

Master Journal

Di – 2. April

Nur rumgebastelt:

- Versehentlich Swics-Ordner gelöscht, wiederhergestellt.
- SVN durchsucht nach fehlendem libacepy.ancil:
svn+ssh://asterix/data/subversion/etph/swics/trunk/ionpositions/libacepy
Softlink gesetzt in Hauptverzeichnis: *ln -s [path]*
- Numpy und Scipy aktualisiert: *sudo pip install numpy upgrade*
→ Fehlermeldungen endlich weg
- Journal 'Osmo' runtergeladen, wieder verworfen, LibreOffice ausprobiert, verworfen, für Latex entschieden: An Diary-Template orientiert
- Dazu: in *newcommand* eingelesen, versucht aktuelles Datum ohne Update einzubauen, guten Beitrag mit tikz-Package gefunden,

Mi – 03. April

- Versucht, *itemize*-Listen anzupassen mit *enumitem*-Package → zu zeitaufwändig, mach ich später. TODO
- außerdem TODO: Todo-command von Stefan/Finn raussuchen

- Jetzt weiß ich, warum ich Dist3D nicht weiter gecheckt habe: Funktioniert nicht ohne AspectAngle-Phi und -Theta und ohne Testen verstehe ich nicht, was passiert. Ich muss also erstmal die Winkel einpflegen.
- TODO: Git aufsetzen (!!) und nochmal PyCharm checken
- Programmstruktur auf Papier gefunden. Yay, sehr hilfreich! *ulysses_traj.py* angefangen anzupassen an zusammengefasste Trajektoriendatei (*traj_data_pool.dat*).
- NEXT: Methode *sync_traj()* in *uswipha.py* anpassen, am besten alte M. einfach umbenennen (als Backup) und eine neue schreiben.

Do – 04. April

- Wollte schnell AA-Phi und AA-Theta einpflegen, da fällt mir auf: Die müssen ja noch berechnet werden aus den HG-Koordinaten. Die Funktion dafür gibts schon: *calc_asp_angles()* aus *ul_calc_traj.py*. Die braucht aber Erd-Koordinaten in HG-Koordinaten und die gibts noch nicht. Ich könnte die von der NASA-Seite holen, würde die aber lieber rechnerisch rausbekommen. Altes Problem rekapituliert: Erd-K. bekannt in HC: LAT immer Null, LONG unbekannt. In HG: LAT unbekannt, LONG berechenbar aus Ulysses-Datenprodukt. Ich brauche eine der unbekannten Größen! (Über Drehmatrix rückwärts rausrechnen: Geht nicht, weil ich keine kartesischen Koordinaten habe(?)) Nächster Versuch: Über SPE, SEP, ESP. → Hilft nicht. Jetzt pflege ich Erddaten von der NASA halt in *ulysses_traj* ein. Mal gucken, ob ichs rückwärts berechnen und damit verifizieren kann...

Sa – 06. April

- Begonnen, die NASA-Daten einzuladen und die Winkel in RTN auszurechnen. Mit Testfunktion kommt leider nicht das Richtige raus. Fehlersuche.

- Fehlersuche ergab: Ich hatte den Long-Shift übermutig aus der Koord.-Trafo für die Erde rausgenommen. Muss aber natürlich drin bleiben, weil ja auf RTN-Koord. des SC (dessen Daten mit dem Shift belastet sind) transformiert wird.

So – 07. April

- Was für Daten sollen jetzt eigentlich synchronisiert werden mit `sync_traj()`? RTN-Koordinates fürs SC sind ja nicht besonders aussagekräftig (weil Null in T und N). Ich nehme wahrscheinlich die verschiedenen AA's und HG-Koordinaten. Kann man ja schnell anpassen.
- Next: Das machen (Anpassung in *ulysses_traj*), dann *Dist3D* damit aufrufen und gucken, was als nächstes fehlt.

