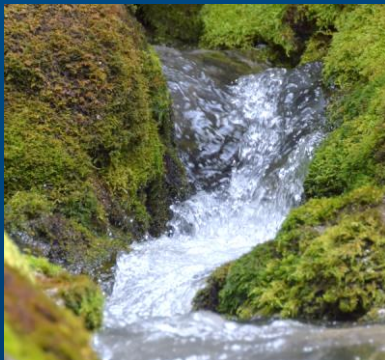


Vortrag auf dem
e-Teaching Day
der TU Dresden 2018

Bereich Bau und Umwelt, Fakultät Umweltwissenschaften
Fachrichtung Hydrowissenschaften

Verstehen auf Knopfdruck: Web-basierte Simulationsmodelle in der Lehre

Thomas Petzoldt¹, Markus Ahnert², Nora Franzke³ und David Kneis¹



¹ Institut für Hydrobiologie

² Institut für Siedlungs- und Industrierewasserwirtschaft

³ Masterstudiengang Hydrobiologie

Simulationsmodelle in der Lehre



Viele Lehrinhalte in MINT-Fächern ...

- beruhen auf physikalischen, chemischen oder biologischen Prozessen
- können mit mathematischen Modellen beschrieben ...
- ... und am Computer simuliert werden

$$\frac{dN}{dt} = r \cdot N \cdot \left(1 - \frac{N}{K}\right)$$
$$r = \frac{r_{max} \cdot S}{k_S + S}$$

Prozesse und Zusammenhänge in einem geschützten Rahmen erkunden

- Eigene Erfahrungen sammeln
- Eingangsgrößen und Modellparameter manipulieren
- Auswirkungen von Entscheidungen werden digital visualisiert
- Hohe Anzahl von Varianten möglich
- Kein Risiko für Mensch und Umwelt



Nach: Medienzentrum TU Dresden (2016): Digitales Lernen und Lehren in der Hochschule, Heft 1

Aber: Es gibt auch Hürden

Inhaltliche Hürden

Angst vor Formeln

Komplexität der Aufgabe

Numerische Lösungsverfahren

Technische Hürden

Installation, Portabilität

Bedienphilosophie

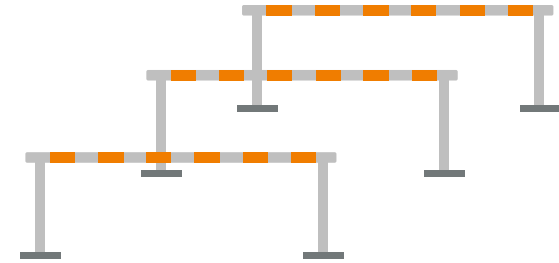
Sprachen (englisch, deutsch)

Lizenzfragen

Interpretation der Ergebnisse

Professor: **Gestern zu Hause ging es noch.**

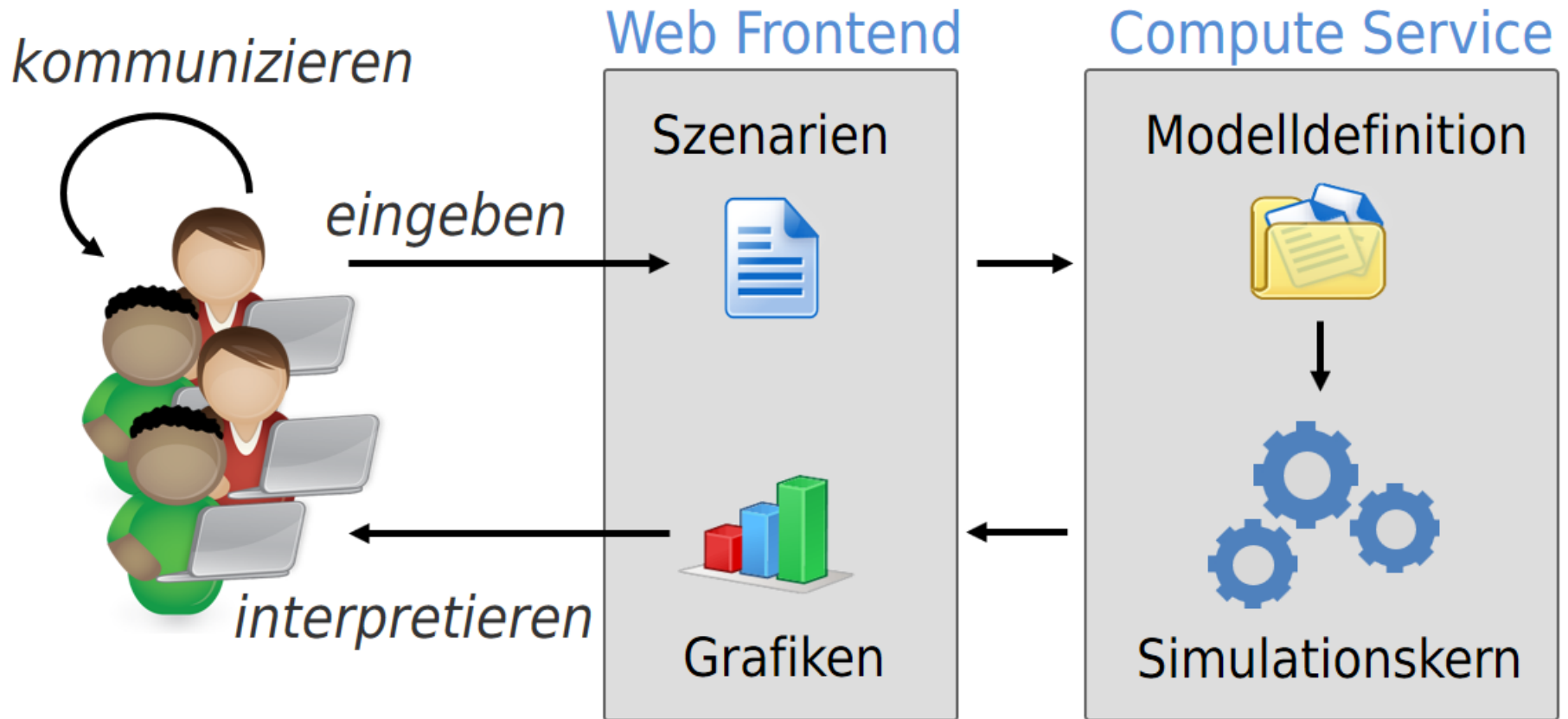
Student: **Wie soll ich das allein schaffen.**



Beispiele

- Installieren Sie bitte die xyz Software
 - Läuft das auch auf dem Mac?
- Zunächst müssen wir ein Projekt anlegen
- Dann erstellen Sie eine Textdatei ...
- Bitte auf Dezimalpunkt achten!
- Nur mit Excel 2010 32bit kompatibel, ...
- Bei Excel 2016 Makros aktivieren ...
- Das geht nur in der PRO-Version ...
- Was ist eine PATH-Variable...?

