

Anleitung für das Kommandozeilen-Programm XtremA

B. Sc. Felix Krickl

27. Januar 2022

Inhaltsverzeichnis

1	Disclaimer	2
2	Anwendungsbereich und Funktionalität	2
3	Installation und Aufrufen	3
3.1	Direkte Ansprechen des .py-files	3
3.2	Kompilierung zu einem binary file	3
4	Format	5
4.1	Eingabedatei	5
4.2	Ausgabedatei	5
5	Syntax	6
5.1	Allgemeines	6
5.2	Übersicht aller Argumente und Erläuterungen	6
6	Anwendungsbeispiele	7
6.1	Minimalbeispiel	7
6.2	Komplizierteres Beispiel	11

1 Disclaimer

Diese Software unterliegt dem Urheberrecht von B. Sc. Felix Krickl und wird unter der Gnu Public License v2.0 verteilt. Der Lizenz-Text ist in der beiliegenden Datei „gpl-2.0.pdf“ nachzulesen.

2 Anwendungsbereich und Funktionalität

XtremA ist ein Kommandozeilenprogramm zur Bestimmung von lokalen Extremstellen in einem csv-Datensatz. Dies schließt sowohl Minima, Maxima, als auch Wendestellen ein. Prinzipiell lässt sich das Programm für jeden Datensatz anwenden, der mindestens x,y-Wertepaare enthält.

Die Erkennung der Extremstellen funktioniert über simple numerische Differentiation des Eingabedatensatzes. Nach dem notwendigen und hinreichenden Kriterium wird entschieden, um welche Art von Extremstelle es sich handelt. Dieser Ansatz ist natürlich sehr empfindlich und wird bei verrauschten Datensätzen zur Erkennung zahlreicher Extremstellen im Rauschen führen. Für viele Anwendungen genügt dieser Ansatz aber bereits. In jedem Fall sollte man die Ausgabe sichten und prüfen bevor man damit weiterarbeitet. In zukünftigen Versionen soll, wenn möglich, das Prinzip der Prominenz von Extremstellen implementiert werden, um die Empfindlichkeit der Erkennung kontrollieren zu können.

Die Nützlichkeit insbesondere der Wendestellen-Erkennung wird maßgeblich durch die Auflösung im Datensatz bestimmt. Bei geringer Auflösung um die Extremstellen herum werden mitunter auch Minima und Maxima als Wendepunkt erkannt. Für Messkurven geringer Komplexität und hoher Auflösung ist die Erkennung aber gut.

Hauptaugenmerk sind bei diesem Programm jedoch tatsächlich nur die Minima und Maxima. Besonders empfehlenswert ist es bei der Analyse von Spektrendaten wie IR-Spektren. Hier gilt es, aus einem relative langen Datensatz eine Reihe von lokalen Extremstellen auszulesen, was mit gängiger Tabellenkalkulation machbar ist, aber mit **XtremA** schneller geht.

Die Ausgabe des Programms ist wiederum ein csv-Datensatz, in dem die x,y-Wertepaare wiedergegeben werden und in zusätzlichen Spalten die gewählten Extremstellen-Arten (Eine Spalte pro Extremstellenart). Die Extremstellen werden dabei dargestellt durch den y-Wert des Datenpunkts, an dem sie erkannt wurden. Dieser y-Wert erscheint dann auch in der Zeile, in der das entsprechende Wertepaar steht. Bei den Zeilen, in denen keine Extremstelle erkannt wurde, wird stattdessen ein *leerer String* geschrieben.

Dieses Format erlaubt es, die Ausgabe in einem Tabellenkalkulationsprogramm direkt einzulesen und ist auch praktisch bei der Verwendung von plotting Programmen wie zum Beispiel *Gnuplot*. Hier könnte man sich etwa mit dem plot-style *impulses* die Extremstellen anzeigen lassen und mit *lines* das Spektrum (oder den sonstigen Datensatz).

