

Wissenschaftliche Softwareentwicklung

31. Finale furioso: Zusammenfassung und Ausblick

Till Biskup

Physikalisch-Technische Bundesanstalt

11.03.2024





- 🔑 Programmierung ist eine Kernkompetenz für Naturwissenschaftler. Trotzdem hat sie einen viel zu geringen Stellenwert.
- 🔑 Wissenschaft beruht auf Nachvollziehbarkeit und Reproduzierbarkeit. Das ist in der Praxis oft nicht gewährleistet.
- 🔑 Die primäre Aufgabe von Code ist Kommunikation. Voraussetzung ist Ausdruckstärke und Lesbarkeit.
- 🔑 Kernaspekt aller Softwarearchitektur ist Modularität. Das führt zu Flexibilität und Wiederverwendbarkeit.
- 🔑 Wissenschaftliche Datenauswertung erfordert ein durchdachtes Gesamtkonzept. Einzelaspekte sind (relativ) einfach umsetzbar.

Zusammenfassung: Themen der Vorlesung im Überblick

Fünf Thesen

Ausblick: Gesamtsystem zur wissenschaftlichen Datenverarbeitung

Feedback

1 Motivation

- Programmierung ist essentieller Bestandteil von Forschung.

2 Infrastruktur

- notwendige Voraussetzungen für die Softwareentwicklung

3 Sauberer Code

- Kommunikation erfordert Ausdruckstärke und Lesbarkeit.

4 Softwarearchitektur

- Modularität führt zu Flexibilität und Wiederverwendbarkeit.

5 Datenverarbeitung und -Analyse in den Naturwissenschaften

- Gesamtkonzept von der Datenaufnahme bis zur Publikation

👉 Ziel: Software, die wissenschaftlichen Kriterien entspricht

Zusammenfassung: Themen der Vorlesung im Überblick

Fünf Thesen

Ausblick: Gesamtsystem zur wissenschaftlichen Datenverarbeitung

Feedback

These

Programmierung ist eine Kernkompetenz für moderne Naturwissenschaftler. Ihr Stellenwert ist trotzdem viel zu gering.

- Datenauswertung ist (fast) immer rechnergestützt.
 - Programmierkenntnisse werden vorausgesetzt, sind aber selten in ausreichendem Maß vorhanden.
- Die Komplexität der Software entspricht der Fragestellung.
 - Beherrschung erfordert Kenntnis entsprechender Konzepte
- Softwareentwicklung wird nicht ausreichend gewürdigt.
 - Entsprechend gering ist die Motivation, Zeit zu investieren.

These

Wissenschaft beruht auf Nachvollziehbarkeit und Reproduzierbarkeit.
Das ist in der Praxis oft nicht gewährleistet.

- Voraussetzungen: Dokumentation und Archivierung
 - Zeitskala: tendenziell Jahrzehnte
 - Stichworte: Unversehrtheit, Versionierung
- Die Realität sieht in vielen Fällen anders aus.
 - Nachvollziehbarkeit der Auswertungen nicht gegeben
 - Reproduzierbarkeit von publizierten Ergebnissen unmöglich

These

Die primäre Aufgabe von Code ist Kommunikation.
Voraussetzung ist Ausdrucksstärke und Lesbarkeit.

- Code wird viel häufiger gelesen als geschrieben.
 - „Code for people, not computers.“
 - (Wirklich) guter Code ist offensichtlich.
- Ausdrucksstärke erfordert Abstraktionsvermögen.
 - Konzepte durchdenken, verstehen und treffend benennen
 - Gute Namen sind das Ergebnis eines Prozesses.

These

Kernaspekt aller Softwarearchitektur ist Modularität.
Das führt zu Flexibilität und Wiederverwendbarkeit.

- Ansprüche an Software ändern sich ständig.
 - Das Verständnis der Problemstellung wächst.
 - Jede Fragestellung ist neu und leicht unterschiedlich.
- Gute Software ist wie ein Legokasten.
 - kleine, durchdachte, zueinander kompatible Bausteine
 - nahezu endlose Möglichkeiten, Neues zu schaffen

These

Wissenschaftliche Datenauswertung erfordert ein durchdachtes Gesamtkonzept. Einzelaspekte sind (relativ) einfach umsetzbar.

