

Berechnung von Lichtkurven mit *lightcurve*, lehrbuchartige Anleitung

Diese Anleitung entstammt eine Mail an Andreas Beck, der sich für die Berechnung von Lichtkurven interessiert hat. Aus diesem Grund werdet ihr mit „du“ angesprochen und nicht in der Mehrzahl. Es ist kein Gewinn für den Text, das alles zu ändern, also bleibt es so.

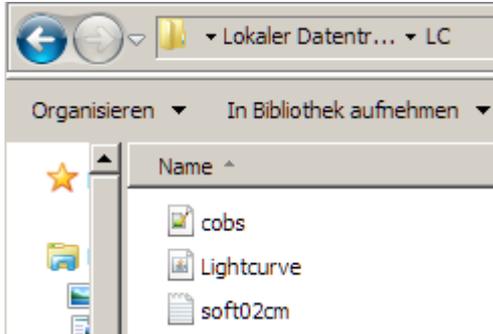
Uwe Pilz, Ende April 2016.

Installation:

Es müssen nur Dateien in ein Verzeichnis gelegt werden.

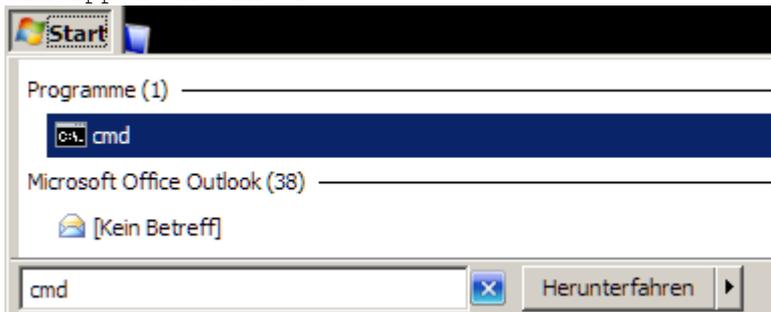
Ich habe alle notwendigen Dateien auf unserer Archivseite <http://kometen.fg-uds.de/archiv.htm> hinterlegt, dort halte ich sie auch aktuell. Du benötigst cobs.zip, soft02cm.zip und Lightcurve.zip. Die ersten beiden musst du regelmäßig aktualisieren.

Zur Arbeit mit dem Lichtkurvenprogramm legst du die am besten ein eigenes Verzeichnis an. Ich verwende hier im Beispiel V:\LC, für dich ist wahrscheinlich c:\LC geeignet. Die Inhalte der drei zip-Archive (jeweils eine Datei) müssen nach c:\LC geschoben werden. Es sollte dann so aussehen:



Programm probierhalber starten

Du startest eine Konsole (schwarzes Fenster) : Start - „cmd“ in das weiße Feld eintippen und Enter



Es öffnet sich die Eingabeaufforderung. Diese hat immer ein sogenanntes Arbeitsverzeichnis im Bauch, welches auch angezeigt wird. Normalerweise ist dies „User“+der Anmelde-name, bei mir also c:\user\0pilz:



Wenn du `cd \LC` eintippst (dem Rückstrich nicht vergessen), dann wechselt diese Anzeige zu `c:\LC`. Wenn du eine anderes Verzeichnis benutzt hast oder benutzen musstest, dann musst du dieses eingeben. Falls das Ganze gar nicht auf dem Laufwerk C: stattfindet, sondern wie bei mir auf V:, dann musst du vor dem `cd`-Befehl noch das Laufwerk wechseln, in dem du z.B. V: eintippst (mit Doppelpunkt). Wenn du `dir` eintippst, dann bekommst du die drei Dateien angezeigt:

```
Administrator: Command Prompt
U:\LC>dir
Volume in drive U is Users
Volume Serial Number is 009A-9A03

Directory of U:\LC

14.04.2016 09:17 <DIR> .
14.04.2016 09:16 <DIR> ..
14.04.2016 08:59      33.327.243 cobs.dat
14.04.2016 09:00      140.851 soft02cm.txt
14.04.2016 08:59      63.683 Lightcurve.jar
           3 File(s)      33.531.777 bytes
           2 Dir(s)     1.766.105.088 bytes free

U:\LC>_
```

Du solltest prüfen, ob Java installiert ist. Das geht. In dem du `java -version` eintippst. Wenn es fehlt, sieht das so (oder so ähnlich aus):

```
U:\LC>java -version
'java' is not recognized as an internal or external command,
operable program or batch file.

U:\LC>_
```

Es muss dann installiert werden, wofür Adminsitratorenrechte nötig sind. Du brauchts nur die Laufzeitumgebung JRE. Diese gibt es z.B. unter <http://www.java.com/de/download/manual.jsp> .

Wenn java installiert ist, meldet es sich mit einer Versionsnummer. Ich habe hier 1.8, das Programm wird auch unter 1.7 laufen, noch ältere Java-Versionen genügen nicht.

```
v:\LC>java -version
java version "1.8.0_73"
Java(TM) SE Runtime Environment (build 1.8.0_73-b02)
Java HotSpot(TM) Client VM (build 25.73-b02, mixed mode, sharing)

v:\LC>_
```

Da es java nun gibt, kann auch das Programm aufgerufen werden mit `java -jar Lightcurve.jar`

```
v:\LC>java -jar Lightcurve.jar
Hmm...no comet given.

v:\LC>_
```

Das Programm merkt an, dass es nicht weiß, welcher Komet betrachtet werden soll: Es hat somit erst mal gearbeitet und die Installation war erfolgreich.

Du solltest jetzt noch prüfen, ob die Bahnelemente- und Messwert-Dateien lesbar sind und probierhalber eine Auswertung durchführen. Hierzu die Zeile mit Pfeil hoch wieder hervorholen (damit man nicht immer alles eintippen muss und `-gk81` dahinter schreiben. Jetzt gibt es ein Rechenergebnis für den Kometen 81P.

```
v:\LC>java -jar Lightcurve.jar -gk81
81: m = 7.30 + 14.42 log r + 5 log D [n=5.77, r^2=0.58, N=509, fqs=0.821862]

v:\LC>
```

Damit ist die Installation abgeschlossen und geprüft. Klingt alles kompliziert, aber wenn du einmal so weit bist, bleibt es beim nächsten Mal bei Konsole öffnen - Verzeichnis wechseln - los geht's. Nun gut, vorher vielleicht noch die Messwerte- und Bahnelemente-Dateien neu herunterladen.

