Programmierkonzepte in der Physikalischen Chemie

1. Motivation (Physikalische Chemie)

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

Dr. Till Biskup

Institut für Physikalische Chemie
Albert-Ludwigs-Universität Freiburg
Wintersemester 2018/19

Zentrale Aspekte

UNI FREIBURG

- Rechnergestützte Datenauswertung spielt in der Physikalischen Chemie oft eine bedeutende Rolle.
  - Programmierung ist notwendiges Mittel zum Zweck.
- Software zur wissenschaftlichen Datenanalyse sollte einer Reihe von Anforderungen genügen:
  - Wiederverwendbarkeit, Selbstdokumentation, Zuverlässigkeit, Überprüfbarkeit, Nutzerfreundlichkeit, Erweiterbarkeit, Reproduzierbarkeit.
- Auswertungssoftware wird schnell komplex.
  - Kenntnis von Strategien professioneller Software-Entwicklung ist deshalb hilfreich.

Übersicht

UNI FREIBURG

Was ist Wissenschaft?

Das Wesen der Physikalischen Chemie

Anforderungen an die wissenschaftliche Datenanalyse

Größere Projekte erfordern Kenntnisse in Software-Entwicklung

Was ist Wissenschaft?

Ein kurzer Ausflug – und keine formale Antwort

UNI FREIBURG

> *If I have seen further*
> *it is by standing on y^e shoulders of giants.*
>
> – Sir Isaac Newton

Was ist der Kern von Wissenschaft?

- Erkenntnisgewinn
- Unabhängigkeit vom Betrachter/Experimentator
- gegründet auf den Erkenntnissen früherer Generationen
- überprüfbar, nachvollziehbar, ggf. reproduzierbar

Sir Isaac Newton: Brief an Robert Hooke, 5. Februar 1676

Die Verantwortung der Wissenschaftler

Wissenschaftliches Arbeiten ist eine verantwortungsvolle Tätigkeit

UNI FREIBURG

> *If I have seen further*
> *it is by standing on y^e shoulders of giants.*
>
> – Sir Isaac Newton

- Verantwortung gegenüber denen,
  die auf den gewonnenen Erkenntnissen aufbauen

Empirische Wissenschaften

- Interpretationen ändern sich, Daten sollten Bestand haben.
- Voraussetzung: Daten nach bestem Wissen und Gewissen akkurat aufgenommen (und dokumentiert)

Sir Isaac Newton: Brief an Robert Hooke, 5. Februar 1676

Das Wesen der Physikalischen Chemie

Verständnis der Grundlagen und Hintergründe

UNI FREIBURG

Physikalische Chemie

- Ordnung des experimentell gewonnenen Erfahrungsmaterials mit Hilfe der theoretischen, numerischen und experimentellen Methoden der Physik und der Theoretischen Chemie,
- Auffinden qualitativer Zusammenhänge und
- Aufstellen quantitativer Beziehungen

☛ Verständnis der Grundlagen und Hintergründe

Wedler, Freund: Lehrbuch der Physikalischen Chemie, 6. Aufl., Wiley-VCH, Weinheim 2012, S. XXIII

Das Wesen der Physikalischen Chemie

Motivation für die Beschäftigung mit Programmierkonzepten

UNI FREIBURG

- Die meisten Daten liegen heute elektronisch vor (egal ob Messdaten oder Ergebnisse von Rechnungen).
- Datenanalyse „realer“ Daten ist in der Regel so komplex, dass sie durch Rechner unterstützt wird.
- Ein reales Verständnis der Zusammenhänge erfordert häufig die eigenständige Entwicklung von Auswertungsstrategien.

☛ Programmierung von Auswertungssoftware kann einen wesentlichen Teil der wissenschaftlichen Arbeit in der Physikalischen Chemie ausmachen.