Unsere Idee



Vorteile Interaktiver Web-basierter Modelle

- Keine Installation erforderlich
- Modelle lassen sich an die Lehre anpassen
- Windows, Linux, Mac, Android, iOS
- Kleine Demos hohe Komplexität
- Direkte Verlinkung mit OPAL:
Dokumentation, Aufgaben, Feedback
- Output-Download für kreative
eigene Ideen der Studierenden
- Zielgruppen-Anpassung

(Bachelor, Master, Postgradualstudium /
deutsch, englisch, polnisch, ukrainisch, ...)

Logistic Growth

$$\frac{\Delta N}{\Delta t} = r_{max} \cdot N \cdot (1 - N/K)$$

N

time

N(t-1)

N(t)

N0 (initial value): 0,1

K (carrying capacity): 1

r_max (growth rate): 2.75

Integrator: euler

Note:

- **euler** : discrete simulation of the difference equation with $\Delta t = 1$,
- **lsoda** : continuous simulation with $\Delta t \rightarrow 0$ (differential equation)

Compute + Web → R + Shiny



R: Programmiersprache für statistische Analysen und Grafiken

- *lingua franca* of computational statistics
- OpenSource, 12.000 Zusatzpakete (-- zwölf tausend --)
- Statistik, data mining, machine learning, Finanzen, ..., Simulation
- Auch in den Bio- und Umweltwissenschaften sehr verbreitet
- Vergleichbar mit anderen Matrix-Sprachen

Shiny: R-Paket für **interaktive** Webseiten (mit JavaScript)

- Modernes Programmierparadigma
- Einfach zu lernen

Wir nutzen **R** in Forschung und Lehre als Standardwerkzeug.
Grundprinzip auch in anderem Umfeld möglich, z.B. mit Python.

Anwendungsbeispiele

1. Webseiten mit online-Rechenfunktion
2. Visualisierung von Funktionen und Algorithmen
3. Dynamische Simulation: Beispiel Räuber-Beute-Modell
4. Komplexe dynamische Modelle: Beispiel ökologisches Seenmodell
 - Technologie
 - Aufbau der Web-App
 - Einbettung in Lehrveranstaltungen
5. Praktische Erfahrungen
6. Ausblick: WebLab: eine virtuelle Kläranlage im Hörsaal

Online-Rechner

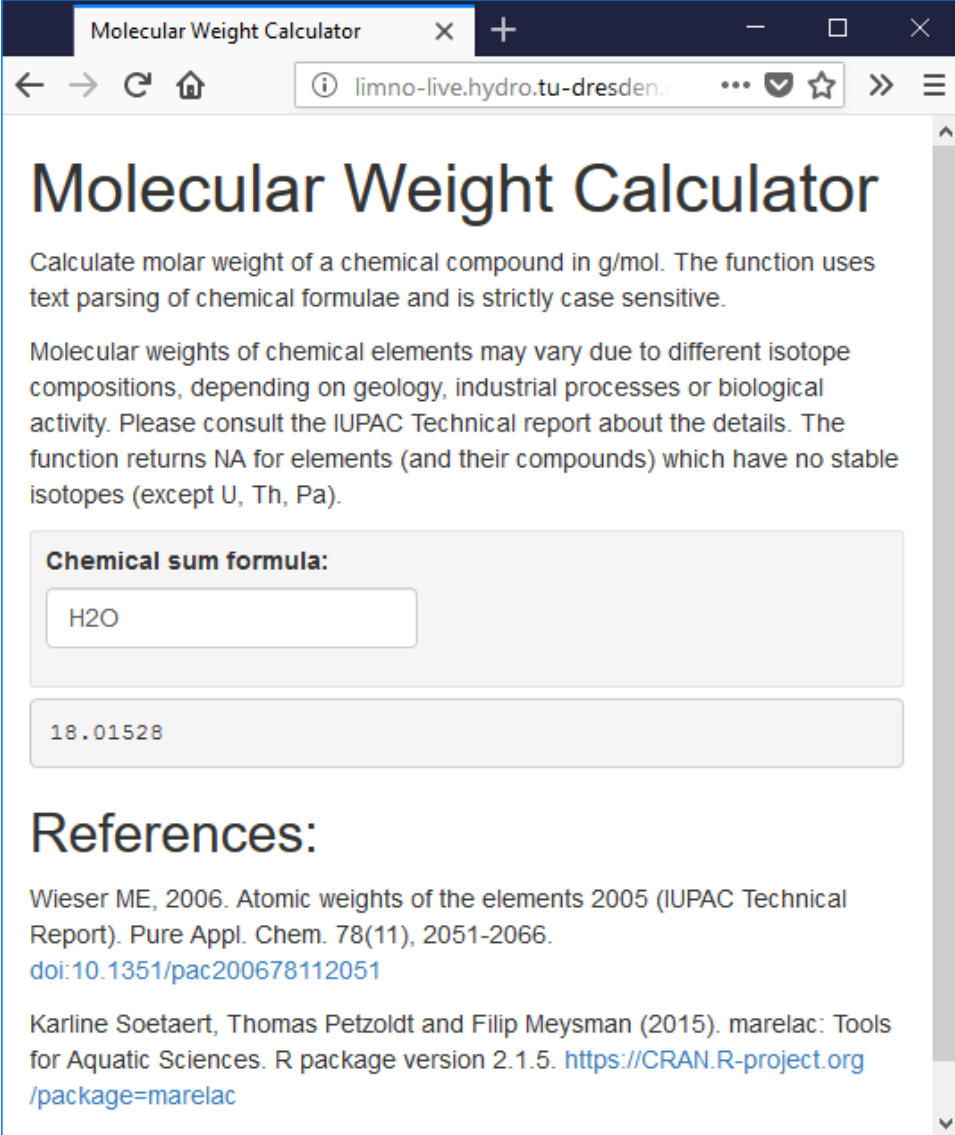
Dahinter steckt eine Datentabelle und ein Parser, der eine chemische Summenformel in einen mathematischen Ausdruck umwandelt.

Didaktische Idee:

Rechner bietet Mehrwert und wirbt für weitere Informationen

- Literatur zum Thema „Molmasse“
- R-Paket mit offline-version des Calculators und weiteren fachspezifischen Tools

Prinzip funktioniert natürlich auch mit mehr als einem Eingabefeld.



Molecular Weight Calculator

Calculate molar weight of a chemical compound in g/mol. The function uses text parsing of chemical formulae and is strictly case sensitive.

Molecular weights of chemical elements may vary due to different isotope compositions, depending on geology, industrial processes or biological activity. Please consult the IUPAC Technical report about the details. The function returns NA for elements (and their compounds) which have no stable isotopes (except U, Th, Pa).

