-1])
        recharge.append(0)
        recharge_total.append(0)
        eact.append(0)
        eact_total.append(0)

        if layers >= 2:
            sm2.append(sm2[-1])
            recharge2.append(0)
            recharge_total2.append(0)

        if layers == 3:
            sm3.append(sm3[-1])
```

```
        recharge3.append(0)
        recharge_total3.append(0)


# Split current timestep into 1/delta_t timesteps
if infiltration[i] == 0:
    delta_t = 1/100

else:
    delta_t =1/(100*infiltration[i])
    delta_t_inv = int(1/delta_t)
    delta_t = 1/delta_t_inv


for j in range(int(1/delta_t)):
    if sm[-1]-min_sm > 0:
        recharge[-1] = MyUtilities.calculate_recharge_diffuse(sm[-1], k, min_sm)
        if recharge[-1] > sm[-1]-min_sm:
            recharge[-1] = sm[-1]-min_sm
    else:
        recharge[-1] = 0
    eact[-1] = MyUtilities.calculate_actual_evaporation_min(epot[i], sm[-1], max_sm, lp)
    recharge_total[-1] += recharge[-1]*delta_t
    eact_total[-1] += eact[-1]*delta_t
    sm[-1] += (infiltration[i] - eact[-1] - recharge[-1])*delta_t

    if layers >= 2:
        if sm2[-1] - min_sm2 > 0:
            recharge2[-1] = MyUtilities.calculate_recharge_diffuse(sm2[-1], k2, min_sm2)
            if recharge2[-1] > sm2[-1] - min_sm2:
                recharge2[-1] = sm2[-1] - min_sm2
        else:
            recharge2[-1] = 0
        recharge_total2[-1] += recharge2[-1]*delta_t
        sm2[-1] += (recharge[-1] - recharge2[-1])*delta_t
```

```
            if layers == 3:
                if sm3[-1] - min_sm3 > 0:
                    recharge3[-1] = MyUtilities.calculate_recharge_diffuse(sm3[-1], k2, min_sm3)
                    if recharge3[-1] > sm3[-1] - min_sm3:
                        recharge3[-1] = sm3[-1] - min_sm3
                else:
                    recharge3[-1] = 0
                recharge_total3[-1] += recharge3[-1]*delta_t
                sm3[-1] += (recharge2[-1] - recharge3[-1])*delta_t



    # Check the water balance
    total_infiltration = sum(infiltration)
    tolerance = 0.1
    total_evaporation = sum(eact_total)

    if layers == 1:
        total_recharge = sum(recharge_total)
        soil_water_change = sm[-1] - sm_measured[0]
        soil_water_change_year = sm[365] - sm_measured[0]



    elif layers == 2:
        total_recharge = sum(recharge_total2)
        soil_water_change = sm[-1] + sm2[-1] - sm_measured[0] - sm_measured2[0]
        soil_water_change_year = sm[365] + sm2[365] - sm_measured[0] - sm_measured2[0]


    else:
        total_recharge = sum(recharge_total3)
        soil_water_change = sm[-1] + sm2[-1] + sm3[-1] - sm_measured[0] - sm_measured2[0] -
sm_measured3[0]
        soil_water_change_year = sm[365] + sm2[365] + sm3[365] - sm_measured[0] -
sm_measured2[0] - sm_measured3[0]
```

```
    water_balance = total_infiltration  - total_recharge - soil_water_change - total_evaporation
    if abs(water_balance) > tolerance:
        print(f'Warning: Water balance check failed. Water balance: {water_balance}, infiltration: {total_infiltration}, change in water storage {soil_water_change}, recharge {total_recharge}, evaporation {total_evaporation}')

    #Calculate NNSE and printing results from the run
    if layers == 1:
        if is_weighted:
            nse = MyUtilities.calculate_nse_weighted(sm_measured, sm, weighted)

        else:
            nse = MyUtilities.calculate_nse(sm_measured, sm)

        nnse = MyUtilities.calculate_nnse(nse)

        print(f'NNSE: {nnse}, max_sm: {max_sm}, lp: {lp}, k: {k}, min_sm: {min_sm}, Potential evaporation: {np.sum(epot)} Actual evaporation: {np.sum(eact_total)}, Recharge {np.sum(recharge_total)}, Infiltration, {np.sum(infiltration)}')