XtremA ändert nie die Eingabedatei und löscht diese auch nicht. Gibt man als Ausgabedatei einen bereits existierende Datei an, wird man gefragt ob tatsächlich in diese Datei geschrieben werden soll, oder nicht. Wenn man die Abfrage verneint, muss der Befehl noch einmal eingegeben werden.

3 Installation und Aufrufen

Es gibt im Wesentlichen zwei Arten, **XtremA** zu installieren und zu verwenden. Beide setzen eine vorhandene Python-Installation voraus. Bestenfalls wäre das die Version 3.6. Andere Versionen werden auch funktionieren. Die erste Art ist das

3.1 Direkte Ansprechen des .py-files

also des source codes, aus einer Python-Shell heraus, oder durch die Eingabe von

```
python3 xtrema.py
```

in der Kommandozeile. Diese Art hat den Vorteil, dass sie eigentlich keinen Installationsvorgang benötigt. Der Nachteil ist jedoch, dass hierfür das xtrema.py-file immer im selben Verzeichnis sein muss, wie die Eingabedatei. Ein weiterer Nachteil ist, dass man jedes mal python3 und die Dateiendung .py mitschreiben muss. Allerdings lassen sich diese Nachteile beseitigen, indem man die Datei in einem Verzeichnis ablegt, welches in der PATH-Variable steht. Bei mir ist das zum Beispiel ~/bin. Dann muss die Datei ausführbar gemacht werden und auch die Dateiendung kann man weglassen:

```
sudo chmod a+x xtrema.py
mv xtrema.py xtrema
```

3.2 Kompilierung zu einem binary file

Die Kompilierung von python code ist wenigstens unter Linux mit dem Programm PyInstaller möglich. Um den PyInstaller zu installieren, muss man aber erst pip installieren (eine python-utility). Das funktioniert in ubuntu-basierten Distributionen mit dem Befehl (root Rechte vorausgesetzt)

```
sudo apt-get install python3-pip
```

Als nächstes kann man den PyInstaller selbst installieren mit:

```
python3 -m pip install pyinstaller
```

Dann folgt der eigentliche Kompilierungsschritt. Hierzu legt man sich im besten Fall ein neues Arbeitsverzeichnis an, in welches man das .py-file kopiert. Nach Navigation in dieses Verzeichnis benutzt man den Befehl

```
pyinstaller --onefile xtrema.py
```

dies wird eine Menge an Dateien und Unterverzeichnissen produzieren. Wichtig ist - solange alles richtig funktioniert hat - nur die Datei xtrema (ohne Dateiendung) im dist-Unterverzeichnis. Dies ist die gewünschte binär Datei und sie sollte 5,7 MB groß sein. Dieses binary sollte jetzt zum Abschluss in ein Verzeichnis verlegt werden, das in der \$PATH Variable gespeichert ist. In den meisten Fällen wird /usr/bin funktionieren (was die Konvention ist). Im Zweifelsfall lässt man sich die \$PATH Variable anzeigen mit

```
echo $PATH
```

Ist das binary im korrekten Verzeichnis macht man es ausführbar, beispielsweise durch

```
sudo chmod a+x xtrema
```

wenn alle user **XtremA** benutzen können sollen oder

```
sudo chmod u+x xtrema
```

wenn nur der aktuell angemeldete user **XtremA** benutzen können soll. In letzterem Fall müsste man das u vor dem + nicht mal mitschreiben.

Hat man alles richtig gemacht müsste es nun möglich sein, von jedem working directory aus xtrema direkt aufzurufen, indem man schlicht

```
xtrema
```

eingibt.

Bemerkung: Mit pip und Pyinstaller bekommt man leider häufiger mal verschiedenste Probleme. Diese kann ich hier nicht alle beschreiben. Sollte etwas nicht funktionieren muss ich daher generisch auf die jeweilige offizielle Quelle und auf Foren verweisen. Man könnte argumentieren dass diese Art der Installation und Verwendung schlechter ist, weil die Größe der binary Datei die des python-Codes deutlich überschreitet. Allerdings sind 5,7 MB nicht wirklich ein Argument.

