

Einsatz von Scripting und die Modellierung von Steuerungen

Dipl.-Math Benedikt Rothe



- ▶ Was ist Scripting?
- ▶ Wie wird Scripting eingesetzt?
 - ▶ Standardanwendungen
 - ▶ Fortgeschrittene Anwendungen
 - ▶ Exoten



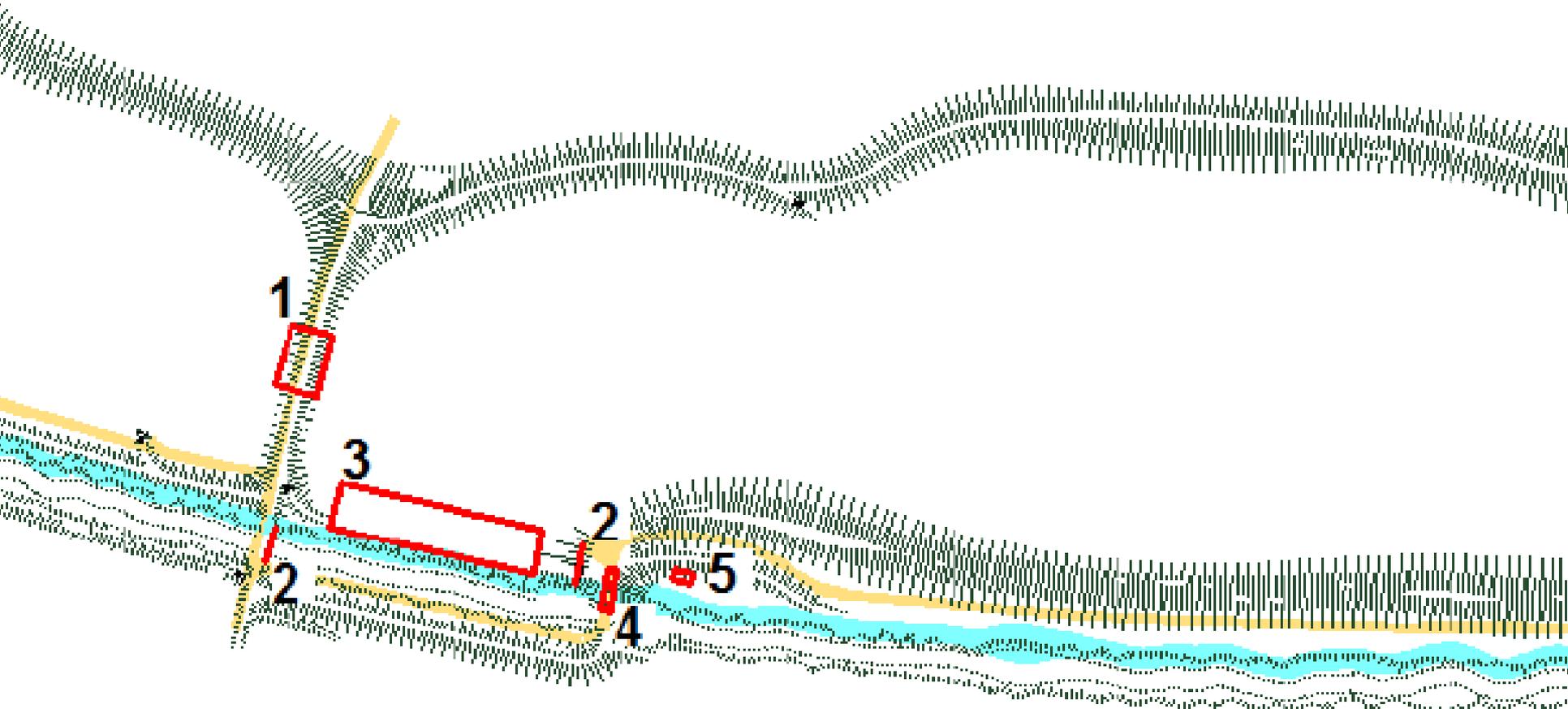
IT-Sicht: Was ist Scripting?