Generelle Arbeit mit dem Programm

Konsolenanwendungen sind meist so gebaut, dass man dem Programm vorher sagt, wie es arbeiten soll, und das macht es dann einfach. Wenn das falsch oder unzweckmäßig war, startet man das Programm neu mit geänderten Vorgaben. Diese Vorgaben starten immer mit einem Minuszeichen, gefolgt von einem weiteren Zeichen (meist ein Buchstabe). Oft kommt dahinter noch ein weiteres Wort. Angaben, die nur aus einem Buchstaben bestehen, heißen Schalter, solche mit noch einem Wort dahinter heißen Parameter. Beispiel für Schalter: Der Schalter `-h` bewirkt, dass eine Hilfeseite ausgegeben wird. Wenn du `java -jar Lightcurve.jar -h` eintippst (h wie Hilfe), bekommst du einen Hilfebildschirm mit allen verfügbaren Schalter und Parameter angezeigt (ist ziemlich

viel). Was nicht angezeigt wird ist, wie man das sinnvoll benutzt. Dazu gibt es einen andern Schalter -H. Hier kommt ein Bildschirm, der die Grundzüge der Benutzung in Erinnerung ruft. Ich werde ab sofort übrigens statt `java -jar Lightcurve.jar ...` einfach `Lightcurve ...` schreiben. Das ist kürzer, und es lässt sich leicht einrichten, dass man das tatsächlich so eintippen kann. Bei mir ist es so eingerichtet. Beispiel für einen Parameter: `-k` (wie Komet) erlaubt es, einen Kometen anzugeben. Bei periodischen ist es nur die Nummer, bei periodischen JahrBuchstabeNummer: Also `Lightcurve -k2013su10` oder `Lightcurve -k252` oder meinetwegen auch `Lightcurve -k 252` (dem Programm ist es egal, ob du noch ein Leerzeichen zwischen dem Buchstaben oder dem Wort tippst). Es erscheint eine Grafik, die in diesem Fall ziemlich wild aussieht. Wenn man das Grafik-Fenster geschlossen hat, sieht man in der Konsole die Rechenergebnisse als Text.

Die Eingabe von `Lightcurve -kC/2013US10` oder `Lightcurve -k252P` führt zum Fehler „Comet not found in orbital elements' file“, bitte unbedingt die Kurzform benutzen. Optionen und Parameter können kombiniert werden. So erzeugt der Schalter `-P` eine Grafikversion, welche sich besser für Veröffentlichung in Zeitschriften eignet. Ein Komet muss zusätzlich angegeben werden:

```
Lightcurve -k 252 -P
```

Wenn man nicht mehr weiß, welche Option wie heißt, kann man immer `-h` hinten dran schreiben. Es kommt dann nur die Hilfe, es wird nichts gerechnet. Mit Pfeil hoch kann man die letzte Zeile holen, das `-h` löschen und weiter an der Parameterverfeinerung basteln.

Schalter können gestapelt werden und ganz zum Schluss kann noch ein Parameter stehen. `Lightcurve -k 252 -P -G -K` erzeugt für den Kometen 252 (`-k`) eine Kurve für eine Printmedium (`-P`), benutzt eine deutsche (German, `-G`) Legende und gibt Kalenderdaten (`-K`) statt des Abstandes vom Perihel aus. Da ich geübt bin im Umgang hiermit würde ich viel kürzer

```
Lightcurve -PGKk252
```

schreiben. Die gesamten Schalter sind ohne extra Striche hintereinandergeschrieben, und am Ende noch der Parameter `-k` samt dem Kometennamen. Das muss man zwar nicht machen, meine Beispiele sind aber oft so.

Ein einfaches, reeles Beispiel

(Dieses Beispiel wird nur mit dem jetzigen (April 2016) Datenbestand dieselben Ergebnisse erzeugen. Nicht erschrecken, wenn etwas anderes auf dem Bildschirm erscheint. Es geht ums Prinzip.)

Der Komet 81P/Howell kann derzeit beobachtet werden. Eine Lichtkurve gibt es erst einmal mit `Lightcurve -k81`. In der Grafik erscheint zwar die theoretische Kurve, aber keine Messwerte, es gibt lila Kleckse. Das liegt daran, dass das Perihel erst im Juli ist und die Grafik nur ± 100 Tage darum herum zeigt. Ein verwertbares Textergebnis gibt es aber: $m_0=7,5$ $n=5.77$, dazu die Berechnungsformel, in die das 2,5-fache von n eingeht. Dazu das Bestimmtheitsmaß r der Regression und die Fehlerquadratsumme.

Mit `-h` kurz die Optionen anzeigen lassen ... ah ja, `-d` (Tage vor dem Perihel).

`Lightcurve -k18 -d300` sieht schon besser aus, allerdings kleben die Messwerte am unteren Rand. Obere und untere Magnitude der Grafik lassen sich mit `-m` und `-M` verschieben: `Lightcurve -k81 -d300 -m18 -M10`. Das sieht schon mal gut aus, allerdings passt die Kurve nicht ganz zu den Messwerten. Es könnte aber sein dass manche Punkte „mehrfach“ sind (identische Messungen zum selben Zeitpunkt), und die Kurve dorthin ziehen. Um das zu prüfen, lässt man sich auf der Konsole die Messwertliste ausgeben.

Die Grafik schaltest du mit `-g` ab, die stört jetzt nur, und dafür mit `-L` die Daten an, welche die lineare Regression bekommt. Die Grafikoptionen lasse ich stehen, sie werden ignoriert, wenn man die Grafik abschaltet: `Lightcurve -k81 -d300 -m18 -M10 -gL`

Die Liste ist viel zu lang, um sie vernünftig würdigen zu können. Man kann sie entweder seitenweise anzeigen, in dem man „more“ dahinter schaltet, oder in eine Datei umleiten und mit irgendwas (z.B. einem Editor) anzeigen zu lassen. Die einfachere Variante mit more:

```
Lightcurve -k81 -d300 -m18 -M10 -gL | more
```

Ergibt (Ausschnitt)

$\log(r)$	mH	r	D	$m1$	yyyymmdd	day
0.204	12.053	1.598	0.976	12.000	19780203	-14048
0.204	11.654	1.598	0.975	11.600	19780203	-14048