4 Format

4.1 Eingabedatei

Die Dateieindung der Eingabedatei spielt für XtremA keine Rolle. Wichtig ist nur die Formatierung innerhalb der Datei. Im Wesentlichen ist dies die Formatierung von csv-Dateien (comma separated values):

1. Der zu verarbeitende Datensatz muss mit Punkten als Dezimalinterpunktation dargestellt werden (angelsächsische Darstellung).
2. Hilfsinterpunktationen für bessere Lesbarkeit in den Zahlen können nicht verwendet werden.
3. Die einzelnen Zahlen in einer Spalte müssen entweder durch Kommata, Freizeichen (Leerzeichen) oder Tabulator (default) voneinander getrennt sein.
4. Die Zeilen dürfen nicht von Leerzeilen unterbrochen werden. Eine freie Zeile am Ende des Datensatzes führt zu einem „Traceback“-Fehler.
5. Eine Kopfzeile kann eingebaut werden, muss aber nicht. Wenn eine Kopfzeile verwendet wird, muss diese durch ein voraus gestelltes „#“ gekennzeichnet werden.
6. Datensätze, die mehr als zwei Spalten besitzen, können auch verarbeitet werden. Allerdings immer nur 2 Spalten davon gleichzeitig (ein Wertepaar). Die restlichen Spalten werden dann ignoriert.

4.2 Ausgabedatei

In der Ausgabedatei werden die Wertepaare, von denen die Extremstellen bestimmt wurden in den ersten beiden Spalten erneut wiedergegeben. Die Spalten rechts davon sind den Extremstellen zugeordnet; in der Reihenfolge, in der diese in der Kommandozeile erfragt wurden. In den Spalten der Extremstellen wird jeweils an der Position der Extremstelle ihr y-Wert wiedergegeben. Somit ist es möglich, wahlweise den Datensatz selbst, oder auch die Extremstellen als einzelne Datenpunkt aus der Ausgabedatei heraus zu plotten.

5 Syntax

5.1 Allgemeines

Die Reihenfolge der angegebenen Argumente spielt für XtremA keine Rolle. Man kann sich an die Reihenfolge in untenstehender Liste halten, aber man kann stattdessen zum Beispiel auch erst die Ausgabedatei angeben, dann die Eingabedatei und zuletzt das **xtype**-Argument.

5.2 Übersicht aller Argumente und Erläuterungen

-d, --delimiter Der Delimiter ist das Trennsymbol, mit dem die einzelnen Werte in der Eingabedatei voneinander getrennt sind. Mögliche Werte sind: **comma**, **space**, **tab**. Der Delimiter aus der Eingabedatei wird auch bei der Ausgabedatei wieder verwendet.
Default: **tab**.

-i, --input Dateinamen der Eingabedatei mit Dateiendung angeben. Wie die Dateiendung lautet ist egal. Wichtig ist die Formatierung innerhalb des Eingabedokuments. Die Endung ist dabei beliebig. txt-Dateien zum Beispiel werden auch problemlos eingelesen.
Nutzungsbeispiel: **-i input.csv**.
Die Angabe des input-Arguments und des output-Arguments ist obligatorisch.

-o, --output Dateinamen der Ausgabedatei mit Dateiendung angeben. Wie input Argument.

-t, --xtype Art der Extremstellen, die erkannt werden sollen. Es kann ausgewählt werden aus **min**, **max**, **inf**.
min bezieht sich auf Minima, **max** auf Maxima und **inf** auf Wendepunkte (Inflections). Es können auch mehrere Optionen in beliebiger Reihenfolge ausgewählt werden. Die Angabe dieses Arguments mit wenigstens einer Option ist obligatorisch. Es existiert **kein Default**.

-u, --using Das using-Argument bezieht sich auf die einzulesenden Spalten aus der Eingabedatei. Beginnend links mit eins wird erst die Spalte, welche den Definitionswert enthält angegeben, dann ein Doppelpunkt geschrieben und dann die Spalte, welche den Funktionswert enthält angegeben.
Nutzungsbeispiel: **-u 3:4** sagt XtremA, dass die dritte Spalte in der Eingabedatei die x-Werte enthält und die vierte Spalte die y-Werte. Es kann auch eine spätere Zeile als x-Wert-Zeile definiert werden und eine frühere als y-Wert-Zeile.
Default: **1:2**.