- Modellierer schreibt (kleines) Programm
 - Ausführung in jedem internen Zeitschritt
- Fragt HYDRO_AS-2D-Daten ab
 - Wasserstand Pegel X?
 - Durchfluss beim Kontrollquerschnitt Z?
 - ...
- Verändert HYDRO_AS-2D-Daten
 - Z-Wert eines Knotens
 - Größe eines Durchlasses
 - ...
- Führt Datenverarbeitung durch
 - Berechnungen
 - Dateien lesen/schreiben
 - Programme aufrufen
 - ...

```

- the weir shall be opened to flood the polder
: weirStatus == 0 and gauge1:w() > 104 then
-- lower the weir
weir:setZ(100)
zWriter.write()
weirStatus = 1
print("***** Open weir:", hydroas.Global.timestepStart())
- if flood has finished weir shall be closed
.seif weirStatus == 1 and gauge1:w() < 101 then
-- raise the weir to it's original height
weir:setZ(109.96)
zWriter.write()
weirStatus = 0
print("***** Close weir:", hydroas.Global.timestepStart())
id
:I: Polder
- Outlet is closed: free capacity in river and Polder is filled?
- Open Outlet
: outlet:qMax() == 0 and gauge2:w() < 96 and gaugePolder:h() > 1 then
outlet:setQMax(outletQMax)
print("***** Open outlet:", hydroas.Global.timestepStart())
- Outlet is open: water in river is too high: Close it
.seif outlet:qMax() > 0 and gauge2:w() > 99 then
outlet:setQMax(0)
print("***** Close outlet:", hydroas.Global.timestepStart(

```



- ▶ Phänomen:
 - ▶ Starkregenabfluss über Straße
 - ▶ Wasserstand über Bordstein: Abfluss in tiefer gelegene Senke
 - ▶ Beispiel: Tiefgarage, U-Bahn,...
 - ▶ Niederschlag/Abfluss auf Bebauung oberhalb:
Unterirdische Senke nicht als Senke im 2D-Modell

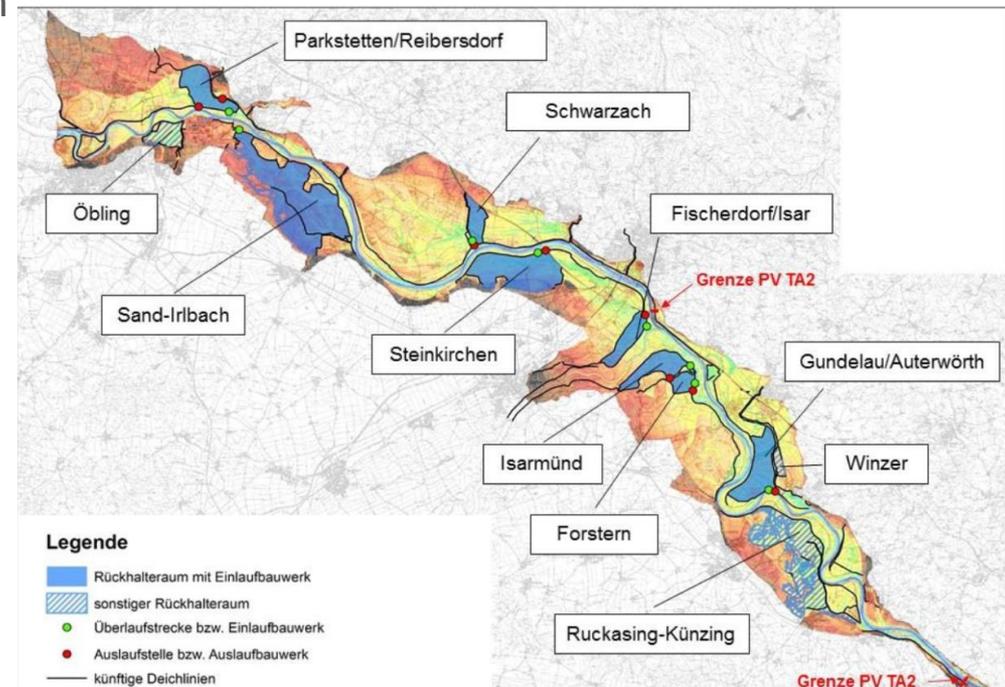
- ▶ Modellierung
 - ▶ Auslaufrandbedingung am Bordstein
 - ▶ Scripting:
 - ▶ Senke gefüllt: $Q=0$
 - ▶ Ausbauzustand: Wasserstand Senke kontinuierlich berücksichtigen