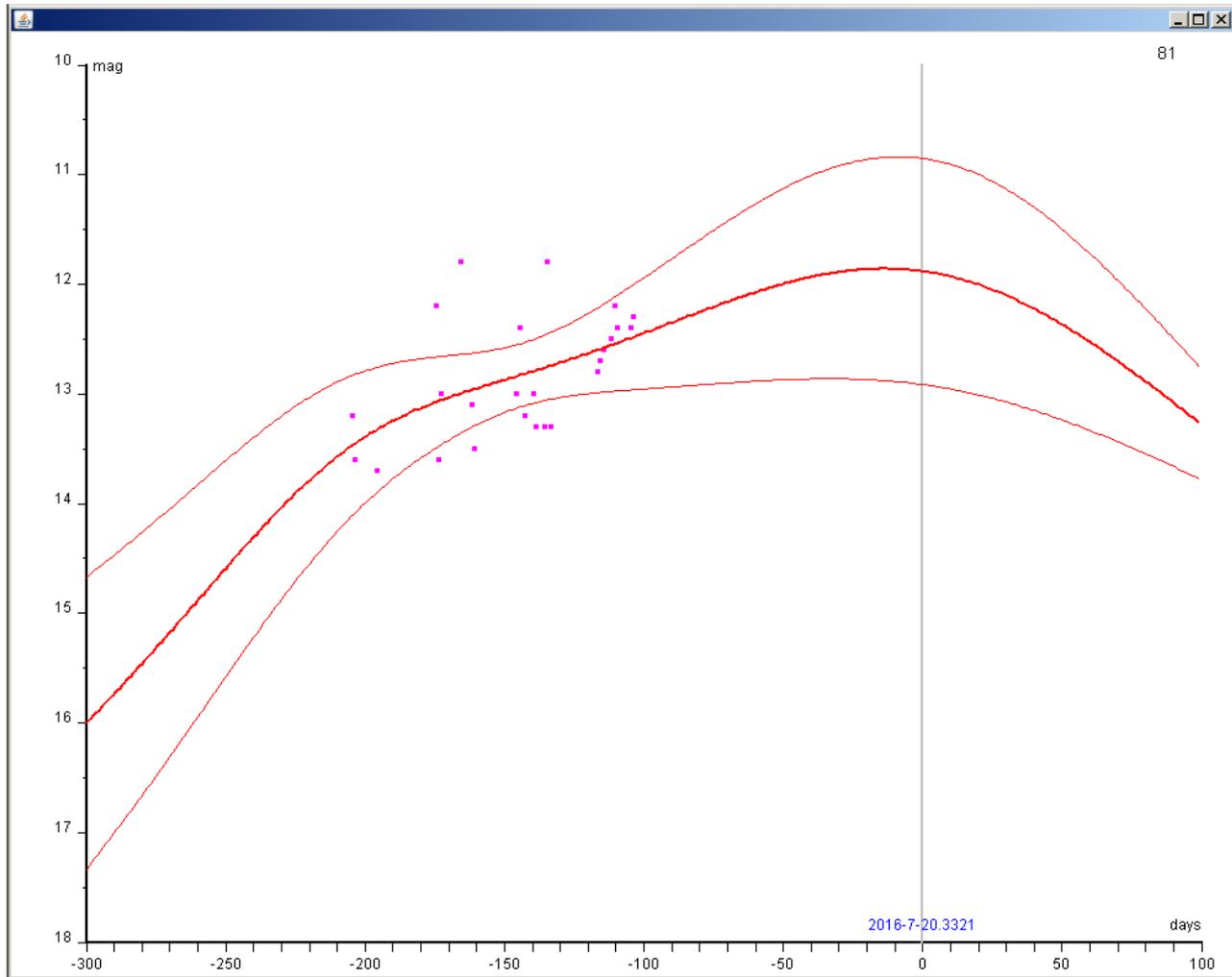
Ausgabe in Datei würde mit `Lightcurve -k81 -d300 -m18 -M10 -gL > meienDatei.txt` ermöglicht.

In der vorletzten Spalte ist das Kalenderdatum zu sehen, und dort erkennt man auch, was los ist: Es sind Daten der vorigen Perihels mit in die Rechnung eingegangen. Um diese

auszublenden gibt es die Optionen `-t -T (time)`. Für uns genügt es, den frühesten Zeitpunkt festzulegen:

```
Lightcurve -k81 -d300 -m18 -M10 -t 20150101 -gL
```

Die Liste passt jetzt auf den Bildschirm und zeigt an, dass der jüngste Messwert vom Dezember 2015 ist, das kommt hin. Kurve wieder anschalten (`-gL` weg), und voilà, alles passt viel besser:



(Falls eine Grafik einmal unvollständig angezeigt wird, das Fenster verkleinern und wieder vergrößern, das hilft)

Auch die Ergebnisse haben sich geändert, in diesem Perihel gilt $m_0=7,39$ und $n=5,36$:

```
Lightcurve -k81 -d300 -m18 -M10 -t 20150101
```

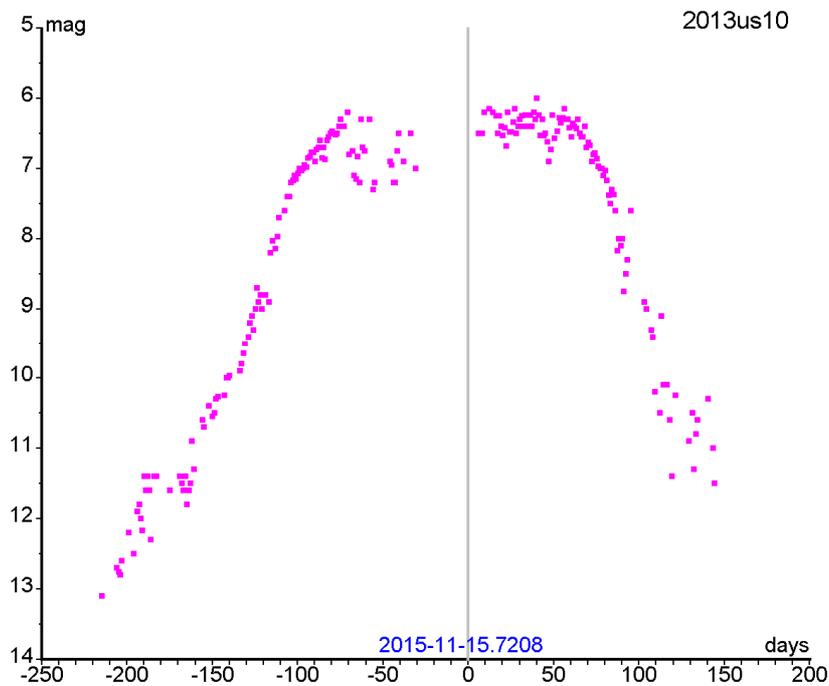
```
81: m = 7.39 + 13.40 log r + 5 log D [n=5.36, r^2=0.50, N=25, fqs=0.217437]
```

Das Bestimmtheitsmaß ist eher schlecht je näher an 1 desto besser), was man auch in den Streubalken der Kurve sieht. Wenn diese stören, dann kann man sie mit `-v` abschalten. Wenn man einmal nur die Messwerte braucht, kann man sogar die Vorhersagelinie abschalten (`-V`), dann besser aber `-vV`, sonst kommen die Konfidenzintervalle trotzdem.