6 Anwendungsbeispiele

6.1 Minimalbeispiel

Wir beginnen mit dem Datensatz. Hier wurde die Funktion $-(x-5)^2+5(x-7)^3+8$ verwendet, um mit LibreOffice Calc die Datei „Datenreihe_minimal.csv“ zu kreieren:

0.00	-360
0.25	-322
0.50	-287
0.75	-254
1.00	-224
1.25	-196
1.50	-171
1.75	-147
2.00	-126
2.25	-107
2.50	-89.4
2.75	-73.8
3.00	-60.0
3.25	-47.8
3.50	-37.1
3.75	-27.9
4.00	-20.0
4.25	-13.4
4.50	-7.88
4.75	-3.45
5.00	0.00
5.25	2.58
5.50	4.38
5.75	5.48
6.00	6.00
6.25	6.02
6.50	5.63
6.75	4.92
7.00	4.00
7.25	2.95
7.50	1.88
7.75	0.86
8.00	0.00
8.25	-0.61
8.50	-0.88
8.75	-0.70
9.00	0.00
9.25	1.33
9.50	3.38
9.75	6.23
10.00	10.0

Wir navigieren nun in das Verzeichnis dieser Datei und geben folgendes Kommando ein:

```
xtrema -i Datenreihe_minimal.csv -o Datenreihe_minimal_Maxima.csv  
-t max
```

Dies sollte folgenden output im Terminal ergeben:

```
Eingabedatei eingelesen  
Differentialle berechnet  
Ausgabedatei geschrieben  
Programm ohne XtremA Errors beendet.
```


Und die Datei „Datenreihe_minimal_Maxima.csv“ mit folgendem Inhalt sollte erzeugt werden:

0.00	-360	
0.25	-322	
0.50	-287	
0.75	-254	
1.00	-224	
1.25	-196	
1.50	-171	
1.75	-147	
2.00	-126	
2.25	-107	
2.50	-89.4	
2.75	-73.8	
3.00	-60.0	
3.25	-47.8	
3.50	-37.1	
3.75	-27.9	
4.00	-20.0	
4.25	-13.4	
4.50	-7.88	
4.75	-3.45	
5.00	0.00	
5.25	2.58	
5.50	4.38	
5.75	5.48	
6.00	6.00	
6.25	6.02	6.02
6.50	5.63	
6.75	4.92	
7.00	4.00	
7.25	2.95	
7.50	1.88	
7.75	0.86	
8.00	0.00	
8.25	-0.61	
8.50	-0.88	
8.75	-0.70	
9.00	0.00	
9.25	1.33	
9.50	3.38	
9.75	6.23	
10.00	10.0	

Der Unterschied zur Eingabedatei ist nun die 6.02 in der dritten Spalte, welche das Maximum markiert.

Die gefundene Extremstelle könnte nun grafisch dargestellt werden, indem wir die Datei „plot-Minimal“ mit folgenden Anweisungen für GNUPLOT erstellen:

```
set xlabel "x"
set ylabel "y"
set xrange [0:20]
set yrange [-10:10]
set key bottom right

plot 'Datenreihe_minimal_Maxima.csv' u 1:2 t "$-(x-5)^2+5(x-7)^3+8$"
w l lw 2 lc '#00008800', \
'Datenreihe_minimal_Maxima.csv' u 1:3 t "Maxima" w imp lw 2
lc '#00880000'
```

```
set terminal epslatex
set size 1
set output "graphDatenreihe_minimal.tex"
replot
set output
```

Dies ergäbe den Plot in Abbildung 1.