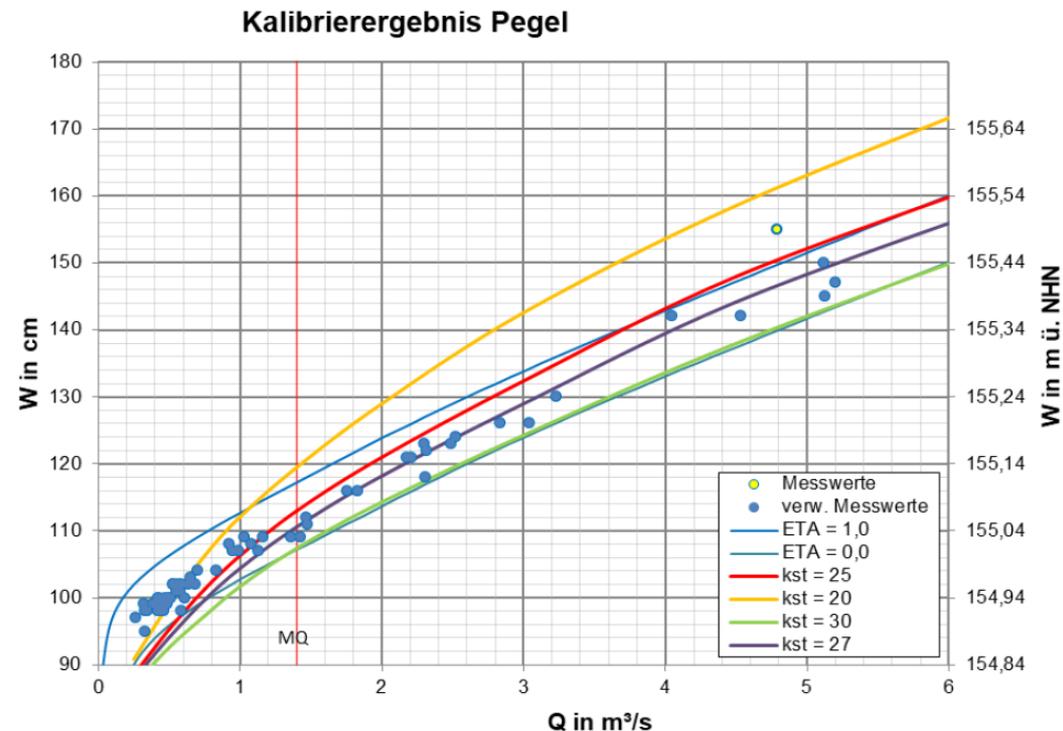
Quelle: <https://www.hessenschau.de>

- ▶ 11:15 Uhr: „Kopplung von HYDRO_AS-2D mit einem Kanalnetzmodell“ (Robert Mittelstädt)
- ▶ 1D-Kanalnetzmodell SWMM
- ▶ SWMM und HYDRO_AS-2D rechnen jeweils ca. 1 Minute
- ▶ Modellierer identifiziert jeweilige Modellobjekte:
„Straßeneinläufe/Gullys SWMM“ ↔ „Knoten HYRO_AS-2D“
- ▶ Überstau SWMM: Positiver Knoten-Quellterm in HYDRO_AS-2D
- ▶ Einlauf SWMM: Negativer Knoten-Quellterm in HYDRO_AS-2D
- ▶ LUA-Script
 - ▶ HYDRO_AS-2D Knoten abfragen
 - ▶ Eingabedaten für SWMM erstellen
 - ▶ SWMM aufrufen
 - ▶ SWMM auswerten
 - ▶ Ergebnisse auf HYDRO_AS-2D übertragen
- ▶ Komplexes Script – Weitergabe an Kunden bei Schulung