    elif layers == 2:
        if is_weighted:
            nse1 = MyUtilities.calculate_nse_weighted(sm_measured, sm)
            nse2= MyUtilities.calculate_nse_weighted(sm_measured2, sm2)

        else:
            nse1 = MyUtilities.calculate_nse(sm_measured, sm)
            nse2 = MyUtilities.calculate_nse(sm_measured2, sm2)

        nnse1 = MyUtilities.calculate_nnse(nse1)
        nnse2 = MyUtilities.calculate_nnse(nse2)
        nnse = (nnse1*nnse2)**(1/2)

        print(f'NNSE: {nnse}, max_sm: {max_sm}, lp: {lp}, k: {k}, min_sm: {min_sm}, min_sm2: {min_sm2}, Potential evaporation: {np.sum(epot)} Actual evaporation: {np.sum(eact_total)}, Recharge {np.sum(recharge_total)}, Infiltration, {np.sum(infiltration)}')
```

```
    else:
        if is_weighted:
            nse1 = MyUtilities.calculate_nse_weighted(sm_measured, sm)
            nse2= MyUtilities.calculate_nse_weighted(sm_measured2, sm2)
            nse3= MyUtilities.calculate_nse_weighted(sm_measured3, sm3)

        else:
            nse1 = MyUtilities.calculate_nse(sm_measured, sm)
            nse2 = MyUtilities.calculate_nse(sm_measured2, sm2)
            nse3 = MyUtilities.calculate_nse(sm_measured3, sm3)

        nnse1 = MyUtilities.calculate_nnse(nse1)
        nnse2 = MyUtilities.calculate_nnse(nse2)
        nnse3 = MyUtilities.calculate_nnse(nse3)
        nnse = (nnse1*nnse2*nnse3)**(1/3)

        print(f'NSE1: {nse1}, NSE2: {nse2}, NSE3: {nse3}')
        print(f'NNSE: {nnse}, max_sm: {max_sm_frac}, lp: {lp}, k: {k}, min_sm: {min_sm}, min_sm2: {min_sm2}, min_sm3 {min_sm3}, Potential evaporation: {np.sum(epot)} Actual evaporation: {np.sum(eact_total)}, Recharge_3 {np.sum(recharge_total3)}, Infiltration, {np.sum(infiltration)}')

    if layers == 1:
        sm = np.array(sm)*100/depth_layer1
        sm_measured = np.array(sm_measured)*100/depth_layer1


    elif layers == 2:
        sm = np.array(sm)*100/depth_layer1
        sm2 = np.array(sm2)*100/depth_layer2
        sm_measured = np.array(sm_measured)*100/depth_layer1
        sm_measured2 = np.array(sm_measured2)*100/depth_layer2

    else:
        sm = np.array(sm)*100/depth_layer1
        sm2 = np.array(sm2)*100/depth_layer2
```

```
        sm3 = np.array(sm3)*100/depth_layer3
        sm_measured = np.array(sm_measured)*100/depth_layer1
        sm_measured2 = np.array(sm_measured2)*100/depth_layer2
        sm_measured3 = np.array(sm_measured3)*100/depth_layer3



    if create_file:
        if layers == 1:
            model_data = pd.DataFrame({'Date': date, 'ModeledSoilMoisture1': sm, 'SoilMoisutre1':
sm_measured, 'Recharge': recharge,
                                       'Infiltration': infiltration, 'ActualEvaporation': eact,
'PotentialEvaporation': epot})
            model_data.to_excel('Soil_routine_one_layer.xlsx', index=False)


        elif layers == 2:
            model_data = pd.DataFrame({'Date': date, 'ModeledSoilMoisture1': sm,
'ModeledSoilMoisture2': sm2,
                                       'SoilMoisutre1': sm_measured, 'SoilMoisutre2': sm_measured2,
'Recharge': recharge,
                                       'Recharge2': recharge2, 'Infiltration': infiltration, 'ActualEvaporation':
eact,
                                       'PotentialEvaporation': epot})
            model_data.to_excel('Soil_routine_two_layers.xlsx', index=False)


        else:
            model_data = pd.DataFrame({'Date': date, 'ModeledSoilMoisture1': sm,
'ModeledSoilMoisture2': sm2,
                                       'ModeledSoilMoisture3': sm3, 'SoilMoisutre1': sm_measured,
'SoilMoisutre2': sm_measured2,
                                       'SoilMoisutre3': sm_measured3, 'Recharge': recharge, 'Recharge2':
recharge2,
                                       'Recharge3': recharge3, 'Infiltration': infiltration, 'ActualEvaporation':
eact,
                                       'PotentialEvaporation': epot})
            model_data.to_excel('Soil_routine_three_layers.xlsx', index=False)