Wissenschaftliche Datenanalyse

Die Anforderungen sind die gleichen wie für die experimentelle Arbeit

UNI FREIBURG

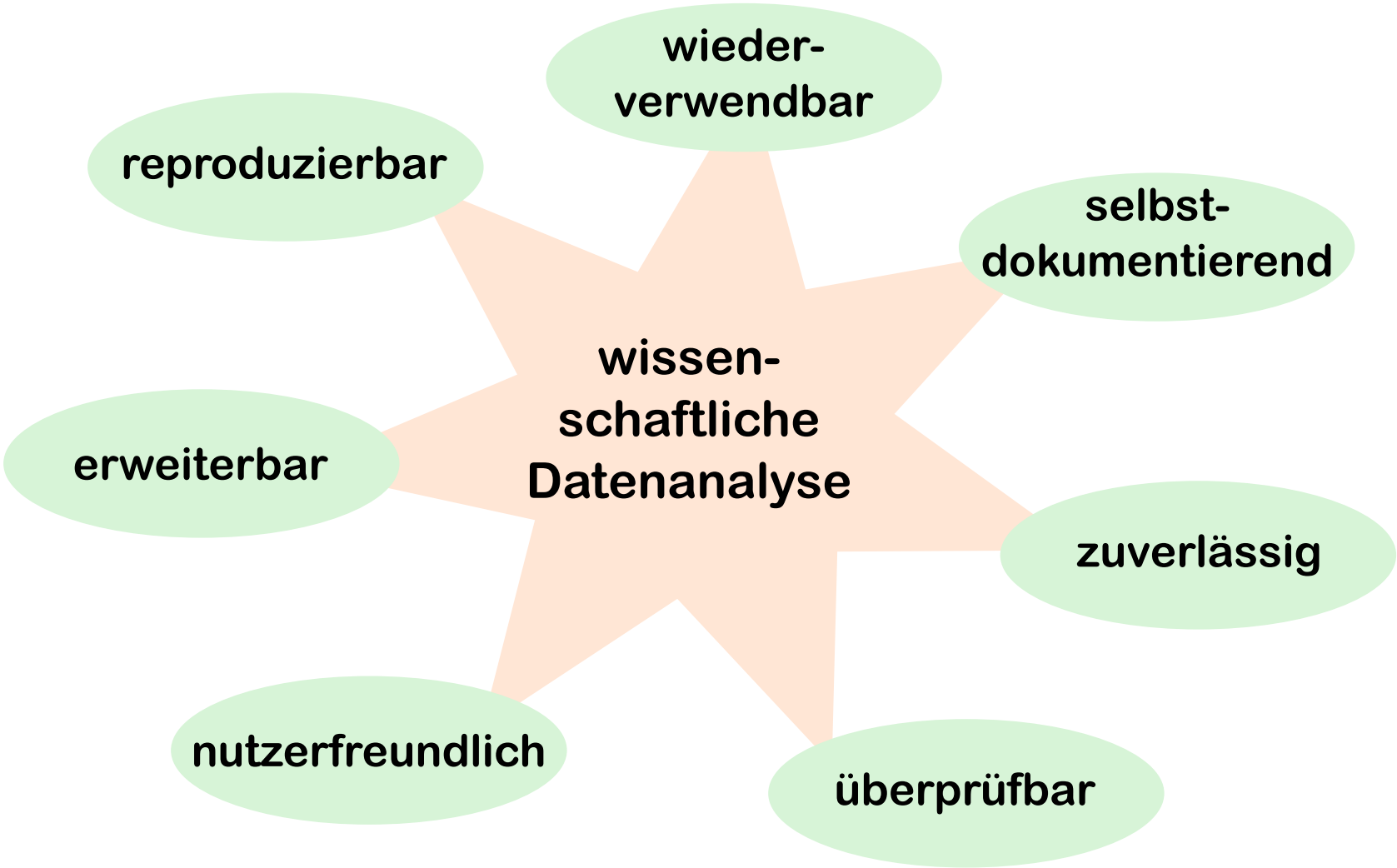
> *If experimentalists don't calibrate their equipment, check their reagants' purity, and take careful notes, what they're doing isn't considered science.*
> *In contrast, computationalists don't even learn how to assess their software's quality in any systematic way, and very few would be able to recreate and rerun the programs they used to produce last year's papers.*
> *As a result, most computational science is irreproducible and of unknown quality.*
>
> – Greg Wilson

- Ziel der Vorlesung:
  bessere Software zur Datenauswertung

Wilson, *Comput. Sci. Eng.* **8**:66, 2006

Wissenschaftliche Datenanalyse

Übersicht über die Anforderungen



Zwei Strategien für Auswertungssoftware

- ein Skript pro Datensatz
- ein verallgemeinerter Satz von Routinen zur Auswertung