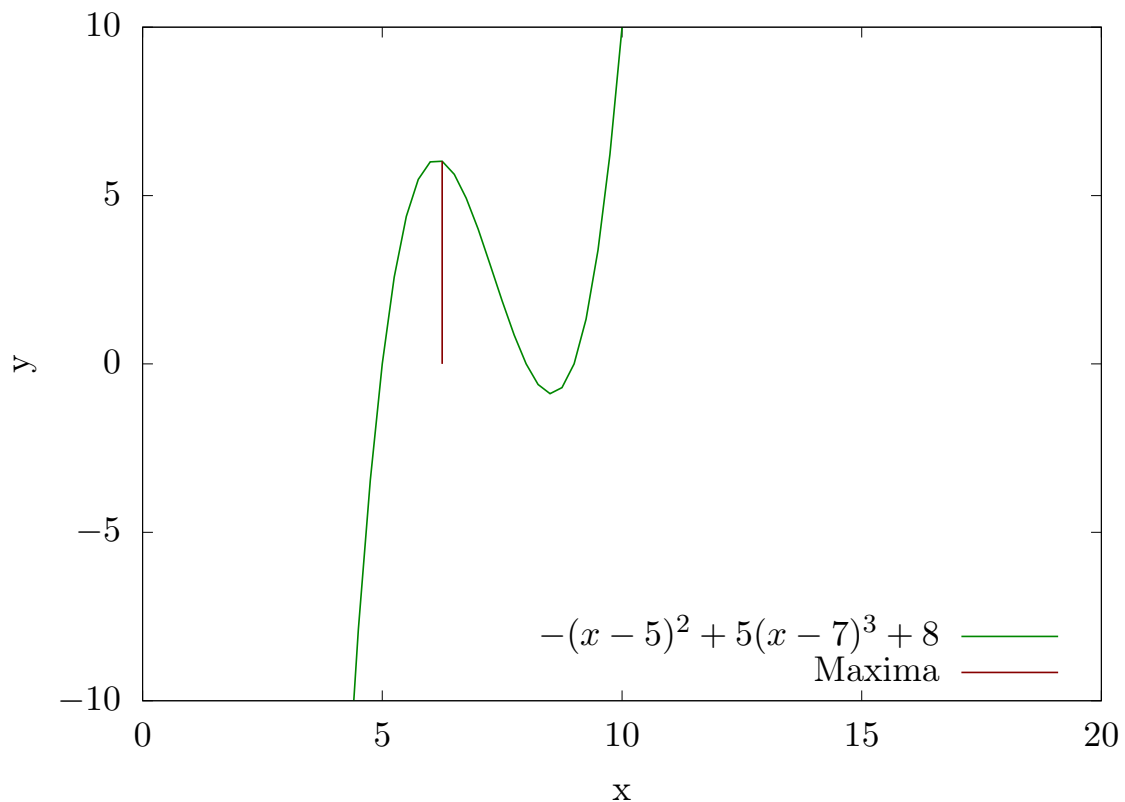


Abbildung 1: Plot einer simplen ganzrationalen Funktion mit Markierung des lokalen Maximums.

6.2 Komplizierteres Beispiel

Wir haben einen csv-Datensatz vorliegen, welcher einen Ausschnitt aus dem Infrarot-Spektrum einer bestimmten Substanz entspricht. Um die Positionen der Absorptionsbanden herauszufinden, benutzen wir **XtremA** mit der Option „-t min“:

```
xtrema -i IR-Daten.csv -o IR-Daten_Minima.csv -t min
```

Die generierten Daten in „IR-Daten_Minima.csv“ könnten wir gemäß dem folgenden GNU-PLOT-Code visualisieren:

```
set xlabel "Wellenzahl  $[cm^{-1}]$ "
set ylabel "Transmission  $[\%]$ "
set xrange [1500:1000]
set yrange [80:100]

set key box

plot 'IR-Daten_Minima.csv' u 1:2 w l lw 2 lc
'#00008800' t "Spektrum", \
      'IR-Daten_Minima.csv' u 1:3 w imp lw 2 lc '#00880000'
t "Absorptionsbanden"

set terminal epslatex
set size 1
set output 'graphKompliziert.tex'
replot
set output
```

Das Ergebnis ist in Abbildung 2 zu sehen. Es ist zu erkennen, das **XtremA** nur echte Peaks erkennt. Schultern werden nicht beachtet.