    #Create figures of result from the run
    if plot:
```

```
even_months = pd.date_range(start=date.min(), end=date.max(), freq='2MS')

if layers == 1:

    plt.figure(figsize=(10, 6))
    plt.plot(date, sm, label='Modeled Soil Moisture')
    plt.plot(date, sm_measured, label='Measured Soil Moisture')
    plt.ylabel('Soil Moisture (%)')
    plt.title('Soil Moisture Over Time (Layer 1)')
    plt.legend()
    plt.xticks(even_months, rotation=45)
    ax = plt.gca()  # Get the current Axes
    ax.xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
    plt.tight_layout()
    plt.show()

elif layers == 2:

    fig, axes = plt.subplots(2, 1, figsize=(10, 12))
    axes[0].plot(date, sm, label='Modeled Soil Moisture (Layer 1)')
    axes[0].plot(date, sm_measured, label='Measured Soil Moisture (Layer 1)')
    axes[0].set_ylabel('Soil Moisture (%)')
    axes[0].set_title('Soil Moisture Over Time (Layer 1)')
    axes[0].legend()
    axes[0].tick_params(rotation=45)
    axes[0].set_xticks(even_months)
    axes[0].xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))

    axes[1].plot(date, sm2, label='Modeled Soil Moisture (Layer 2)')
    axes[1].plot(date, sm_measured2, label='Measured Soil Moisture (Layer 2)')
    axes[1].set_ylabel('Soil Moisture (%)')
    axes[1].set_title('Soil Moisture Over Time (Layer 2)')
    axes[1].legend()
    axes[1].tick_params(rotation=45)
    axes[1].set_xticks(even_months)
```

```
    axes[1].xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))


    plt.tight_layout()
    plt.show()

elif layers == 3:

    # Create the figure and axes
    fig, axes = plt.subplots(3, 1, figsize=(10, 12))

    # First subplot:
    axes[0].plot(date, sm, label='Modeled Soil Moisture (Layer 1)')
    axes[0].plot(date, sm_measured, label='Measured Soil Moisture, depth: 9 cm')
    axes[0].set_ylabel('Soil Moisture [%]')
    axes[0].legend()
    axes[0].tick_params(rotation=45)
    axes[0].set_xlim([date.min(), date.max()])
    axes[0].set_xticks(even_months)
    axes[0].xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
    axes[0].set_ylim(0, 25)

    # Second subplot:
    axes[1].plot(date, sm2, label='Modeled Soil Moisture (Layer 2)')
    axes[1].plot(date, sm_measured2, label='Measured Soil Moisture, depth: 62 cm')
    axes[1].set_ylabel('Soil Moisture [%]')
    axes[1].legend()
    axes[1].tick_params(rotation=45)
    axes[1].set_xlim([date.min(), date.max()])
    axes[1].set_xticks(even_months)
    axes[1].xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
    axes[1].set_ylim(0, 25)

    # Third subplit
    axes[2].plot(date, sm3, label='Modeled Soil Moisture (Layer 3)')
```

```
    axes[2].plot(date, sm_measured3, label='Measured Soil Moisture, depth: 115 cm')
    axes[2].set_ylabel('Soil Moisture [%]')
    axes[2].legend()
    axes[2].tick_params(rotation=45)
    axes[2].set_xlim([date.min(), date.max()])
    axes[2].set_xticks(even_months)
    axes[2].xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
    axes[2].set_ylim(0, 25)

    plt.tight_layout()
    plt.show()



eact = np.array(eact_total)
if layers == 1:
    recharge_plot = np.array(recharge_total)
elif layers == 2:
    recharge_plot = np.array(recharge_total2)
else:
    recharge_plot = np.array(recharge_total3)