Chemical sum formula:

H₂O

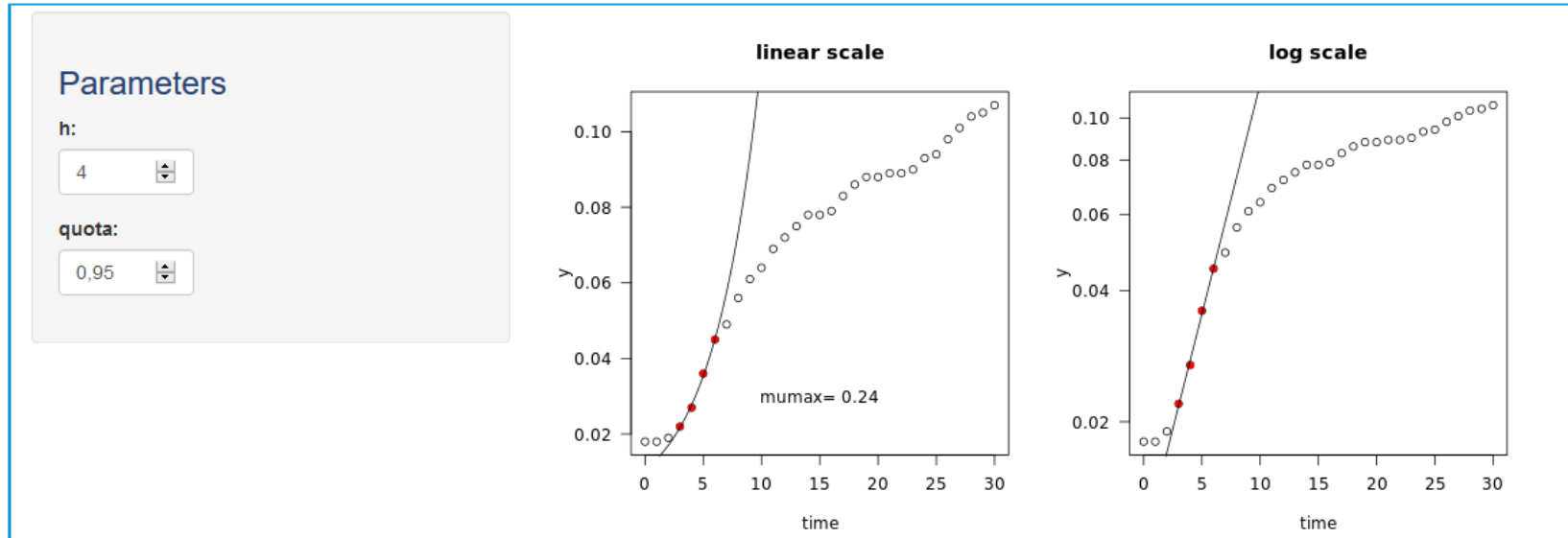
18.01528

References:

Wieser ME, 2006. Atomic weights of the elements 2005 (IUPAC Technical Report). Pure Appl. Chem. 78(11), 2051-2066.
doi:10.1351/pac200678112051

Karline Soetaert, Thomas Petzoldt and Filip Meysman (2015). marelac: Tools for Aquatic Sciences. R package version 2.1.5. <https://CRAN.R-project.org/package=marelac>

Visualisierung einer Funktion oder Algorithmus



Nutzer ändert Parameter, Ergebnisse werden sofort graphisch veranschaulicht.

App beinhaltet eigentlich auch Erläuterung und Literatur

→ abschaltbar für Einbettung in andere „normale“ Webseiten

• Dargestellt ist die einbettbare Version:

<http://weblab.hydro.tu-dresden.de/apps/grow-easy/?embed>

Räuber-Beute-Modell

Dynamisches Modell
(Simulation über die Zeit)

Drei Differentialgleichungen
schnelle numerische Lösung

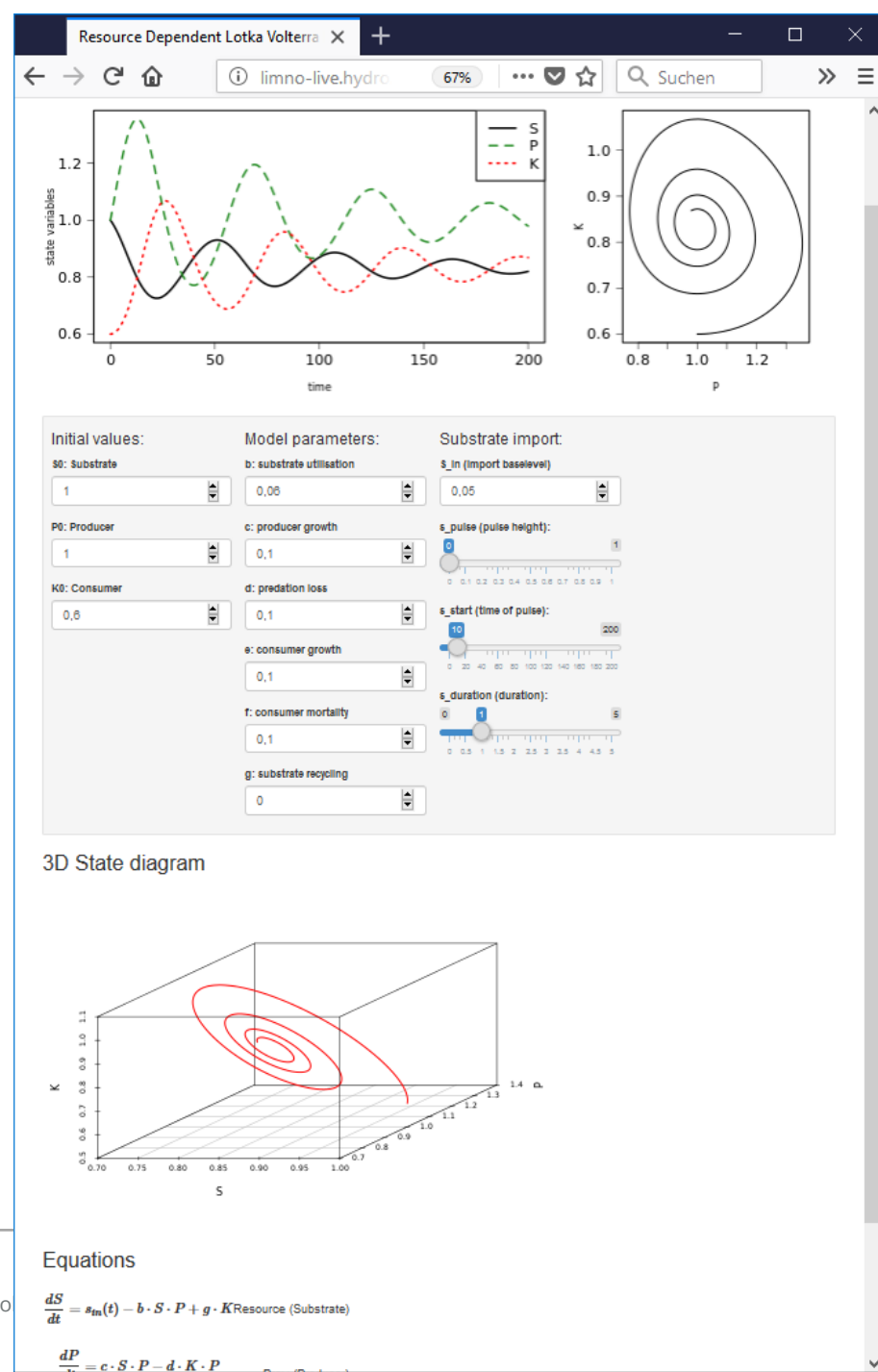
Einsatz in Übungen

- Master Hydrobiologie
 - Postgradualkurs Ökotoxikologie
- Aufgabenstellung in OPAL

Nutzbar auch als Demo in Vorlesungen,
und durch Schulen.