- Nachvollziehbarkeit erfordert lückenlose Dokumentation.
 - von der Datenaufnahme bis zur Veröffentlichung
 - weitestgehende Automatisierung einzelner Schritte
- plattform- und medienunabhängig und modular
 - nutzerfreundlich: Fokus auf einfacher Bedienbarkeit
 - Vorteile der Nutzung müssen offensichtlich sein.

Zusammenfassung: Themen der Vorlesung im Überblick

Fünf Thesen

Ausblick: Gesamtsystem zur wissenschaftlichen Datenverarbeitung

Feedback

ASpecD: A modular framework for the analysis of spectroscopic data focussing on reproducibility and good scientific practice

*Jara Popp, Till Biskup**



Chemistry—Methods **2**:e202100097, 2022

Herausforderungen

- Vollständig nachvollziehbare Datenverarbeitung und -Analyse inkl. eines lückenlosen Protokolls jedes Schrittes
- Den meisten Wissenschaftlern wurde nie formal beigebracht, wie man programmiert oder Software entwickelt.

Anforderungen

- Weitgehend automatisiertes „*scientific workflow system*“, das große Mengen von Daten verarbeiten kann
- Hinreichend einfach und elegant zu nutzen: sollte für einen Bachelor-Studenten funktionieren
- Passt sich an die (variablen) Anforderungen der Wissenschaft an: robust, modular, flexibel und (einfach) erweiterbar

Listing 1: Rezept

```
1 format:
  type: ASpecD recipe
3 version: '0.2'

5 datasets:
  - /path/to/first/dataset
7  - /path/to/second/dataset

9 tasks:
  - kind: processing
11   type: BaselineCorrection
    properties:
13     parameters:
        kind: polynomial
15     order: 0
  - kind: singleplot
17   type: SinglePlotter1D
    properties:
19     filename:
        - first-dataset.pdf
21     - second-dataset.pdf
```

Listing 2: Rezept-Historie

```
1 system_info:
  python:
3   version: "3.7.3 ..."
  packages:
5   aspect: 0.6.4
  # ...
7  - kind: processing
    type: BaselineCorrection
9   properties:
      parameters:
11     kind: polynomial
        order: 0
13     coefficients:
        - -0.04609818536259180
15     fit_area:
        - 10
17     - 10
        axis: 0
19  apply_to:
    - /path/to/first/dataset
21  # ...
```

☞ Wir haben meist eine klare Idee, was mit den Daten passieren soll.

- Rezeptgetriebene Datenauswertung
 - Keine Programmierkenntnisse notwendig
- Datensatz als Einheit von Daten und Metadaten
 - Abstraktion von den Hersteller-Dateiformaten
- Volle Reproduzierbarkeit
 - Die Historie ist ein voll funktionierendes Rezept
- Modular und erweiterbar
 - Fokussiert auf Anwendung, nicht auf die Infrastruktur
- Unterstützung für unterschiedliche (spektroskopische) Methoden
 - Python-Pakete für einige Methoden sind verfügbar

Verarbeitungsschritte

- Normalisation
- Integration
- Differentiation
- ScalarAlgebra
- Projection
- SliceExtraction
- SliceRemoval
- RangeExtraction
- BaselineCorrection
- Averaging
- ScalarAxisAlgebra
- DatasetAlgebra
- Interpolation
- Filtering
- CommonRangeExtraction
- Noise
- ChangeAxesValues
- RelativeAxis

Analyseschritte

- BasicCharacteristics
- BasicStatistics
- BlindSNREstimation
- PeakFinding
- PowerDensitySpectrum
- PolynomialFit
- LinearRegressionWithFixedIntercept

Plotter

- SinglePlotter
- SinglePlotter1D
- SinglePlotter2D
- SinglePlotter2DStacked
- MultiPlotter
- MultiPlotter1D
- MultiPlotter1DStacked
- CompositePlotter
- SingleCompositePlotter