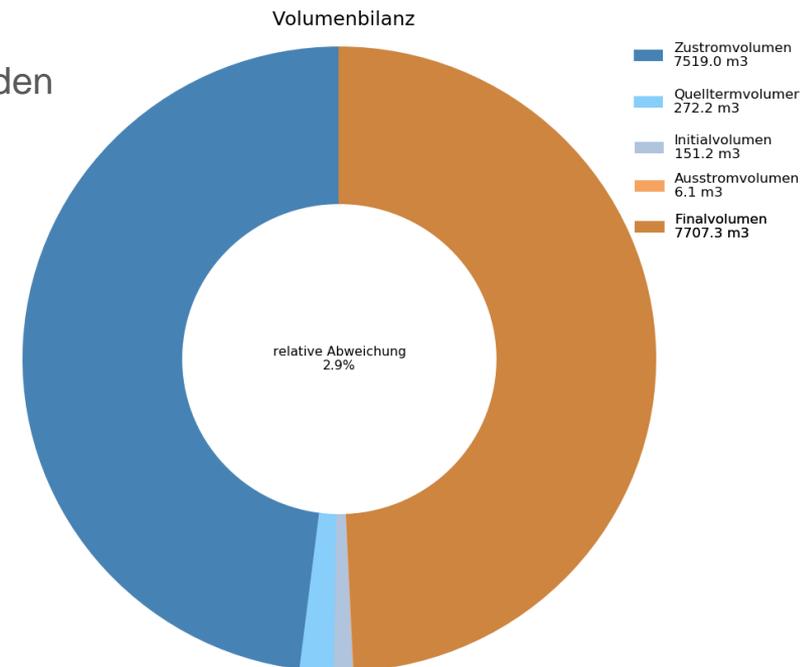
- ▶ 8 Polder entlang der Deutschen Donau
- ▶ Deiche mit „Sollbruchstellen“
- ▶ Beim Erreichen eines Wasserstandes eines Aktivierungspegels wird Dammmaterial kontrolliert weggeschwemmt
- ▶ Auslauf im unteren Polderbereich – Aktivierung wenn Binnenpegel ~ Gewässerpegel
- ▶ HYDRO_AS-2D-Modell mit 8 Poldern
- ▶ Früher: Pegel.dat „beobachten“, Modell unterbrechen, Datensatz verändern...
- ▶ Scripting: 1 Rechenlauf
- ▶ Script im Rahmen einer Schulung mit Kunden erstellt
- ▶ Modellierung: Z-Werte Deichknoten absenken
- ▶ „Reproduzierbarkeit/Prüfbarkeit“



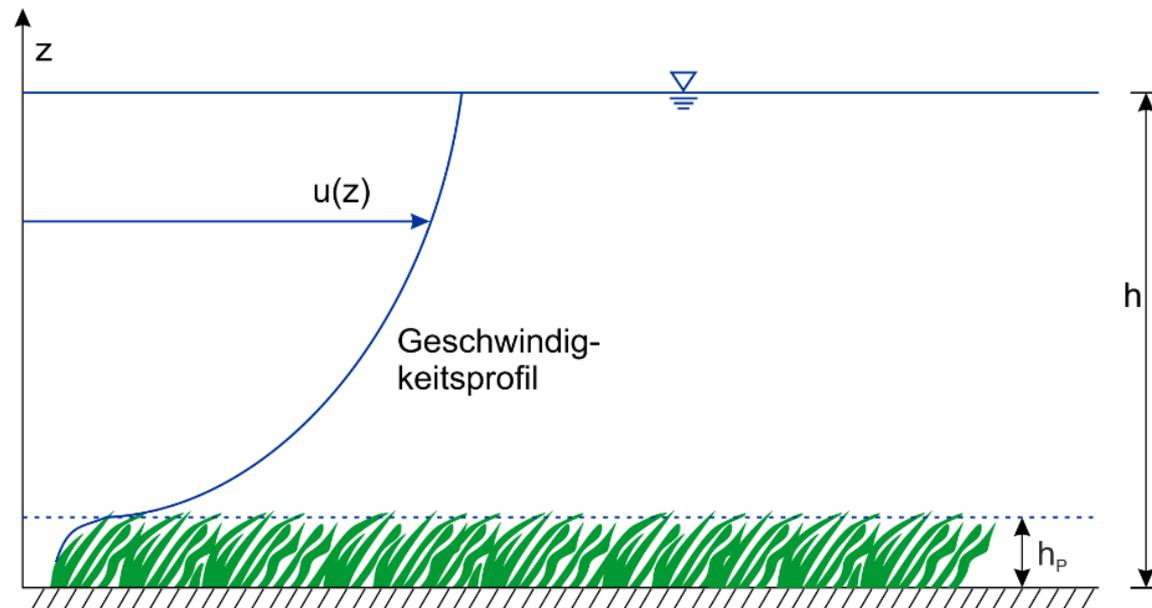
- Stationäre Wasserstände für verschiedene Zuflüsse berechnen
- Q-H-Beziehung → Pegelkurve
- Ähnlich: Berechne HQ_1 , HQ_2 , HQ_5 , HQ_{10} , etc.
- Optionen ohne Scripting:
 - Viel Simulationszeit einplanen
 - Manuell pegel.dat beobachten
- Mit Scripting:
 - Stationären Zustand im Script detektieren
 - Mit Scripting Q-Zufluss variieren
 - Ergebnisse zum stationären Zustand in eigene Datei
- Was ist stationär?
 - $Q_{\text{Zufluss}} = Q_{\text{Abfluss}}$
 - Keine Änderung in Q_{Abfluss} einer Stunde