Resource Dependent Lotka Volterra X

limno-live.hydro 67% Suchen



state variables

time

Initial values:

S0: Substrate: 1

P0: Producer: 1

K0: Consumer: 0,6

Model parameters:

b: substrate utilisation: 0,06

c: producer growth: 0,1

d: predation loss: 0,1

e: consumer growth: 0,1

f: consumer mortality: 0,1

g: substrate recycling: 0

Substrate import:

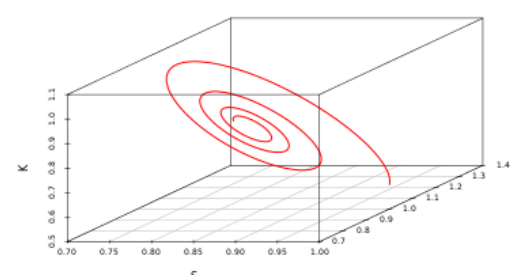
S_in (import baselevel): 0,05

S_pulse (pulse height): 1

S_start (time of pulse): 10

S_duration (duration): 5

3D State diagram



Equations

$$\frac{dS}{dt} = s_{in}(t) - b \cdot S \cdot P + g \cdot K - \text{Resource (Substrate)}$$
$$\frac{dP}{dt} = c \cdot S \cdot P - d \cdot K \cdot P$$

Kleine Web-Apps mit R ...



Sind einfach und schnell „ad-hoc“ implementierbar

- Oft nur eine Seite R-Code; mit Text und Formeln kombinierbar (markdown)
- Aufwand: ab ca. 1 Stunde
- Lauffähig auf Server oder lokalem PC
- In Kursen durch Studierende erstellbar (→ Spaßfaktor +++)

Doch wie steht es um komplexere Anwendungen?

Ist R schnell genug?

→ Rechenzeit!

Wer soll das programmieren?

→ Implementationsaufwand?

Beispiele:

- 1) Modell für die Wassergüte von nährstoffbelasteten Seen
- 2) Modell für Kläranlagen

Seenmodell

Teilergebnis aus einem Forschungsprojekt

1. Wissenschaftliche Fragestellung
2. Daten ^a
3. Modell ^b
4. Ergebnisse



Veranschaulichung für Projektpartner
Öffentlichkeitsarbeit (Projektabschluss)
Nachnutzung in der Lehre

- a) BTU Cottbus
- b) TU Dresden

Und viele weitere Projektpartner!



Stickstofflimitation in Binnengewässern

Ist Stickstoffreduktion
ökologisch sinnvoll
und
wirtschaftlich vertretbar?

Abschlussbericht
NITROLIMIT II

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



FONA
Nachhaltiges
Wassermanagement
BMBF

www.nitrolimit.de

Unsere Technologie: Modular und 100% OpenSource

Herausforderungen



Implementationsaufwand

- Skriptsprache
- Generische Tabellenrepräsentation



Maschinelle Code-generierung



Performance

- Effiziente numerische Lösung der Modelle in C bzw. Fortran-Code



Verfügbarkeit

- Cloud-Service des ZIH

Werkzeuge



Programmiersprache

R-Paket [shiny](#) + Shiny-Webserver-Software

[rodeo](#):

R-Paket für generische Modellbeschreibung

- Tabellenbasiert (Peterson-Matrix)
- Codegenerator

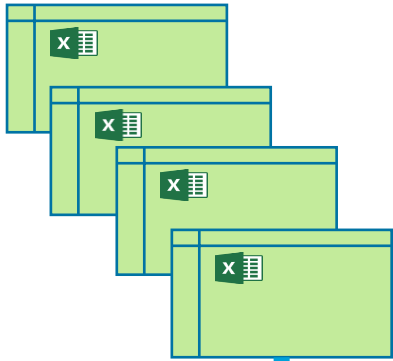
[deSolve](#):

R-Numerikpaket für Differentialgleichungen

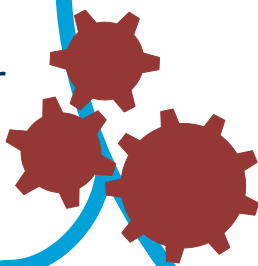
GNU Compiler: [gcc](#), [gfortran](#)

Entwickler

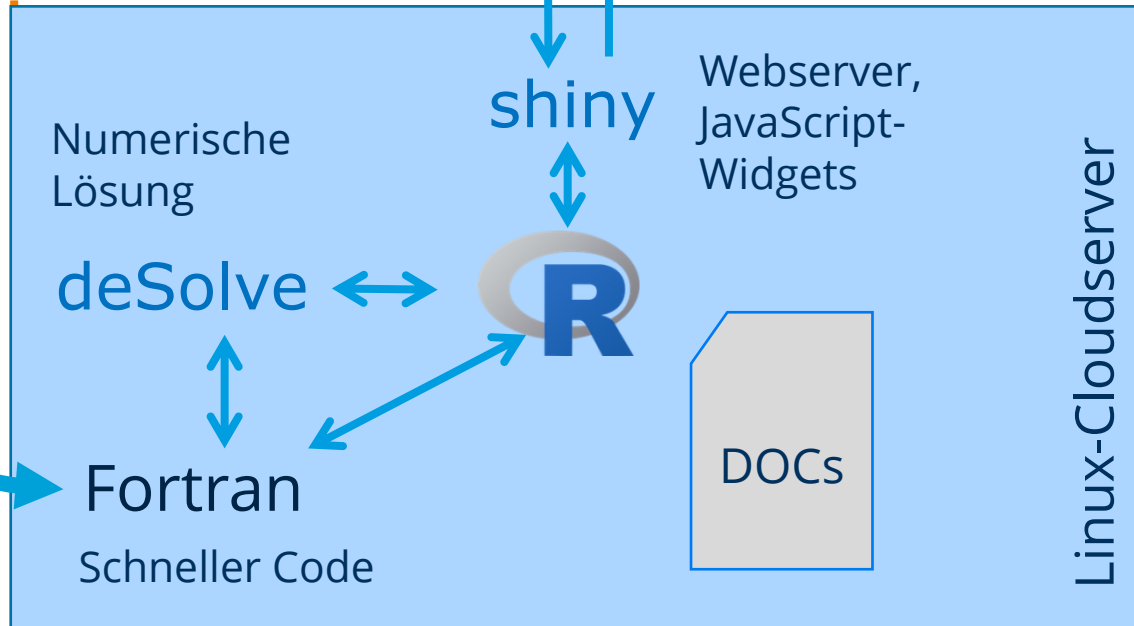
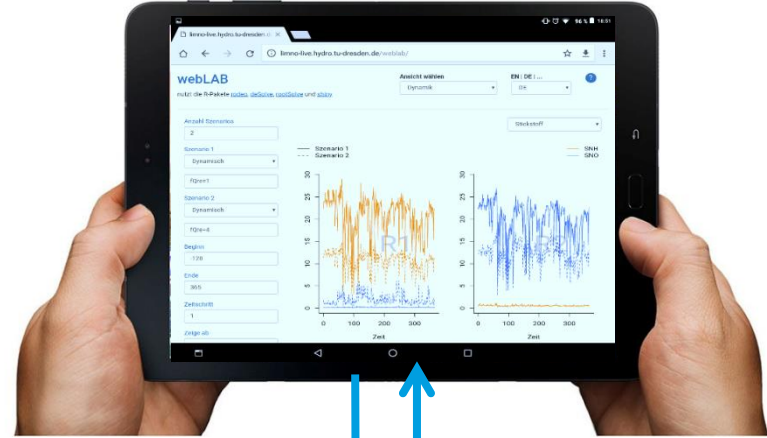
Generische
Modellbeschreibung
(Tabellen)