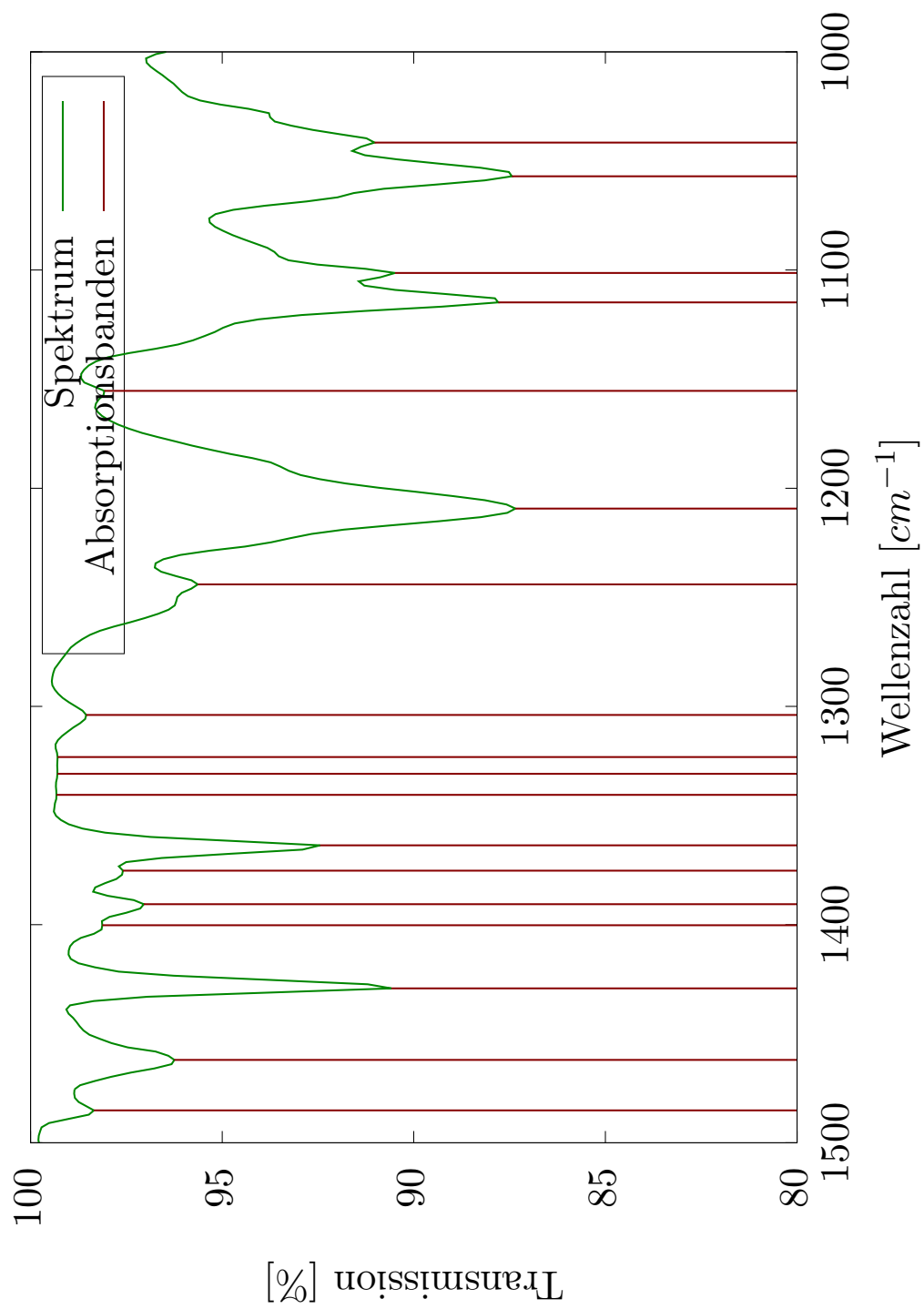


Abbildung 2: Plot eines Ausschnitts eines IR-Spektrums mit Markierung der Absorptionsbanden.