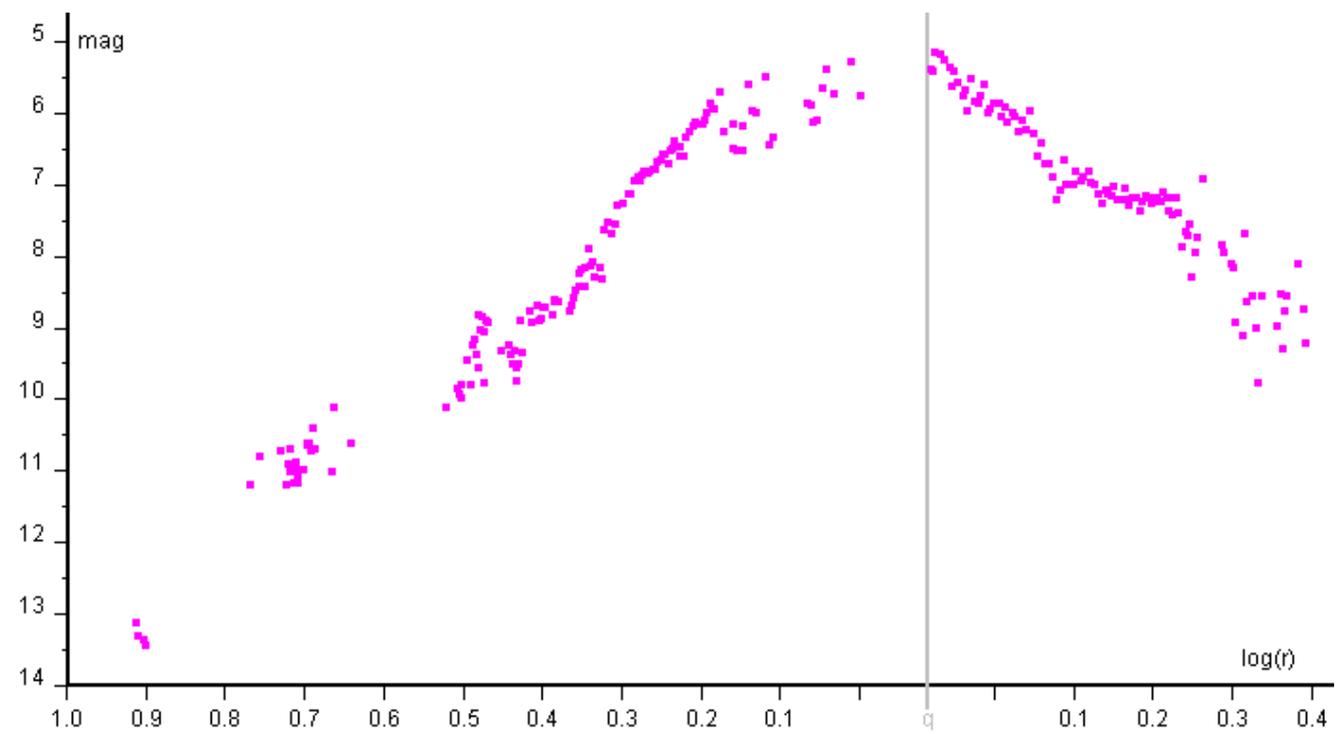
Stückweise Analyse

Es gibt Kometen, die plötzlich aktiv werden, oder einen Einbruch erleiden. Wenn dies mehr als ein geringes Maß ist, sollte man über eine stückweise Analyse nachdenken. Das Prinzip ist so, dass man die Daten mit `-t -T` eingrenzt und jeweils Kurvenabschnitte berechnet. Ich nehme mir C/2013 US10 (Catalina) her, der einiges zu bieten hat

```
Lightcurve -k2013us10 -vV -d250 -D200 -P -M5
```



Besser sieht man das heliozentrisch (... -R8)



Ich entscheide mich für folgende Phasen

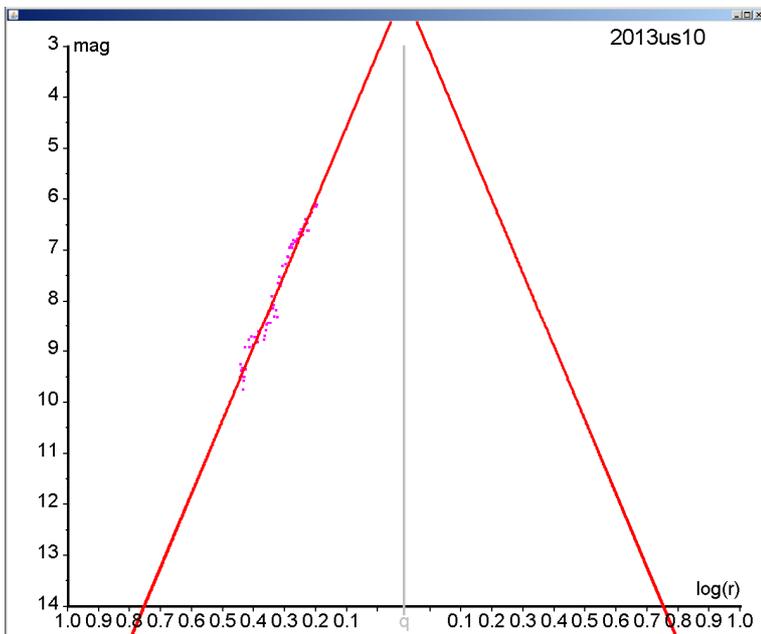
- 1) Einen Abschnitt von Anfang bis $\log(r)=0.35$
- 2) Den stärkeren Helligkeitsanstieg zwischen $\log(r)=0.35$ und $0,20$
- 3) Den Einbruch in der Nähe des Perihels
- 4) Den Abschnitt nach dem Perihel.