# Calculate cumulative sums
cumulative_eact = np.cumsum(eact)
cumulative_recharge = np.cumsum(recharge_plot)

# Create the figure and axes
fig, axes = plt.subplots(2, 1, figsize=(10, 12))

# First subplot: evaporation and recharge
axes[0].plot(date, recharge_plot,  label='Perkolation')
axes[0].plot(date, eact,  label='Faktisk Evaporation')
axes[0].set_ylabel('Perkolation och Faktisk Evaporation [mm]')
axes[0].legend()
axes[0].tick_params(axis='x', rotation=45)
axes[0].set_xlim([date.min(), date.max()])
```

```
    axes[1].plot(date, cumulative_recharge, label='Kumulativ Perkolation')
    axes[1].plot(date, cumulative_eact, label='Kumulativ Faktisk Evaporation')
    axes[1].set_ylabel('Perkolation och Faktisk Evaporation [mm]')
    axes[1].legend()
    axes[1].tick_params(axis='x', rotation=45)
    axes[1].set_xlim([date.min(), date.max()])
    axes[1].set_xlim([date.min(), date.max()])

    even_months = pd.date_range(start=date.min(), end=date.max(), freq='2MS')
    axes[0].set_xticks(even_months)
    axes[1].set_xticks(even_months)

    axes[0].xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
    axes[1].xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))

    # Adjust layout
    plt.tight_layout()
    plt.show()



    # Statistics
    # 2021-10-01 - 2022-10-01
    print(f'Recharge: {cumulative_recharge[365]}, deltaS: {soil_water_change_year}, Evapoartion: {cumulative_eact[365]}')


    # 2021-10-01 - 2023-10-01
    print(f'Recharge: {cumulative_recharge[-1]}, deltaS: {soil_water_change}, Evapoartion: {cumulative_eact[-1]}')




  # Return Nash-Sutcliffe Efficiency (NSE)
  return nnse
```

Samlat skript

```
import numpy as np
```

```
import pandas as pd
import csv


class MyUtilities:

    def calculate_nse_weighted(measured_values, modeled_values, weighted=None):
        """
        Calculate Nash-Sutcliffe Efficiency (NSE) between measured and modeled values.

        Args:
        measured_values (list): List of measured values.
        modeled_values (list): List of modeled values.
        weighted (list): Optional list of weights for timesteps.

        Returns:
        float: Nash-Sutcliffe Efficiency (NSE) value.
        """
        # Create an array of ones with the same length as date
        weighted_array = np.ones(len(measured_values))

        # Apply weighting if provided
        if weighted is not None:
            if len(weighted) == len(measured_values):
                weighted_array = np.multiply(weighted_array, weighted)
            else:
                raise ValueError("Length of 'weighted' array must be equal to the length of
'measured_values'.")



        # Convert lists to numpy arrays for easier computation
        measured_values = np.array(measured_values)
        modeled_values = np.array(modeled_values)

        # Calculate mean of measured values
        mean_measured = np.mean(measured_values)
```

```
    # Calculate numerator and denominator for NSE with or without weights
    numerator = np.sum((weighted_array * (modeled_values - measured_values)) ** 2)
    denominator = np.sum((weighted_array * (measured_values - mean_measured)) ** 2)

    # Calculate NSE
    nse = 1 - (numerator / denominator)

    return nse

def calculate_nse(measured_values, modeled_values):
    """
    Calculate Nash-Sutcliffe Efficiency (NSE) between measured and modeled values.

    Args:
    measured_values (list): List of measured values.
    modeled_values (list): List of modeled values.

    Returns:
    float: Nash-Sutcliffe Efficiency (NSE) value.
    """
    # Convert lists to numpy arrays for easier computation
    measured_values = np.array(measured_values)
    modeled_values = np.array(modeled_values)


    # Calculate mean of measured values
    mean_measured = np.mean(measured_values)

    # Calculate numerator and denominator for NSE
    numerator = np.sum((modeled_values - measured_values) ** 2)
    denominator = np.sum((measured_values - mean_measured) ** 2)

    # Calculate NSE
    nse = 1 - (numerator / denominator)
```

```
    return nse

def calculate_nnse(nse):
    """
    Calculate normalized Nash-Sutcliffe Efficiency (NSE) between measured and modeled values.

    Args:
    nse (float): Nash-Sutcliffe Efficiency (NSE) value.
    layers (int): Modeled layers.