- ▶ Starkregen: Wasserbilanz
 - ▶ Zuflüsse
 - ▶ Abflüsse
 - ▶ Niederschlag
 - ▶ Versickerung
 - ▶ Änderung im Modell
- ▶ Versickerung mit „Feflow-in“
 - ▶ Wird nur versickert, wenn genügend Wasser vorhanden
- ▶ Script
 - ▶ Ermittelt für jede Sekunde die Wasserbilanzkomponenten
 - ▶ Bilanziert/Addiert
 - ▶ Ausgabe für Grafikaufbereitung (Python)
- ▶ Integration in HYDRO_AS-2D „coming soon“
- ▶ Scripting meist kein Performanceproblem
Hier: Häufig alle Knoten abfragen 🤖



- ▶ Bewuchs: Fließwiderstand abhängig von
 - ▶ Wasserstand
 - ▶ Fließgeschwindigkeit
- ▶ Pflanzen werden bei hohem Wasserstand und Fließgeschwindigkeit „gelegt“
→ Fließwiderstand nimmt ab
- ▶ Abbildung über Änderung der Rauheitsbeiwerte
- ▶ $k_{st} = f(h, v)$
- ▶ Wissenschaftlicher Anwendungsbereich
- ▶ Masterarbeit Michel Heidmann
- ▶ Vorstufe zur Übernahme in HYDRO_AS-2D



- Schulungen
- Auch: Erfolgreiche Autodidakten 
- Grundlegende Programmierkenntnisse hilfreich
- Gut dokumentiert

- Nutzung von fertigen Scripten ohne Zusatzkenntnisse möglich

- ▶ HYDRO_AS 5.2: Die meisten Daten sind nun „scriptable“
- ▶ Scripte sind Teil des Modells
- ▶ Scripting: Objektiviert, Reproduzierbarkeit, Übertragbarkeit
- ▶ „Ad-Hoc-Scripte“ ✓
 - ▶ Script oft auf konkretes Modell zugeschnitten
- ▶ Softwaretechnisch ausgereifte Software ✓
 - ▶ Trennung von Programm und Parametern
 - ▶ Wiederverwendbarkeit für andere Modelle
- ▶ Scripte testen!
 - ▶ Schnell rechnende Modelle aufbauen
 - ▶ Für komplexe Scripte: Grob- oder Teilmodell entwickeln
- ▶ Wann „muss“ man scripten?
 - ▶ Modelldaten werden in Abhängigkeit von Modellzuständen verändert
- ▶ Wann „kann“ man scripten?
 - ▶ Preprocessing/Postprocessing
- ▶ Modellkopplung noch ausbauen

Einsatz von Scripting und die Modellierung von Steuerungen

Dipl.-Math Benedikt Rothe

Danke für Ihre Aufmerksamkeit ...

