



Institut für Physikalische Chemie

**Übungen zum Methodenkurs „Programmierkonzepte in der Physikalischen Chemie“  
im WS 2013/2014**

Dr. Till Biskup

— Aufgabenblatt 5 vom 29.01.2014 —

---

### Aufgabe 5—1

Schreiben Sie eine *robuste* Funktion zum Einlesen von UV/vis-Daten, die sich an der in Listing 1 definierten Schnittstelle orientiert. Die Ausgabe, `data`, soll hier eine Matrix mit zwei Spalten (Wellenlänge und Intensität) sein, der Eingabeparameter, `filename`, ist eine Zeichenkette (*string*).

Listing 1: Schnittstelle einer Funktion zum Einlesen vom UV/vis-Daten

```
data = loadUVVisData(filename)
```

---

Gehen Sie für das zu importierende Dateiformat davon aus, dass die Daten als ASCII-Textdateien mit Komma als Dezimaltrennzeichen vorliegen<sup>1</sup>. Ein Beispiel finden Sie im auf der Internetseite des Methodenkurses<sup>2</sup> bereitgestellten ZIP-Archiv (Datei „`uvvis.txt`“).

Ihre Funktion soll folgenden Ansprüchen genügen:

- Einlesen von Absorptionsdaten, die im beschriebenen ASCII-Exportformat vorliegen.
- Einlesen reiner ASCII-Daten ohne Dateikopf und mit Punkt als Dezimaltrenner.  
Ein Beispiel finden Sie im bereitgestellten ZIP-Archiv (Datei „`background.txt`“).
- Überprüfung der Existenz des angegebenen Dateinamens.
- Automatische Ergänzung der Dateiergung (`.txt`), wenn nicht angegeben.
- „Graceful exit“: Bei Problemen eine Warnmeldung ausgeben, verbunden mit der Rückgabe einer leeren Matrix (`data`).

Zum Test Ihrer Funktion steht Ihnen eine eigene Testroutine<sup>3</sup> zur Verfügung, die Ihre Funktion hinsichtlich der oben genannten Kriterien überprüft und Ihnen zusätzlich eine Zusammenfassung der durchgeführten Tests liefert. Nutzen Sie das als Chance für testgetriebene Entwicklung in der Praxis. Schreiben Sie Ihre Funktion so, dass sie jeden der Tests besteht und möglichst wenige Warnungen angezeigt werden. (Hinweis: Nicht alle Warnungen sind komplett eliminierbar.)

---

<sup>1</sup>Es handelt sich um das ASCII-Exportformat der Software eines großen Herstellers optischer Absorptionsspektrometer. Die Verwendung eines Kommas statt eines Punktes als Dezimaltrennzeichen verkompliziert den Import in Matlab geringfügig.

<sup>2</sup><http://till-biskup.de/de/lehre/programmierkonzepte/>

<sup>3</sup>Diese Testroutine wurde eigens für diesen Zweck geschrieben. Die elegantere Lösung, das Unit-Test-Framework, hätte den Preis der Abhängigkeit von aktuellen Matlab-Versionen (2013a und neuer) oder zusätzlichen (freien) Toolboxes.

## Aufgabe 5—2

Machen Sie sich mit den Möglichkeiten vertraut, die Ihnen Matlab bietet, um Ihren Code zu beschleunigen. Einen guten Einstieg ermöglicht die Matlab-Hilfe, insbesondere die Kapitel „Techniques for Improving Performance“ und „Vectorization“ aus dem Abschnitt „Code Performance“<sup>4</sup>.

**Tipp:** Mit einem MathWorks-Konto können Sie sich die PDF-Version der Matlab-Handbücher von der MathWorks-Internetseite herunterladen.

Gehen Sie jedes der dort aufgeführten Beispiele durch und versuchen Sie, es im Detail zu verstehen. Ziel sollte sein, einen Überblick über die Funktionen von Matlab zu bekommen, die häufig für die Vektorisierung und die Code-Beschleunigung genutzt werden. Viele der vorgestellten Funktionen sind in dieser Form für Matlab spezifisch, andere Programmiersprachen bieten andere Möglichkeiten der Optimierung.

Darüber hinaus gibt es weitere interessante Artikel zum Thema auf den Seiten von MathWorks zum Thema, z.B. zur Indizierung von Matrizen („Matrix Indexing in MATLAB“<sup>5</sup>), auf die u.a. aus den Handbüchern heraus verwiesen wird.

## Aufgabe 5—3

Viele Operationen sind bei einmaliger Ausführung zu schnell, als dass die Zeitmessung mit den Matlab-Bordmitteln (`tic`, `toc`) verlässlich wäre. Dazu kommen Schwankungen in der Geschwindigkeit der Ausführung durch parallel vom jeweiligen Rechner verarbeitete andere Prozesse. Für aussagekräftige Ergebnisse der Zeitmessung muss also über mehrere bis viele Durchläufe gemittelt werden.

Auf der zum Kurs gehörigen Internetseite finden Sie eine Reihe konkreter Beispiele für zwei unterschiedliche Implementationen der jeweils gleichen Aufgabe, eine eher „herkömmliche“<sup>6</sup> und eine für die Gegebenheiten von Matlab optimierte Variante.

Schreiben Sie eine Funktion (ggf. auch mehrere Funktionen oder Unterfunktionen), die den Rechenzeitbedarf dieser Paare von alternativen Implementationen der gleichen Aufgabe ermittelt. Führen Sie dazu eine (frei wählbare) Zahl an Funktionsaufrufen (Iterationen) aus, mitteln Sie die Zeit für die einzelnen Iterationen und berechnen Sie die Standardabweichung. Achten Sie darauf, die Funktionen zur Zeitmessung an den korrekten Stellen einzusetzen. Sie wollen lediglich die beiden Implementationen direkt vergleichen, unverfälscht durch zusätzliche (einmalig notwendige) Vorarbeiten. Variieren Sie zudem noch die Größe der jeweils verwendeten Datenstruktur (`cell array` bzw. Matrix/Vektor).

Stellen Sie in einem zweiten Schritt Ihre Ergebnisse grafisch in geeigneter Weise vergleichend für die beiden Implementationen dar. Was fällt auf? Was bedeutet das für die Bedeutung der jeweiligen Optimierungsstrategie?

**Tipp:** Meist ist es aussagekräftiger, wenn man die Größen der jeweiligen Datenstrukturen über Größenordnungen (logarithmisch) und nicht linear variiert. Eine doppelt logarithmische Auftragung liefert bei linearer Abhängigkeit der benötigten Rechenzeit eines Algorithmus von der Größe der zu verarbeitenden Daten entsprechend eine Gerade. Die Form der Kurve lässt Rückschlüsse über die Ordnung (Komplexität einer Aufgabe) der jeweils verwendeten Algorithmen bzw. Implementationen zu.<sup>7</sup>

---

<sup>4</sup><http://www.mathworks.de/de/help/matlab/code-performance.html>

<sup>5</sup><http://www.mathworks.de/company/newsletters/articles/matrix-indexing-in-matlab.html>

<sup>6</sup>Matlab teilt sich mit vielen anderen Programmiersprachen grundsätzliche Sprachaspekte, wie z.B. Schleifen. Aus einer gewissen Erfahrung mit anderen Programmiersprachen ergibt sich somit oft ein Tendenz für Lösungsmuster, die in Matlab aufgrund seiner Besonderheiten nicht unbedingt effizient sind.

<sup>7</sup>Für Interessierte zwei Stichworte zur weiteren Recherche: O-Notation bzw. Landau-Symbole.