Code-
Generator
rodeo



Anwender



Aufbau Seenmodell-App

Webseite mit mehreren Tabs

1. Wissenschaftliche Fragestellung
2. Modellübersicht
(Schema, Zustandsgrößen, Prozesse)
3. Hauptfenster Simulation
4. Literaturverzeichnis
→ Links auf Originalquellen
5. „Experten-Menü“
 - Download der Simulationsergebnisse
(direkt nach Excel de/en)
 - Änderung weiterer Modellparameter

Shallow Lake Simulation

1 2 3 4 5

Shallow Lake Simulation Introduction The Model Simulation Acknowledgments + Links

Effect of Nitrogen Reduction in Shallow Lakes

Most inland waters in Germany did not meet the good ecological status requirements stipulated in the EU Water Framework Directive for 2015. It was previously assumed that phosphorus is the primary determinant of water quality. In the BMBF funded project **NITROLIMIT** (www.nitrolimit.de) it has been proved that nitrogen is also a crucial control variable for many surface waters and its reduction ecologically meaningful. This has led to demands for procedures to ensure the reduction of nitrogen input.


Scientists from different disciplines and seven scientific institutions worked together on interrelated research modules to clarify whether nitrogen reduction is ecologically meaningful and economically feasible. Modelling was part of research module 2 that aimed to fill knowledge gaps about N- and P-turnover processes.

Scientific Questions

The "toy-model" found at these pages aims to demonstrate some of the main scientific questions:

1. Which nitrogen sources fuel the development of phytoplankton in summer:
 - external nitrogen load entering the lake by its inflow?
 - atmospheric N_2 , fixed by specialized blue-green algae (N_2 fixers)?
 - remineralized nutrients, originating from the organic matter at the sediment surface, especially from the dying algae of the spring bloom?
2. Will it be likely, that N_2 fixing cyanobacteria develop?
3. How is nitrogen and phosphorus turnover affected?

Please note: this model is an extremely simplified representation of the complex processes, and especially their dynamic predictions.



Shallow Lake Simulation

5

Figure scaling

| Phytoplankton | Zooplankton | Phosphorus |
|---------------|-------------|------------|
| 6 | 0,3 | 0,15 |

Nitrogen: 1,5

Expert options

Parameters

Show model parameters

Upload parameter file ...

Browse... No file selected

Output data

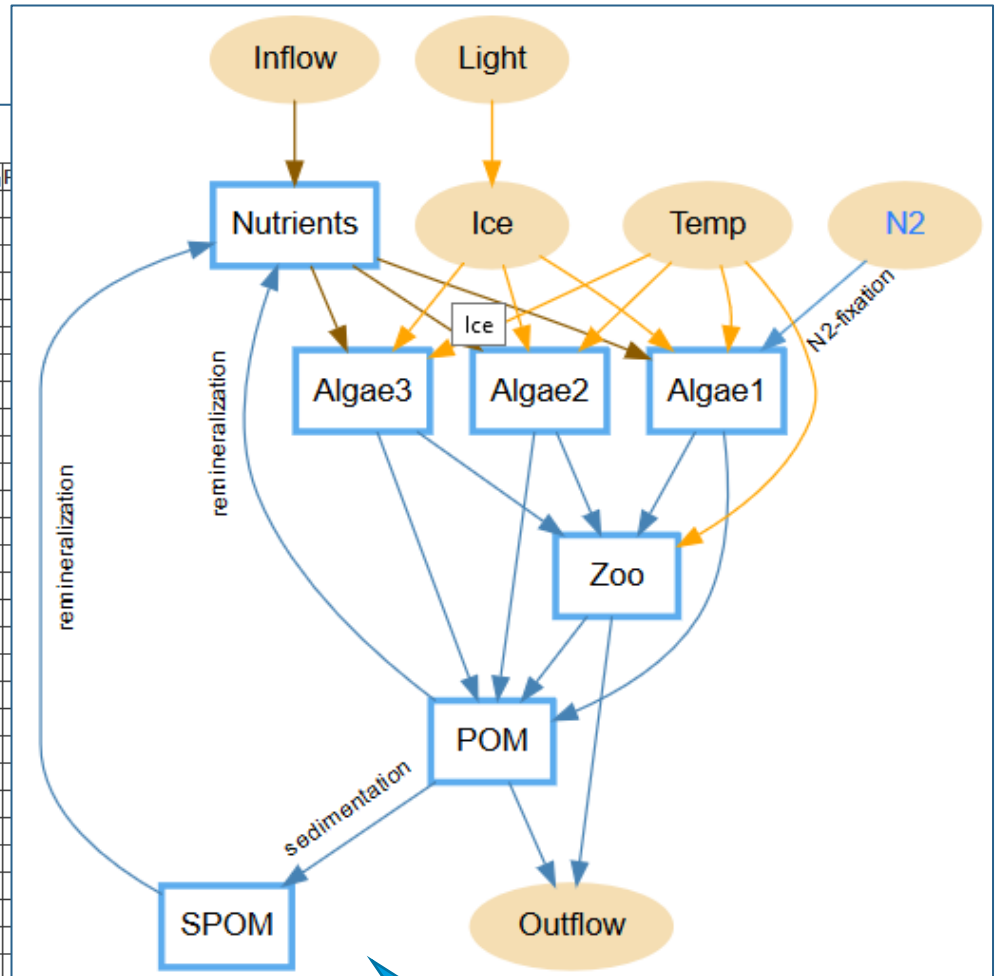
Show numerical output Show POM fractions German decimal comma