Aspekte von Wiederverwendbarkeit

- später vom Autor oder anderen Nutzern
- für ähnliche Fragestellungen
- im Kontext anderer Fragestellungen

Wiederverwendbarkeit

Aspekte aus Programmierersicht

UNI FREIBURG

- modular
  - eine Aufgabe pro Modul
  - klein genug, um es vollständig erfassen zu können
  - abstrakt genug, um es wiederverwenden zu können
- Code les-/verstehbar
  - vollständiges Verständnis jeder Codezeile
  - Alternative: Bibliothek sauber dokumentierter Routinen
- Tests
  - Veränderungen sollen Funktionalität nicht beeinträchtigen
  - nur durch (automatisierte) Tests sicherstellbar

Wiederverwendbarkeit

Aspekte aus Programmierersicht

UNI FREIBURG

- zukunftssichere Formate
  - Daten sind viel langlebiger als Formate in der EDV.
- plattformunabhängig
  - akademischer Kontext: mehrere Plattformen verbreitet
  - Software von Anfang an plattformunabhängig entwickeln
- sprachunabhängig
  - „Program into a language, not in a language.“
  - Konzepte sauber dokumentieren (externe Dokumentation)
- klare Urheberrechte/Lizenzen
  - Quellcode unterliegt *per se* dem Urheberrecht.
  - Weiterverwendung nur bei klarer Lizensierung möglich

Selbstdokumentation

Generelle Aspekte

UNI FREIBURG

- Zentraler Aspekt der empirischen Wissenschaften: Reproduzierbarkeit
- Voraussetzung bei der Datenauswertung:
  - vollständige und lückenlose Dokumentation aller Schritte
  - inklusive Parameter (und Randbedingungen)
- Idealvorstellung
  - komplett automatisierte Dokumentation
  - automatische Erzeugung gut formatierter (lesbarer) Berichte mit den Ergebnissen

Selbstdokumentation

⌨ Aspekte aus Programmierersicht

UNI FREIBURG

- modular
  - jeder Verarbeitungsschritt in separater Routine
  - Dokumentation der Parameter/Randbedingungen
    ggf. ebenfalls durch separate Routine
- (möglichst) einheitliche Schnittstellen
  - vereinfacht die automatische Dokumentation der Parameter/Randbedingungen
- Trennung von Auswertung und Bericht
  - Bericht auf Lesbarkeit für Menschen optimiert
  - Tipp: Verwendung eines Vorlagensystems (Templates)

Zuverlässigkeit hat mehrere Aspekte je nach Kontext

- Programmierung
  - unabhängig von (unerwünschten) Nebenwirkungen:
  - keine Fehler durch (finite) numerische Genauigkeit
  - keine Fehler in der Implementierung (abhängig von bestimmten Parametern)
- Wissenschaften
  - korrekte Ergebnisse unabhängig von den Parametern
  - ggf. klare Definition des gültigen Wertebereiches für jeden Parameter einer Auswertungsroutine

☞ Benennung des Kontextes, Verweis auf Limitationen und auf die Implementierung können wesentlich sein.

Zuverlässigkeit

Aspekte aus Programmierersicht

UNI FREIBURG

- Code les-/verstehbar
  - Verständnis zentrale Voraussetzung der Datenauswertung
  - Der Wissenschaftler ist für das Verständnis verantwortlich.
- robust
  - Überprüfung der Parameter auf Sinnhaftigkeit
  - Detektion numerischer Ungenauigkeiten/Instabilitäten
- modular
  - eine Aufgabe pro Routine
- Tests
  - (automatisierte) Überprüfung der Korrektheit
  - wissenschaftliche Korrektheit ggf. schwer testbar

Überprüfbarkeit

Generelle Aspekte

UNI FREIBURG

> *Vertrauen ist gut, Kontrolle ist besser!*
>
> – Lenin (zugeschrieben)

- Zentraler Aspekt der empirischen Wissenschaften: Reproduzierbarkeit
- Voraussetzung bei der Datenauswertung:
  - transparente und nachvollziehbare Dokumentation jedes einzelnen Verarbeitungsschrittes
  - Zugriff auf den Quellcode der Auswertungsroutinen
- ☛ In der Praxis (leider) eher selten gegeben...