Da die Rechnung in Kalenderdaten geht, muss ich mir diese zu den r -Werten besorgen. Hierzu kann die Listenoption `-L` dienen, Grafik am besten wieder abschalten:

Es ergeben sich die Abschnitte:

- 1) Anfang - 12.7.15
- 2) 12.7.15 - 31.8. 15
- 3) 31.8.15 - 15.11.15 (Periheldatum, steht nicht in der Liste)
- 4) 15.11.15 - Ende

Man kann das nur rechnen lassen, aber eine grafische Anzeige ist auch ganz gut. Man sieht aber stets nur die Werte, welche durch die `-t` und `-T` Schalter eingegrenzt wurden. Beispiel für 2): `Lightcurve -Pvk2013us10 -d250 -D200 -R8 -t 20150526 -T 20150831`



Notiert werden müssen die absolute Helligkeit m_0 und der Aktivitätsparameter. Mit den Daten, die ich jetzt habe, ermittele ich

m_0	n
1) 5.86	2.95
2) 3.10	5.81
3) 5.69	0.98
4) 5.93	3.05

Die Aktivitätsparameter geben das schon grafisch ermittelte Verhalten gut wieder: Die Abschnitte steileren Helligkeitsanstieges haben auch einen höheren Aktivitätswert. In der Auswertung kann man angeben, dass $n \sim 3$ der „normale“ Aktivitätsparameter des Kometen ist, es einen Ausbruch mit n knapp 6 gab, dadurch die oberflächennahen Wasserreserven erschöpft wurden und es danach einen Einbruch gab, der das wieder ausgleicht. Dafür spricht auch, dass eine Rechnung über den gesamten Zeitraum ein n von knapp über 3 ergibt.

Das Programm ermöglicht es, aus diesen Einzelwerten eine Gesamtkurve zu zeichnen. Das ist allerdings etwas unhandlich. Man benötigt den Schalter `-s`, der für jeden Abschnitt ein Stück einer Eingabe benötigt, der in der Hilfe gegeben ist:

`_ddd+DDD+mmmmnnn`

Hier bedeutet:

`_ddd` : Tage bis zum Perihel mit Vorzeichen, Beginn des Abschnitts. Das Vorzeichen Minus (vor dem Perihel) muss durch einen Unterstrich gegeben werden

`+DDD` : Tage bis zum Perihel mit Vorzeichen, Ende des Abschnitts. Das Vorzeichen Minus (vor dem Perihel) muss durch einen Unterstrich gegeben werden

`+mmm`: Absolute Helligkeit mit Vorzeichen, eine Dezimale, wobei der Dezimalpunkt weggelassen wird

`nnn`: Aktivitätsparameter mit einer Kommastelle, auch ohne Dezimalpunkt

Das kann beliebig oft wiederholt werden.

Zuerst muss man sich die Tagesabstände zum Perihel abschreiben. Das Programm bietet hierzu eine Unterstützung in Form der Liste der Eingangswerte, Option `-L`. Diese sollte mit `-g` (Grafik abschalten kombiniert werden:

```
Lightcurve -k2013us10 -d250 -D200 -R8 -gL | more
```

Das Datum steht in der vorletzten Spalte, die Differenz zum Perihel in der letzten. Es ergibt sich:

```
12.7.15 -126
```

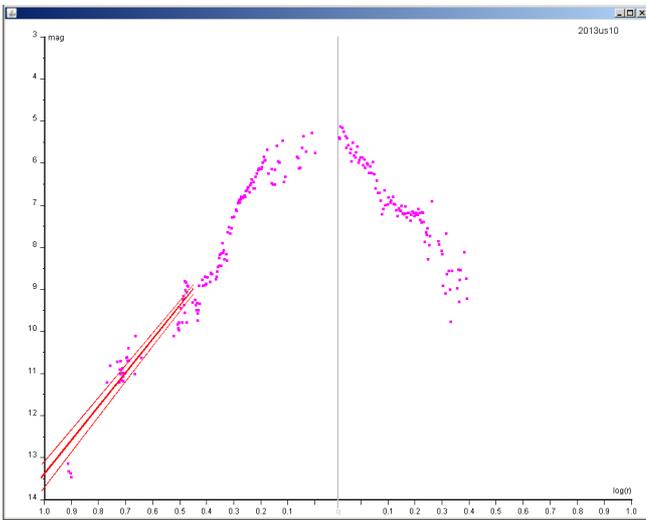
```
31.8.15 -77
```

```
15.11.15 = 0 (Periheldatum, steht meist nicht in der Liste)
```

Den Text hinter `-s` baut man am besten schrittweise auf, weil man sich hier leicht vertut. Der erste Abschnitt geht von „minus unendlich“ bis `-172`:

```
_999_172+054032:
```

```
Lightcurve -k2013us10 -d250 -D200 -R8 -s _999_172+054032
```

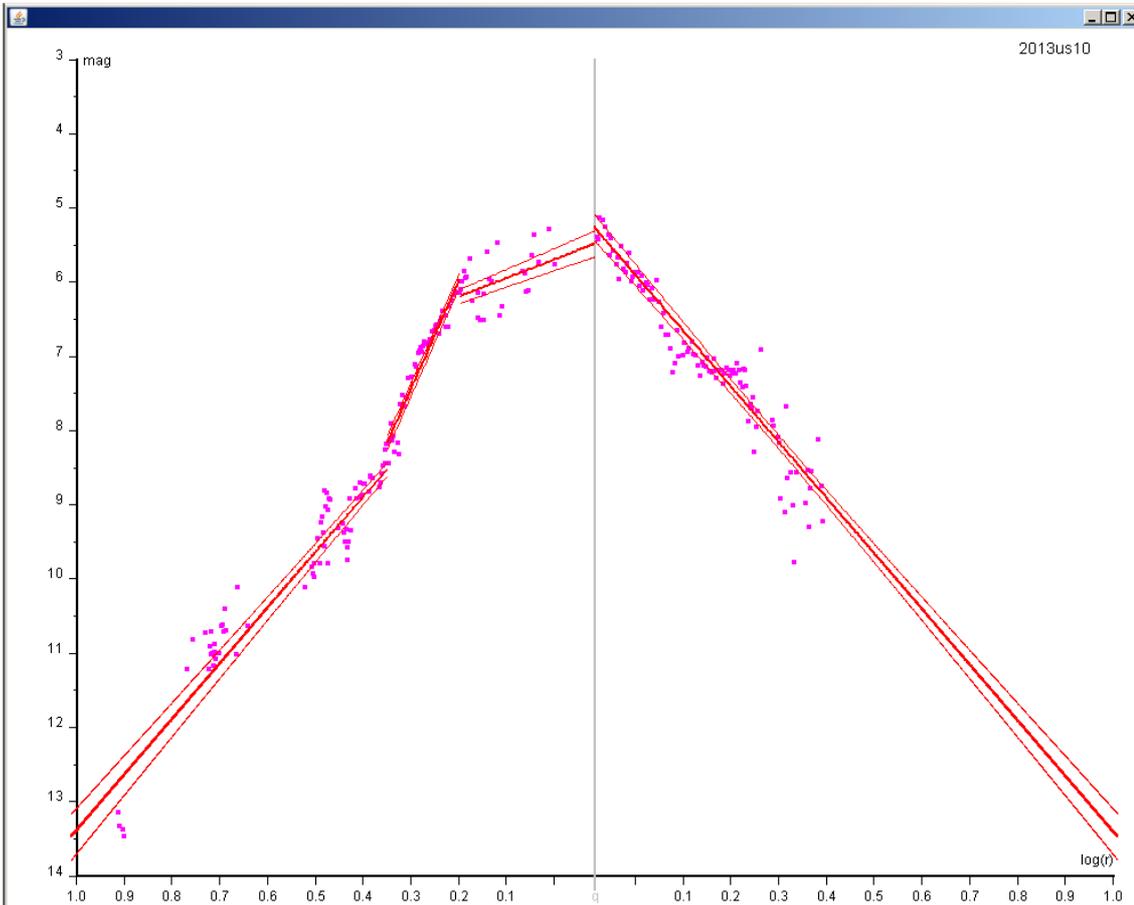


Alle Abschnitte:

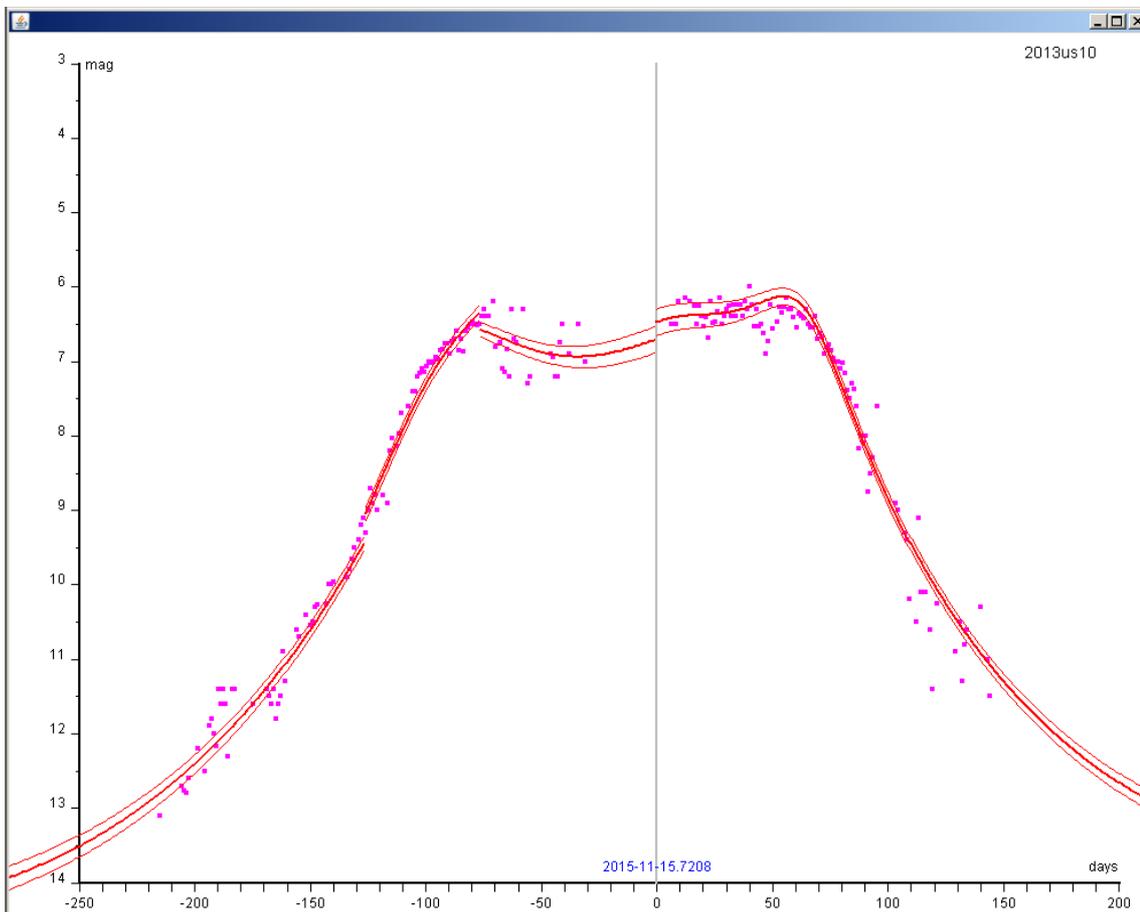
```
_999_126+059030
_126_076+031058
_076_000+057010
_000+999+059030
```

Diese müssen hintereinandergeschrieben werden:

```
... -s _999_126+059030_126_076+031058_076_000+057010_000+999+059030
```



Man sieht, dass die stückweise Berechnung unschöne Effekte hat, die die Kurvenstücke nicht aneinanderpassen. Das ist nicht nur ein Schönheitsfehler, sondern geht auch an der physikalischen Realität vorbei. Wenn die Kurve keinen Sprung macht, sondern nur einen Knick, dann sollte sich nur der Anstieg ändern und die Kurven ansonsten aneinanderpassen. Wenn es einen richtigen Sprung gibt, wie im Falle eines plötzlichen Ausbruches, dann ist dieses Verfahren aber das angemessene. Mit der `-s`-Option lässt sich auch eine geozentrische Kurve ausgeben (`-R8` wieder weg):



Die Mängel der stückweisen Berechnung lassen sich mit dem Lightcurve-Programm beheben, es ist aber etwas aufwendiger.

Stückweise Berechnung unter Erhalt der Kontinuität

Ich gebe es zu: Der Softwarekomfort dieser Berechnung könnte höher sein. Da ich das aber selten brauche, siegte bisher immer der innere Schweinehund. Und: Es geht ja. Die Berechnung ist allerdings auf drei Abschnitte begrenzt.

Um das Perihel herum gibt es keine Messwerte, so dass ich hier einen Sprung akzeptiere (ich habe ja nur drei Abschnitte zur Verfügung). Die Vor-Perihel-Kurve hat die drei bekannten Abschnitte, die du knickfrei und genauer berechnen willst:

- 1) Anfang - 26.5.15 (-172) $\log(r) \sim 0,35$, $r = 10^{0,35} = 2,2$
- 2) 26.5.15 - 31.8.15 (-76) $\log(r) \sim 0,20$ $r = 10^{0,20} = 1,6$
- 3) 31.8.15 - 15.11.15, Perihel

Diese Abschnittsgrenzen wurden visuell geschätzt. Es wäre viel besser, diese mathematisch zu bestimmen. Mit dem Lightcurve-Programm ist das möglich.