Download data

Modellstruktur

Stoichiometry Matrix

| Process | HPO ₄ ²⁻ | ALG_F1 | ALG_F2 | ALG_F3 | ZOO | O ₂ | NH ₄ ⁺ | NO ₃ ⁻ | POMD | ALG_F1 | ALG_F2 | ALG_F3 |
|-------------------|--------------------------------|--------|--------|--------|-----|----------------|------------------------------|------------------------------|------|--------|--------|--------|
| g_ALG_F1_NH4 | ▽ | △ | | | | | △ | ▽ | | | | |
| g_ALG_F2_NH4 | ▽ | | △ | | | | △ | ▽ | | | | |
| g_ALG_F3_NH4 | ▽ | | | △ | | | △ | ▽ | | | | |
| g_ALG_F1_NO3 | ▽ | △ | | | | | △ | ▽ | | | | |
| g_ALG_F2_NO3 | ▽ | | △ | | | | △ | ▽ | | | | |
| g_ALG_F3_NO3 | ▽ | | | △ | | | △ | ▽ | | | | |
| g_ALG_F1_N2 | ▽ | △ | | | | | △ | ▽ | | | | |
| r_ALG_F1 | △ | ▽ | | | | | ▽ | △ | | | | |
| r_ALG_F2 | △ | | ▽ | | | | ▽ | △ | | | | |
| r_ALG_F3 | △ | | | ▽ | | | ▽ | △ | | | | |
| d_ALG_F1 | | ▽ | | | | | | | | △ | | |
| d_ALG_F2 | | | ▽ | | | | | | | | △ | |
| d_ALG_F3 | | | | ▽ | | | | | | | | △ |
| g_ZOO_ALG_F1 | | ▽ | | | | △ | | | | | △ | |
| g_ZOO_ALG_F2 | | | ▽ | | | △ | | | | | | △ |
| g_ZOO_ALG_F3 | | | | ▽ | | △ | | | | | | △ |
| r_ZOO | △ | | | | ▽ | ▽ | △ | | | | | |
| d_ZOO | | | | | ▽ | | | | | | | |
| nitri | | | | | | ▽ | ▽ | △ | | | | |
| miner_POMD_ALG_F1 | △ | | | | | | △ | | | ▽ | | |
| miner_POMD_ALG_F2 | △ | | | | | | △ | | | | ▽ | |
| miner_POMD_ALG_F3 | △ | | | | | | △ | | | | | ▽ |
| miner_SPOMD | △ | | | | | | △ | | | | | |
| sed_POMD_ALG_F1 | | | | | | | | | | ▽ | | |

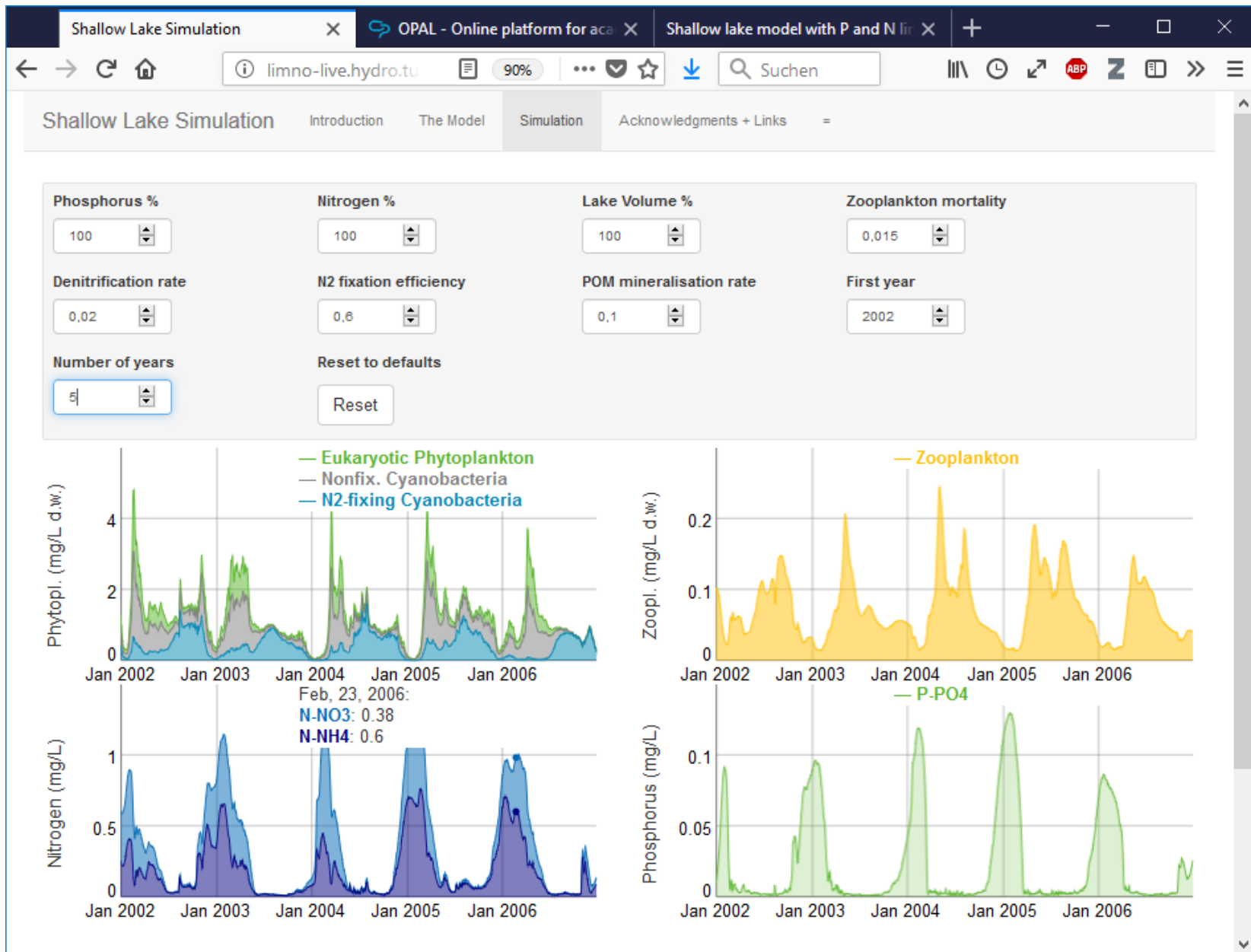


Stöchiometriematrix

Zustandsgrößen
Maßeinheiten
Prozesse

Modellschema
(vereinfacht)