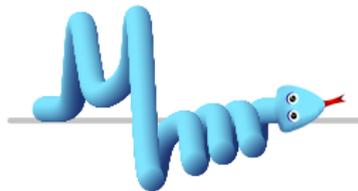
ASpecD



<https://docs.aspecd.de/>

J. Popp, T. Biskup. *Chem. Meth.* 2:e202100097, 2022

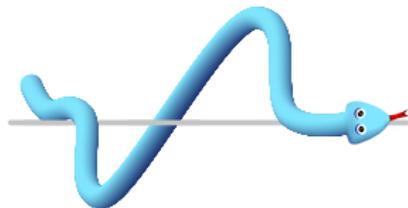
cwepr



<https://docs.cwepr.de/>

M. Schröder, T. Biskup. *J. Magn. Reson.* 335:107140, 2022

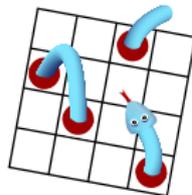
trEPR



<https://docs.trepr.de/>

J. Popp, M. Schröder, T. Biskup. *repr* (2022).
doi:10.5281/zenodo.4897112

FitPy

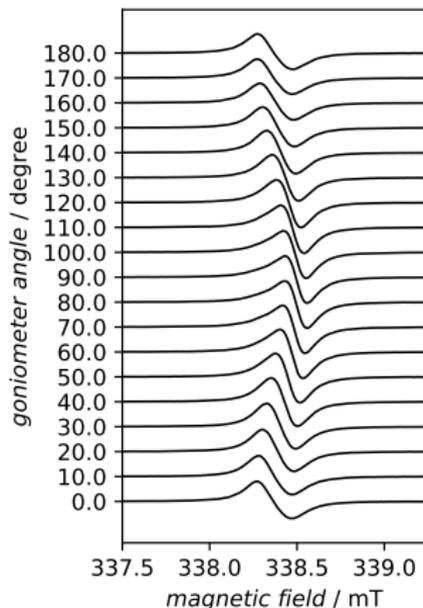
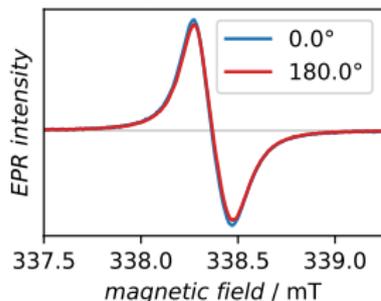
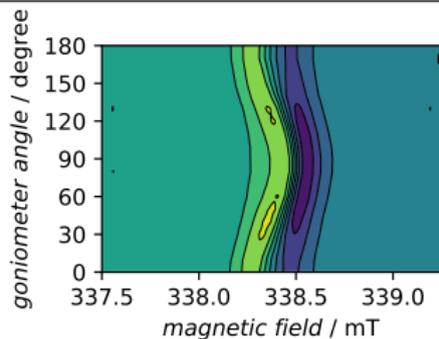


<https://docs.fitpy.de/>

T. Biskup. *fitpy* (2022).
doi:10.5281/zenodo.5920380

Listing 3: „Rezept“ zur Datenverarbeitung

```
1 format
2 type
3 vers
4
5 setting
6 defa
7
8 direct
9 data
10
11 dataset
12 - ma
13
14 tasks:
15 - kind
16 type
17 - kind
18 type
19 prop
20 pr
21
22 fi
```



Folgt bewährten Verfahren der Softwareentwicklung, z.B.:

- Testgetriebene Entwicklung
 - hohe Testabdeckung, bessere Verlässlichkeit
 - Sauberer Code
 - lesbar, ausdrucksstark, selbstdokumentierend
 - ausführlich dokumentiert
 - <https://docs.aspecd.de/>
 - Versionsverwaltungssystem
 - <https://github.com/tillbiskup/aspecd/>
 - quelloffen (open source)
 - BSD-Lizenz: jeder darf es verwenden und verändern
- 👉 Jeder Python-Programmierer könnte das Projekt übernehmen.

Zusammenfassung: Themen der Vorlesung im Überblick

Fünf Thesen

Ausblick: Gesamtsystem zur wissenschaftlichen Datenverarbeitung

Feedback

Ein paar Fragen als Inspiration:

- Waren die Themen verständlich?
- Hat die Thematik für Euch eine Relevanz?
- Entsprach die Vorlesung Euren Vorstellungen?
- Was fehlte?
- Was war zu viel?
- Fühlt Ihr Euch den Ansprüchen an die Programmierung wissenschaftlicher Datenauswertung (mehr) gewachsen?
- ...

Ideen für das weitere Vorgehen

- Sollte es Python-Kurse (mit unterschiedlichem Fokus) geben? Würden derlei Angebote auch wahrgenommen werden?
- Sollen einzelne Themen (in loser Folge) behandelt werden?
- Soll es regelmäßige Treffen geben, bei denen für Euch relevante und von Euch vorgeschlagene Themen/Fragen diskutiert werden?
- ...