    Returns:
    float: Normalized Nash-Sutcliffe Efficiency (NSE) value.
    """

    return 1/(2-nse)



def create_csv(date, exfiltration):
    data = zip (date, exfiltration)
    csv_filename = "Infiltrating_water.csv"
    with open(csv_filename, mode='w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['Date', 'Infiltrated_water'])
        writer.writerows(data)

def create_csv(date, exfiltration):
    data = zip (date, exfiltration)
    csv_filename = "Infiltrating_water.csv"
    with open(csv_filename, mode='w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['Date', 'Infiltrated_water'])
        writer.writerows(data)
```

```
def calculate_recharge(infil, sm, max_sm, beta):
    """
    Calculate recharge based on the given parameters.

    Args:
    infil (float): Infiltration of precipitation or snowmelt.
    sm (float): Soil moisture storage.
    max_sm (float): Maximum soil moisture.
    beta (float): Fitting parameter.

    Returns:
    float: Recharge value.
    """
    return max(0, ((sm / max_sm) ** beta) * infil)

def calculate_recharge_diffuse(sm, k, min_sm):
    """
    Calculate (diffuse) recharge based on the given parameters.

    Args:
    sm (float): Soil moisture storage.
    min_sm (float): Minimum soil moisture.
    beta (float): Fitting parameter.

    Returns:
    float: Recharge value.
    """
    return max(0, (k * (sm-min_sm)))

def calculate_actual_evaporation(epot, sm, max_sm, lp):
    """
    Calculate actual evaporation based on the given parameters.

    Args:
    epot (float): Potential evaporation.
```

```
        sm (float): Soil moisture storage.
        max_sm (float): Maximum soil moisture.
        lp (float): Factor.

        Returns:
        float: Actual evaporation value.
        """
        return min(epot, (sm/(max_sm * lp)) * epot)

    def calculate_actual_evaporation_min(epot, sm, max_sm, lp, min_sm_e=0):
        """
        Calculate actual evaporation based on the given parameters.

        Args:
        epot (float): Potential evaporation.
        sm (float): Soil moisture storage.
        max_sm (float): Maximum soil moisture.
        lp (float): Factor.
        min_sm_e(float): Minimum soil moisture possible for evapoartion

        Returns:
        float: Actual evaporation value.
        """
        return max(0, min(epot, ((sm-min_sm_e)/((max_sm-min_sm_e) * lp)) * epot))
```

Snörutin calibration

```
from scipy.optimize import differential_evolution
import snow_routine



# Define the objective function
def objective_function(params):
    CFMAX, TT, CWH, CFR, SFCF = params
    # Call on the funktion snowrutin
    nse = snow_routine.snow_routine(CFMAX, TT, CWH, CFR, SFCF, plot=False,
is_weighted=True)
    # Return NSE-value
```

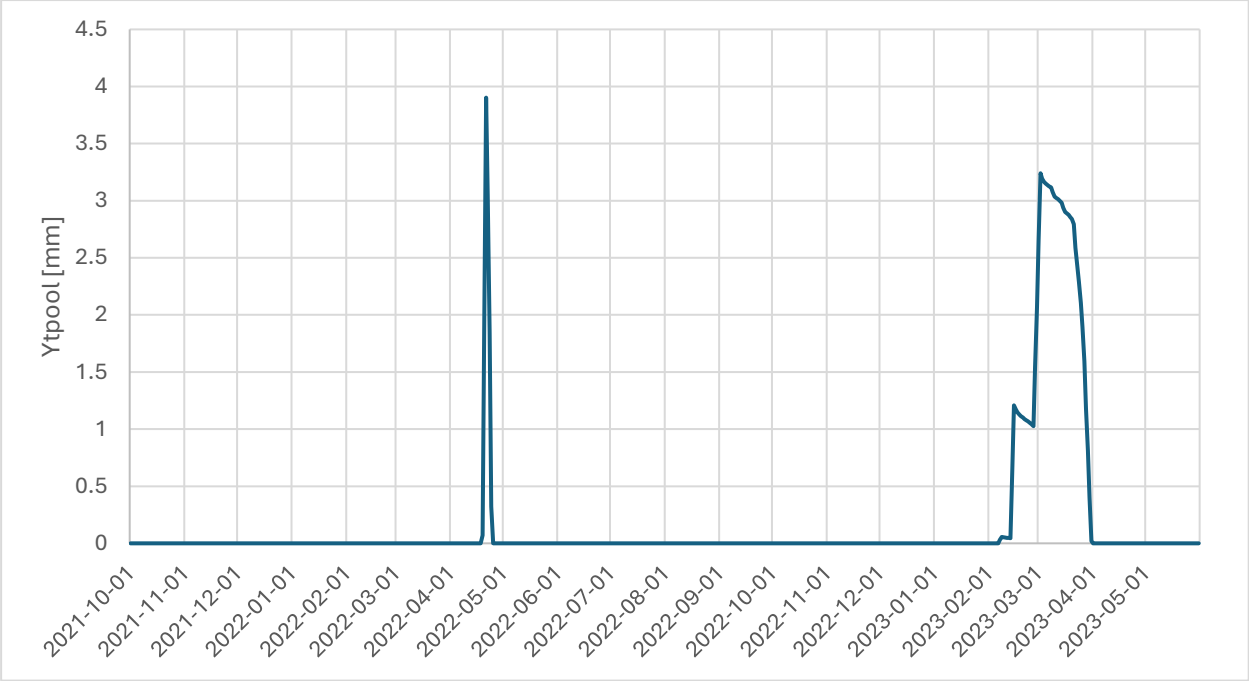
```
    return -nse   # Negativ to minimize