Simulationsfenster: Steuerung und Visualisierung



Live-Demo

<https://weblab.hydro.tu-dresden.de/models/shallowlake/>

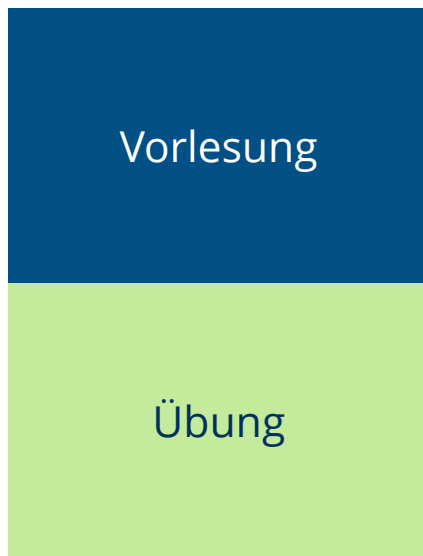


Simulationen im „Flipped Classroom“

Kriterien

1. Simulation nach Aufgabenstellung
2. Literaturrecherche
3. Eigene Kreativität

Reflexion




Eigenständige Erarbeitung

- 1 Woche Zeit
- Gruppengröße 2-3
- Optional: Konsultation, Email



Gruppengröße bis 25

Vorstellung durch Studierende



TECHNISCHE UNIVERSITÄT DRESDEN

Stickstoffdynamik eines eutrophen Flachsees

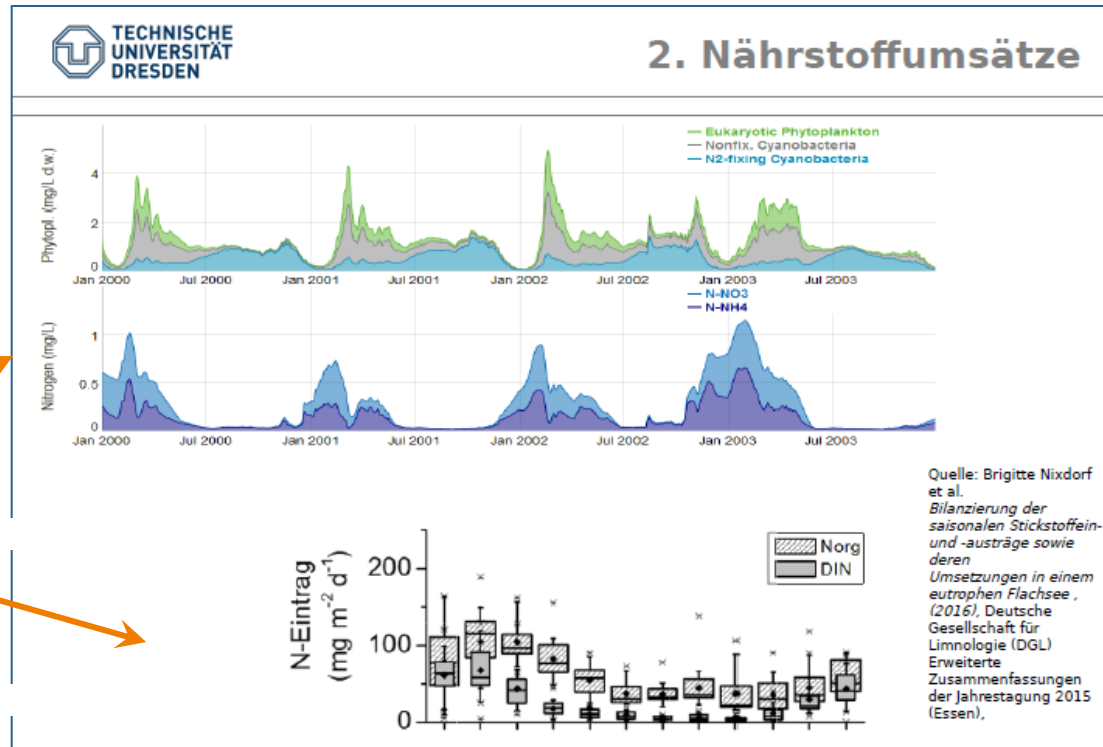
Betrachtung anhand der „shallow lake simulation“

Susann Bromberger
Nora Franzke

Dresden, 31.01.2018

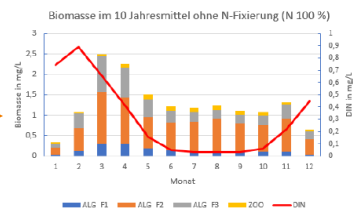


Diskussion der Simulationen im Vergleich zur Literatur



- Grafiken aus App + Literatur

+ Weitere eigene Grafiken und Tabellen aus den Simulationsdaten



| | ALG_F1 in mg/ L | ALG_F2 in mg/ L | ALG_F3 in mg/ L | ZOO in mg/ L | Gesamtbiomasse in mg/ L |
|---------|--------------------|--------------------|--------------------|-----------------|----------------------------|
| N 100 % | 5,5 (35 %) | 6 (38 %) | 3,5 (22 %) | 0,9 (6 %) | 15,9 |
| N 50 % | 7,7 (48 %) | 4,2 (26 %) | 2,6 (16 %) | 0,8 (5 %) | 15,3 |
| N 30 % | 8,9 (56 %) | 2,7 (17 %) | 1,6 (10 %) | 0,7 (4 %) | 13,9 |

Erfahrungen: Doppelte Feedback-Schleife

Studierende

Wie schaffen wir das?

Wir schaffen das!

Wird mein Vortrag von der Gruppe honoriert?

Welche Botschaft ziehe ich aus dem Stoff?



- ++ Aktives Lernen
- ++ Individualisierung
- ++ Flexibilisierung Selbststudium
- /+ Kurs wird komplexer („out of order“)
- + Soziales Lernen in (realer) Gruppe

Lehrende

Wurde der Stoff verstanden?

Funktioniert die Simulation?

Ist die Aufgabenstellung motivierend?

Anreize für Literaturstudium ausreichend?



- ++ hoher Lerneffekt für den Lehrenden
- + direktere Interaktion mit Studierenden
- +/- Es klappt nicht immer
- +/- Struktur des Kurses einhalten?
- ? Skalierbarkeit auf größere Gruppen

Ausblick: WebLab – eine Kläranlage im Hörsaal

Ahnert, Kneis und Petzoldt
Projekt im Multimediafonds
der TU Dresden

Modell ASM3 für Abwasserbehandlungsanlagen

- Kennenlernen der ablaufenden Prozesse
- Simulation von steuernden Eingriffen
- schrittweise Erhöhung der Komplexität

Technologie:

- R/Shiny Compute/Webserver
- Differentialgleichungssystem
- Codegenerator Tabellen → Fortran
- 39 Zustandsgrößen, 124 Prozesse

webLAB*
Eine virtuelle Kläranlage im Hörsaal

Markus Ahnert // Institut für Siedlungs- und Industrieressourcenwirtschaft markus.ahnert@tu-dresden.de
David Kneis, Thomas Petzoldt // Institut für Hydrobiologie david.kneis@tu-dresden.de, thomas.petzoldt@tu-dresden.de

Unser Ausgangspunkt
Simulationen können helfen, komplexe naturwissenschaftliche und ingenieurtechnische Systeme tiefer zu verstehen, aber die verwendeten Modelle:
– sind oft sehr komplex (Formeln!),
– erfordern Spezialsoftware mit teils verschiedenen Bedienkonzepten,
– benötigen Einarbeitungszeit,
– haben oft Akzeptanzprobleme.

Studierende:
Wie soll ich das allein hinkriegen?
Lehrende:
Gestern hat es noch geklappt ...