Überprüfbarkeit

Aspekte aus Programmierersicht

UNI FREIBURG

- Konzepte sauber dokumentiert
  - Quellcode ist idealerweise selbsterklärend.
  - übergreifende Konzepte extern dokumentieren
- Code les-/verstehbar
  - Voraussetzung für die Nachvollziehbarkeit, ob die Auswertung korrekt ist
  - Idealerweise ist Auswertungssoftware quelloffen.
- Tests
  - automatisiert
  - Fehler in Tests verwandeln

Nutzerfreundlichkeit

Generelle Aspekte

UNI FREIBURG

> *Code for people, not computers*
>
> – Programmierer-Regel

- Je einfacher sich eine Software nutzen lässt,
  desto mehr wird sie genutzt werden.
- Schwer nutzbare Software führt ggf. zu „Abkürzungen“,
  die zentrale Konzepte *ad absurdum* führen.

Nutzerfreundlichkeit

Aspekte aus Programmierersicht

UNI FREIBURG

- intuitive (und stabile) Schnittstellen
  - so einfach wie möglich nutzbar
  - (inkompatible) Änderungen nur nach Vorwarnung
- robuster Code
  - Fehler durch falsche Eingaben abfangen
- Nutzerdokumentation
  - Beschreibung der Schnittstelle und Nutzung jeder Routine
  - Fokus auf dem Anwender, nicht auf dem Entwickler
- auf Nutzerbedürfnisse hören
  - Entwickler werden schnell „betriebsblind“.
  - Auswertungssoftware ist immer Mittel zum Zweck.

Erweiterbarkeit

Generelle Aspekte

UNI FREIBURG

- Viele Aspekte der Datenverarbeitung sind Routineaufgaben.
- Wissenschaft lebt von immer wieder neuen (und unvorhergesehenen) Anforderungen.
- Software zur Datenanalyse sollte von Anfang an auf einfache Erweiterbarkeit hin ausgelegt werden.

☛ eigentliche kreative Tätigkeit in der Wissenschaft: Anwendung der vorhandenen „Werkzeuge“ in neuer Weise

Erweiterbarkeit

Aspekte aus Programmierersicht

UNI FREIBURG

- modular
  - eine Aufgabe pro Routine
  - möglichst kleine/kurze Routinen
- Code les-/verstehbar
  - Code wird viel häufiger gelesen als geschrieben.
  - Voraussetzung für Anwendung in neuem Kontext
- Tests
  - Änderungen sollten bestehende Funktionalität nicht negativ beeinflussen.
  - automatisierte Tests

Erweiterbarkeit

Aspekte aus Programmierersicht

- Versionsverwaltung
  - Entwicklung verläuft selten linear.
  - Nutzung bestehender Versionen (inkl. Fehlerbehebung) während der Weiterentwicklung
  - jederzeit Rückgriff auf alte Versionen möglich
- Schnittstellen und Konzepte dokumentiert
  - Schnittstellendokumentation ggf. im Code
  - Konzepte in externer Dokumentation

Reproduzierbarkeit

Generelle Aspekte

UNI FREIBURG

- Zentraler Aspekt der empirischen Wissenschaften: Reproduzierbarkeit
- Damit zusammenhängende Aspekte: Selbstdokumentation, Überprüfbarkeit
- Reproduzierbarkeit geht einen Schritt weiter:
  - Konkrete Version/Implementierung aller verwendeten Routinen sollte nachvollziehbar sein.
  - Ergebnisse sollten im Rahmen der Möglichkeiten vollständig identisch reproduzierbar sein.