# Define limitaions of parameters
bounds = [(2, 8), (-2, 2), (0.05, 0.1), (0.05, 0.1), (0.1, 2)]

# Starting values of the parameters
x0 = [3, 0, 0.1, 0.05, 1]
x0 = [3, 0, 0.099, 0.059, 1]

# Use differential evolution to optimize parameteres
result = differential_evolution(objective_function, bounds, x0=x0, tol=0.01, seed=42)

# Best parameters
best_params = result.x
best_nse = -result.fun
print("Best parameters:", best_params)
print("Best Nash-Sutcliffe Efficiency:", best_nse)
```

Markrutin kalibration

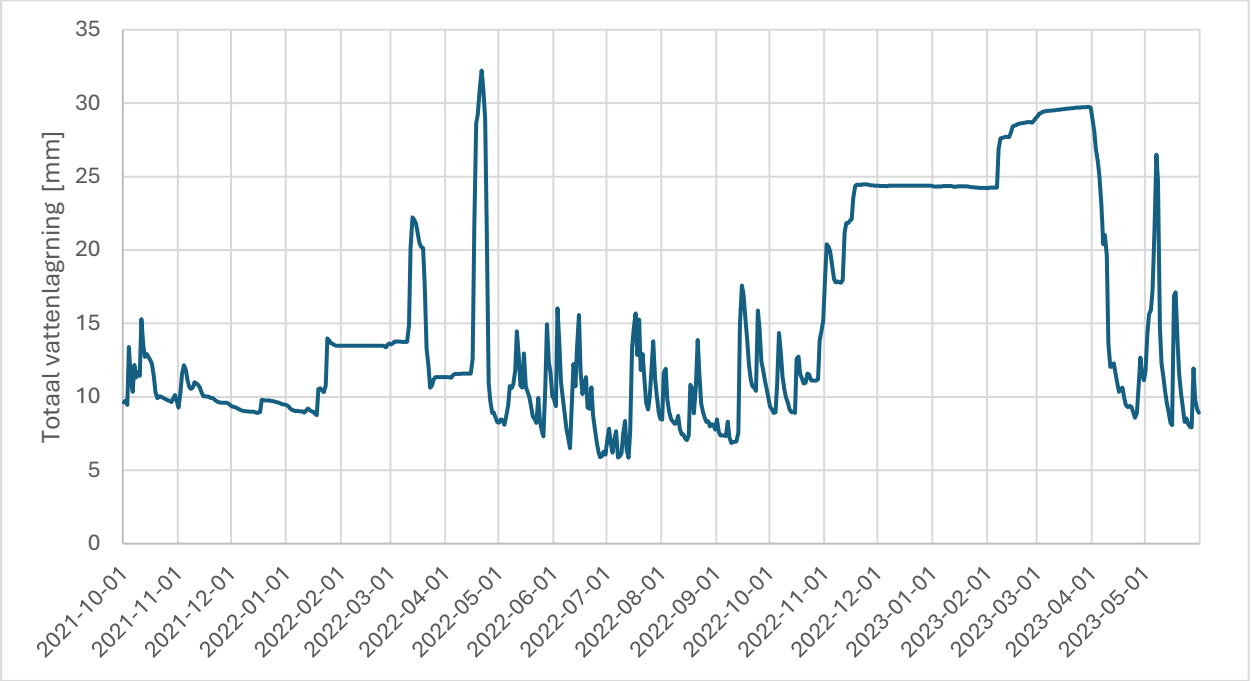
```
from scipy.optimize import differential_evolution
import soil_routine_test