Die Idee
Eine Web-App und ein Compute-Service ermöglichen eine intuitive und interaktive Bedienung:
– spielerischer Einstieg,
– keine Softwareinstallation durch die Nutzer, ein Web-Browser genügt,
– plattformübergreifend und geräteunabhängig (Notebook, Tablet),
– Modellgleichungen sollen einsehbar sein aber nicht abschrecken.

Beispiel: Das Modell ASM3 für Abwasserbehandlungsanlagen kann so bereits in der Grundlagenausbildung eingesetzt werden:
– Kennenlernen der ablaufenden Prozesse ohne Einarbeitung
– schrittweise Erhöhung der Komplexität des vermittelten Wissens
– Simulation von steuernden Eingriffen auf Kläranlagen

Unsere Umsetzung
Die Simulationen laufen auf einer virtuellen Maschine in der Enterprise Cloud der TU Dresden, webweiter Zugriff ist möglich.
Die Programmierung erfolgte in der Skriptsprache www.r-project.org/
Wir nutzen vorhandene bzw. selbst entwickelte R-Pakete für die Web-Interface (shiny), zur numerischen Lösung von Differentialgleichungen (deSolve) und zur automatischen Generierung von schönem Fortran-Code aus einer tabellarischen Modelldefinition (rosol).

Die Software ist so konzipiert, dass Modelle anderer Fachgebiete (z.B. aus Biologie, Chemie, Medizin, Wirtschaftswissenschaften) mit überschaubarem Aufwand implementiert werden können.

Vorteile und zukünftige Nutzung
Die entwickelte Plattform bietet:
– einen einfachen pädagogischen Zugang zu wissenschaftlichen Zusammenhängen,
– Kombinationsmöglichkeiten mit webbasierten Kursen (z.B. OPL),
– eine Ergänzung zur Präsenzaufnahme und Anwendung in externen Weiterbildungsveranstaltungen,
– instituts- und universitätsübergreifender Einsatz,
– vollständige Open Source - Umsetzung (GPL v. 2 bzw. 3),
– mehrsprachige Nutzeroberfläche.

Aktuell wird das System für einen Einsatz in gemeinsamen Weiterbildungen mit Projektpartnern in der Ukraine vorbereitet.

Aktueller Prototyp auf: <https://weblab.hydro.tu-dresden.de/weblab/>

Mitglied im Netzwerk von: **weblab***
DRESDEN concept Web-basiertes Lehrmodell am Beispiel der Abwasserbehandlung
gefördert aus dem Multimediafonds der TU Dresden

<https://weblab.hydro.tu-dresden.de/weblab/>



Fazit



WebLab funktioniert!

- Für Übungen kleinerer Gruppen in Master und Postgradualkursen.
- Als Demo in der Vorlesung, auch bei größeren Gruppen.



Herausforderungen für Phase 2 des Projekts

- Schärfung des didaktischen Konzepts
- Umfrage bei den Studierenden
- Skalierung der technischen Ressourcen



Interesse geweckt?

- Eine Einstiegs-App braucht nur ein paar Zeilen 😊



Appendix

Brusselator-Modell als Web-App

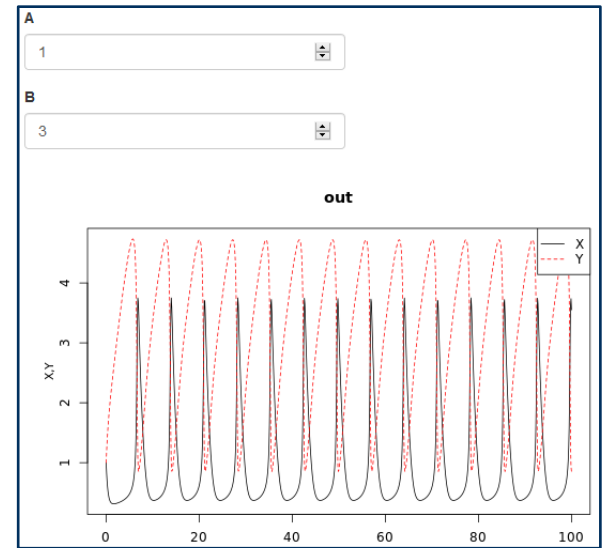
```
library("deSolve")

brusselator <- function(t, y, p) {
  with(as.list(c(y, p)), {
    dX <- k1*A - k2*B*X + k3*X^2*Y - k4*X
    dY <- k2*B*X - k3*X^2*Y
    list(c(X=dX, Y=dY))
  })
}

server <- function(input, output) {
  output$brussels <- renderPlot({
    parms <- c(A=input$A, B=input$B, k1=1, k2=1, k3=1, k4=1)
    out <- ode(y = c(X=1, Y=1), times=seq(0, 100, .1), brusselator, parms)
    matplot.0D(out)
  })
}

ui <- fluidPage(
  numericInput("A", label = "A", value = 1),
  numericInput("B", label = "B", value = 3),
  plotOutput("brussels")
)

shinyApp(ui=ui, server=server)
```



Literatur

- Ahnert M., Alex J., Dürrenmatt D.J., Langergraber G., Hobus I., Schmuck S., Spring V. (2015) Dynamische Simulation als Bestandteil einer Kläranlagenbemessung nach DWA-A 131. KA Korrespondenz Abwasser, Abfall 62 (7) 615-624.
- Kneis, D., Petzoldt, T., & Berendonk, T. U. (2017) An R-package to boost fitness and life expectancy of environmental models. Environmental Modelling & Software, 96, 123-127. <https://doi.org/10.1016/j.envsoft.2017.06.036>
- Medienzentrum der TU Dresden (2017) Themenreihe Digitales Lehren und Lernen in der Hochschule – digitale Simulationen und Planspiele. http://elearning-tu-dresden.blogspot.de/2017/08/themenreihe-digitales-lehren-und-lernen_24.html
- R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org>
- Reichert, P., Mieleitner, J. and Schuwirth, N. (2016) Modelling Aquatic Ecosystems. Course 701-0426-00 ETH Zürich. <http://www.eawag.ch/en/departement/siam/teaching/modelling-aquatic-ecosystems/>
- RStudio Inc. (2013) Easy web applications in R. <http://www.r-studio.com/shiny>
- Riedel, J., Berthold, S., Dubrau, M., Möbius, K. (2018) Digitales Lehren & Lernen in der Hochschule. Heft 1: Flexibilität und Vielseitigkeit mit digitalen Lehr- und Lernmaterialien erhöhen. Köhler, T. (Hrsg.). 2. überarb. aktual. Aufl. <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-217402>
- Soetaert, K. Petzoldt, T. Setzer, R.W. (2010) Solving differential equations in R: Package deSolve. Journal of Statistical Software, 33(9), 1--25. <https://doi.org/10.18637/jss.v033.i09>



Copyright:

<https://tu-dresden.de/Members/thomas.petzoldt>

Die Fotos im Poster sind gekennzeichnet

Nitrolimit-Broschüre, siehe Zitat

Die Rechte für die übrigen Fotos liegen bei den Autoren

Symbole: LibreOffice Symbolbibliothek