Reproduzierbarkeit

Aspekte aus Programmierersicht

UNI FREIBURG

- Dokumentation *aller* Parameter,
  die für eine Operation auf den Daten verwendet wurden
  - inklusive der Standardwerte (ggf. voreingestellt)
  - ggf. vollständig automatisiert
  - Versionsnummern für Routinen, Betriebssystem, etc.
  - ggf. Nutzer und Datum/Uhrzeit
- Versionsnummern
  - Voraussetzung für die Nachvollziehbarkeit
  - im Kontext einer Versionsverwaltung (s.u.)
- Versionsverwaltung
  - Zugriff auf alle alten Versionen
  - gleichzeitig Voraussetzung für verteilte Entwicklung

Zwei Kategorien von Programmieraspekten

Infrastruktur

- (externe) Dokumentation
- Versionsverwaltung
- Versionsnummern

Code

- modular
- lesbar
- robust
- getestet/testbar

- kein Anspruch auf Vollständigkeit
- Beide Kategorien werden nachfolgend im Detail behandelt.

Größere Projekte und Software-Entwicklung

Ausgangsthese der Vorlesung

UNI FREIBURG

These

Software zur wissenschaftlichen Datenauswertung ist i.d.R. so komplex, dass die Kenntnis von Konzepten der professionellen Softwareentwicklung eine notwendige Voraussetzung für qualitativ hochwertigen Quellcode ist.

- Komplexität kommt aus dem Anspruch der Wissenschaftlichkeit
- Verantwortung des einzelnen Wissenschaftlers, den Ansprüchen gerecht zu werden

Größeres Projekt

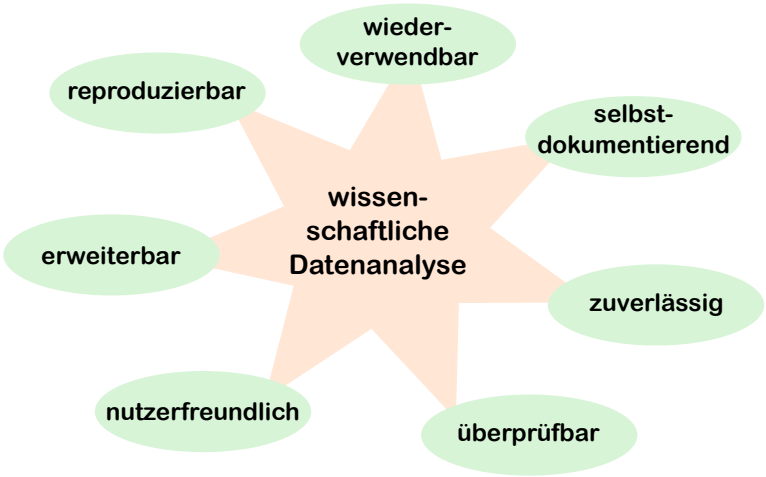
Alles, was mehr als zwei Wochen Arbeit kostet und deutlich mehr als zweihundert Zeilen (reinen) Quellcode bzw. mehr als eine Handvoll Unterfunktionen umfasst

- Genaue Zahlen sind immer problematisch:
  - Was in Assembler zweihundert Zeilen sind, ist in Python vielleicht eine...
- Wichtig ist der Fokus:
  - Soll ein Programm über längere Zeit verwendet werden?
  - Soll ein Programm von anderen verwendet werden?

Größere Projekte und Software-Entwicklung

Auswertungssoftware wird schnell komplex

Beispiel für Komplexität



Software-Entwicklung

- Infrastruktur
  - Arbeitserleichterung
- Code
  - konkrete Funktionalität
- Architektur
  - Zusammenspiel der Komponenten

- Problem komplexer Software seit den 1960ern bekannt
- ☛ Kenntnis existierender Strategien bewahrt davor, das Rad neu zu erfinden, und hilft, sich auf die eigentliche Aufgabe (Datenauswertung/Verständnis) zu konzentrieren.

Zentrale Aspekte

UNI FREIBURG

- Rechnergestützte Datenauswertung spielt in der Physikalischen Chemie oft eine bedeutende Rolle.
  - Programmierung ist notwendiges Mittel zum Zweck.
- Software zur wissenschaftlichen Datenanalyse sollte einer Reihe von Anforderungen genügen:
  - Wiederverwendbarkeit, Selbstdokumentation, Zuverlässigkeit, Überprüfbarkeit, Nutzerfreundlichkeit, Erweiterbarkeit, Reproduzierbarkeit.
- Auswertungssoftware wird schnell komplex.
  - Kenntnis von Strategien professioneller Software-Entwicklung ist deshalb hilfreich.