# Define the objective function
def objective_function(params):
    max_sm_frac, lp, k, k2 = params
    # Call on the funktion
    nnse = soil_routine_test.soil_routine(max_sm_frac, lp, k, k2, plot=False, layers = 3)
    # Return NNSE-value
    return -nnse  # Negativ to minimize

# Define limitaions of parameters
bounds = [(0, 2), (0, 1), (0, 1), (0, 1)]

# Starting values of the parameters
x0 = [1, 0.5, 0.1,  0.1]
```

```
# Använd differential evolution för att optimera parametrarna
result = differential_evolution(objective_function, bounds, x0=x0, tol=0.01, seed=42)

# Best parameters
best_params = result.x
best_nnse = -result.fun
print("Best parameters:", best_params)
print("Best Normalized Nash-Sutcliffe Efficiency:", best_nnse)
```

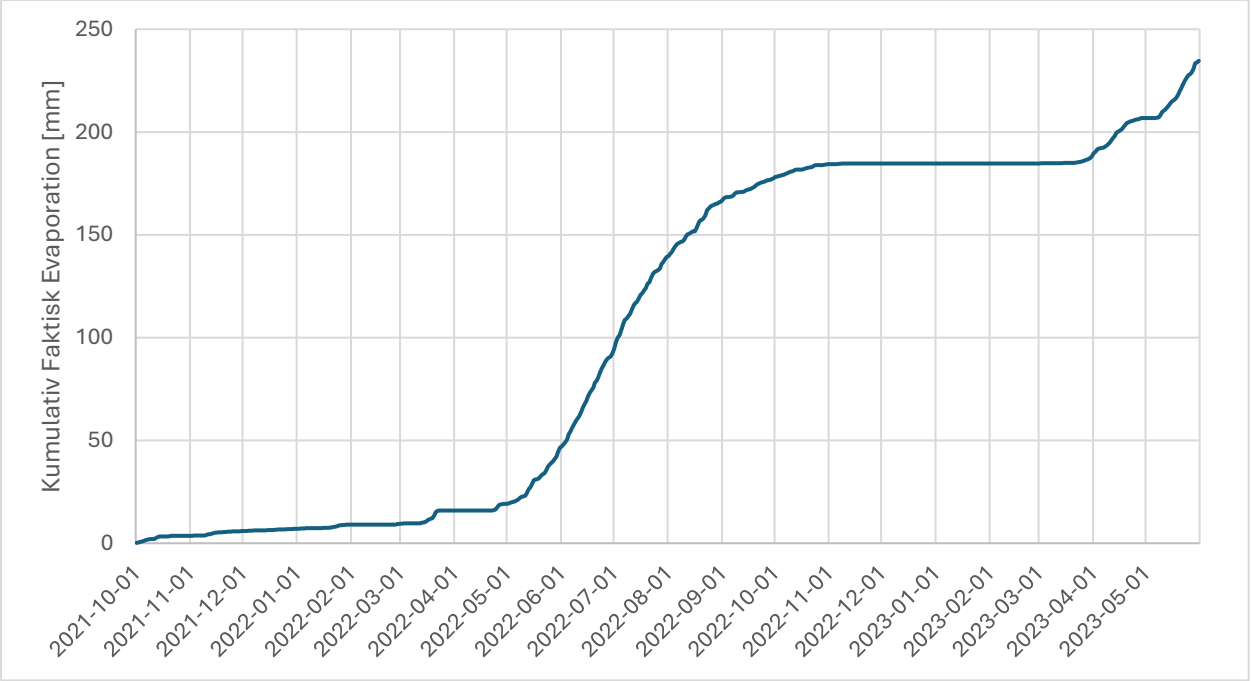
Bilaga E. CoupModel figurer



Figur 33. Vattenpool ovan ytan (surface pool).



Figur 34. Total vattenlagring i det översta modellerade lagret i CoupModel.



Figur 35. Kumulativ evaporation utan kondensation.