Zwei-Abschnitts-Modell

Zuerst wird die genaue Grenze der ersten beiden Abschnitte bestimmt, die um $r=2,3$ herum liegt. Hierzu läßt du mehrere Simulationen laufen, welche die Kurve in zwei Abschnitte betrachten. Diese Rechnung ist aufwendig und dauert ca. eine Minute:

```
Lightcurve -gk2013us10 -d250 -D200 -R8 -T 20150831 -2 2.3
```

Die Parameter:

- Grafik wird abgeschaltet, -g
- Es werden nur Messwerte bis zu Beginn des zweiten Knicks betrachtet, der in der Nähe vom 31.8. liegt (-T)
- Es wird ein Zwei-Abschnitts-Modell berechnet, dessen Knickpunkt zunächst bei 2,3 angenommen wird (-2)

Es erscheint eine andere Ergebniszeile:

```
SDD: 2.3, m8= 5.79, n8= 3.00, m0= 5.79, n0= 6.30, fqs=0.090481
```

Es geht hier um die Fehlerquadratsumme. Die Rechnung wird für alle Knickpunkte von 2,0 bis 2,7 ausgeführt (Optionen -2 2.0 ... -2 2.7) und die Fehlerquadratsummen notiert.

Dort wo, die Summe am kleinsten ist, liegt der mathematisch bestimmte Knickpunkt, berechnet als r:

```
r2      Fqs
2.3    0.090481
2.4    0.089920**
2.5    0.092319
```

Log(2,4) ist 0,38 - unsere Schätzung lag zwar etwas daneben, war aber recht gut.

Falls man nur zwei Abschnitte berechnen möchte, muss man aus dem Ergebnis von 2,4 die Anstiege entnehmen. Sie sind als n8 und n0 angegeben, der Wert n8 gilt für den äußeren Abschnitt, der Wert n0 für den inneren:

SDD: 2.4, m8= 5.94, n8= 2.90, m0= 5.94, n0= 6.10, fqs=0.089920

Sie sind 2,9 und 6,1 im Vergleich zu 2,9 und 5,8 zur knickbehafteten Simulation. Die Werte m8 und m0 benötigt man zum Zeichnen einer Grafik. Die Einzelheiten zur Zeichnung werden im nächsten Abschnitt erläutert: Statt drei Abschnitten wie dort sind eben nur zwei (m0-n0 und m8-n8) einzusetzen.

Drei-Abschnitts-Modell

Um ein Drei-Abschnitts-Modell zu berechnen, kann man zuerst den ersten Knickpunkt wie oben gezeigt bestimmen. Wenn dieser bekannt ist,, wird mit der zusätzliche Option -3 der geschätzte Wert des zweiten Knicks angegeben. Dieser ist $10^{0,2}$, also etwa 1,6 :

```
Lightcurve -gk2013us10 -d250 -D200 -R8 -T 20151115 -2 2.4 -3 1.6
```

Auch diese Rechnung wird wieder für eine Reihe von Knickpunkten ausgeführt.

Es ist auch möglich, keine Zwei-Abschnitts-Modelle zu berechnen und alles mit der Drei-Abschnitts-Variante in einem Rutsch auszuführen. Es sind dann mehr Rechnungen nötig, und die Rechenzeit wird größer. Wen das nicht stört, der kann alles in einem Schema erledigen.

Es ergibt sich in unserem Beispiel

```
R3      Fqs
1,6    0.090984
1,7    0.089092**
1,8    0.091337
```

Der beste Wert ist r3=1,7. wir lagen mit 1,6 wieder etwas daneben.

Benötigt werden die drei Anstiege, die der Ergebniszeile entnommen werden. Sie heißen hier n0, n3 und n8:

SDD= 2.40, FUD= 1.70, m0= 5.60, n0= 1.30, n3= 6.30, n8= 2.90 fqs=0.089092
m0= 5.60, n0= 1.30, m3= 2.72, n3= 6.30, m8= 5.95, n8= 2.90

Ich stelle diese Anstiege einmal den Werten gegenüber, wie wir mit der einfachen Methode berechnet haben.

```
      m0      n(alt) n(verbessert)
1) 5.86  2.95  2,90
2) 3.10  5.81  6,30
3) 5.69  0.98  1,30
```

Für eine Gesamtgrafik benötigen wird die m0-Werte für alle Geradenabschnitte. Diese stehen in der zweiten Ergebniszeile. Der „echte“ m0-Wert, welcher für den perihelnen Teil gilt, heißt auch m0. Der nächste heißt m3 und der äußere m8. Diese Werte liegen ganz in der Nähe derjenigen, welche wir mit der stückweisen Simulation erhalten haben, was ja auch logisch ist.

Du benötigst die Periheldistanzen in Tagen für die jetzt bestimmten Knickpunkte. Mit der Option -gL (Grafik abschalten und Liste der Eingangswerte, also

```
Lightcurve -k2013us10 -gL | more
```

Kann man die benötigten Daten ablesen. Der Sonnenabstand r steht in der dritten Spalte, der Tagesabstand zum Perhel in der letzten. Konkret:

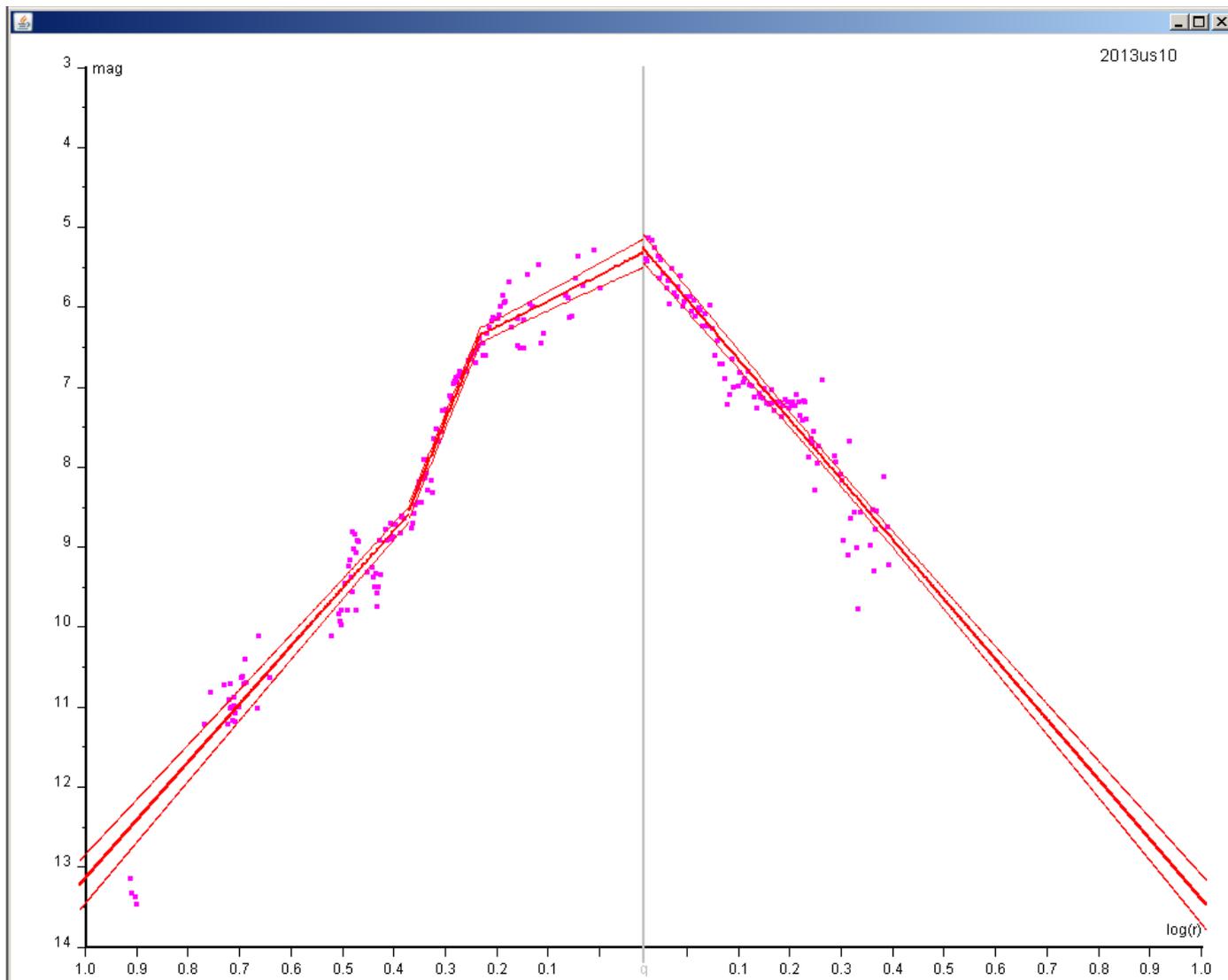
```
r/AE  δt/Tage
2,4   -135 (interpoliert)
1,7   -85
```

Du benutzt wieder die Option -s und baust die zu zeigenden Abschnitte nacheinander auf, also zuerst den ganz linken Teil:

Lightcurve -k2013us10 -d250 -D200 -R8 -s _999_135+059029
und dann anschließend:

_999_135+059029 ganz linker Teil
_135_085+029061 zwischen den Knicken
_085_000+056013 kurz vor dem Perihel
_000+999+059030 nach dem Perihel, gewonnen durch abschnittsweise Rechnung

Alles muss natürlich hintereinander geschrieben werden.

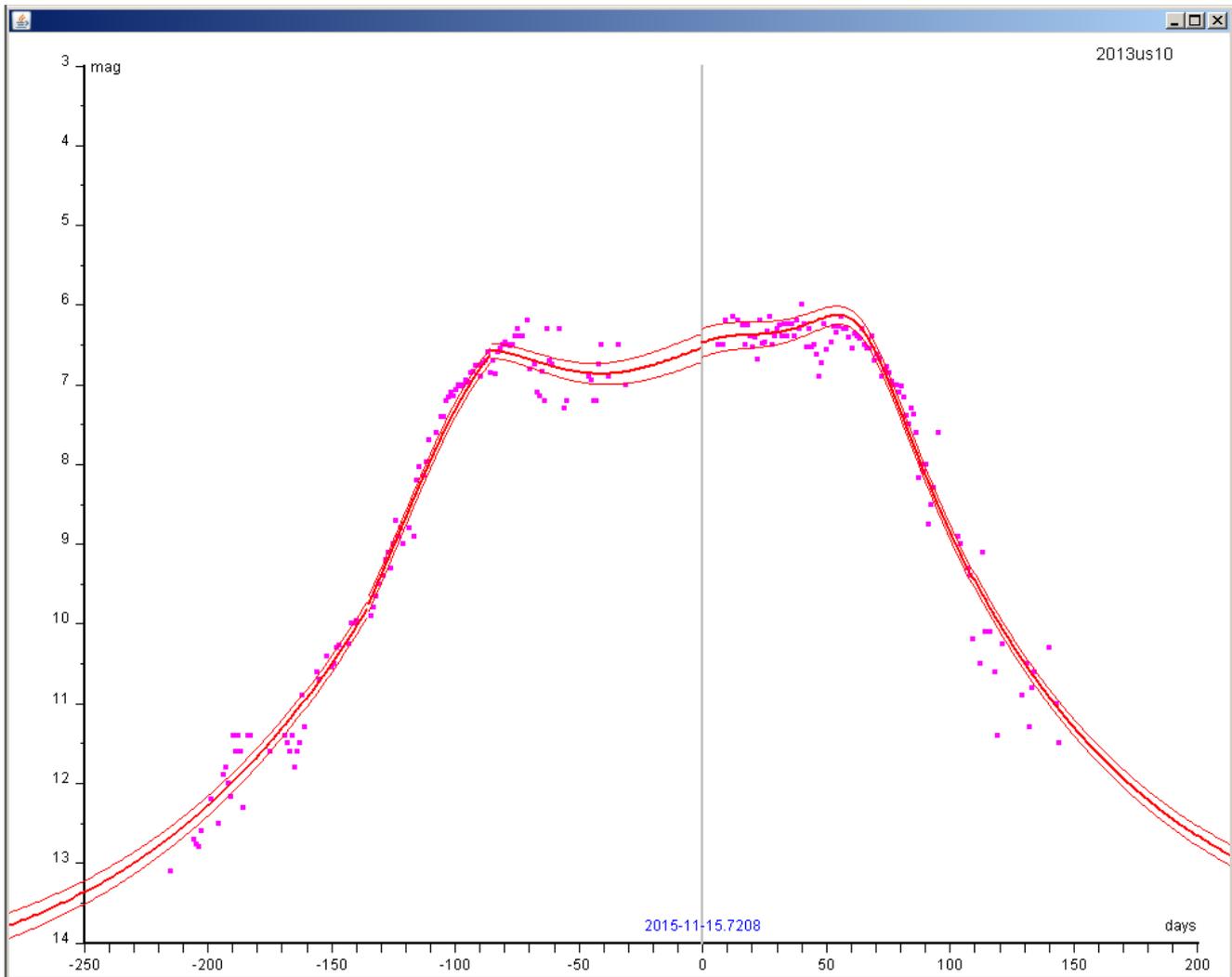


Geozentrisch, zum Schluss noch mal der komplette Befehl:

```
Lightcurve -k2013us10 -d250 -D200 ...
```

```
... -s _999_135+059029_135_085+029061_085_000+056013_000+999+059030
```

(alles in eine Zeile)



Sieht doch chic aus, oder?