Di – 09. April

- AspectAngles werden geschluckt von *Dist3D*. Nächstes Problem: Detektor-ID! → Achja. Ist ja schon drin in den Datenprodukten, als *det*. Ich setze jetzt neues Dist3D speziell für Ulysses SWICS auf (Skript aus *SWICS/trunk/ace_dbd/* nach *Ulysses/swics/software* kopieren).
- Next: Brauche passende *getvelocity*-Methode, dafür muss ich rausfinden, ob Ulysses/SWICS dieselben EpQ-Spannungssteps hat wie ACE/SWICS

Mi – 10. April

- Rausgefunden, in welcher Einheit Werte in *getvelocity* sind: $\frac{keV}{e} = kV$!

- Entsprechendes Werteintervall nicht im SWICS-Paper gefunden. Im ACE-SWICS-Paper geguckt: Da steht zumindest der obere Wert von $100 \frac{\text{keV}}{\text{e}}$ (Main Channel). Die untere Grenze ist allerdings 0.49 anstatt 0.61 (in *ancil.py*). Habe die Werte mal geplottet und in der logarithmierten Darstellung ist ein deutlicher Knick in der Geraden! Interessanterweise sind in dem Paper (TABLE1) auch zwei Werte für Step Size angegeben. Erstens macht eine Step Size ja nicht so viel Sinn, wenn die Steps logarithmisch verteilt sind. Also habe ich mir *exp()* von den Differenzen angeguckt: Tatsächlich zwei konstante Werte! Aber statt 1.0744/1.0365 bekomme ich 0.071 und 0.144. Hm...

Mo – 15. April

- Mit Lars gesprochen. Die EpQ-Steps wurden aus den Housekeepingdaten von ACE ausgelesen. Und bei Ace sind sie in der Tat auf zwei verschiedenen Steigungen. Das ist bei Ulysses wahrscheinlich nicht so. Für U. (64 statt 60, aber kleinere Range) hat Thies schon damit gearbeitet, die muss es also irgendwo geben, Lars schaut mal nach. In den Raw- oder PHA-Daten haben wir sie nicht gefunden. Bis dahin nehme ich die Funktion von ACE als Dummywerte.

Mi – 17. April

- Neuen Softlink (libacepy) gesetzt für Dist3D. Jetzt muss ich mit Dummy-GetVelocity() das Skript verstehen.
- D.vspace gibt an: *welche Geschwindigkeitseigenschaften(Richtung, Betrag) muss ein Teilchen gehabt haben, damit es hier gemessen wurde?*
- Ich mauschel mich da durch, passe alles entsprechend an. Problem: Es gibt in *ds* Datenprodukten (uswipha-Instanz) vier(!) Detektor-ID-Werte: 0,1,2,3. Ich erinnere mich vage, dass ich früher mal wusste, was das war. Muss ich rausfinden und abfangen, weil dist3D natürlich nur

was mit drei Werten anfangen kann.

Achso, abfangen klingt kompliziert: Einfach Maske setzen!

- Unterwegs (in der Kladde) gelvelocity()-Funktion für Ulysses gefunden! Yay! (`/home/asterix/fischer/PUI/Ulysses/swics/software/libulpy/uswiutils.py`) (Lars: Eigentlich noch ein Step mehr: 85 oder so. Aber der ist kaputt)
- Zum Detektor-ID-Problem habe ich nur was Knappes auf S.45/46 gefunden. Leider nicht das Endergebnis und Lars' Aussage zur damaligen Analyse. Ich suche jetzt das Skript, das die jeweiligen Plots erstellt, an denen wir damals was gesehen haben:
- Angefangen, `pui_ulysses.py` durchzugehen. Polarplot() funktioniert nicht. Woher kommen die drei Masken? Muss ich nach Ostern weiter durchgucken...

Mi – 24. April

- Aus Osterurlaub wieder da. Jetzt muss ich wirklich loslegen – noch 3 Wochen bis zum Vortrag.
- Einiges mal überflogen. Grobe Struktur: Alten Vortrag zusammenfassen: PUI → was erwarten wir für eine VDF? Echte VDF sieht aber anders aus. Erklärungen? Messfehler → Übergang SC-frame auf SW-frame bzw. (?) Übergang 1D auf 3D (diesen Zusammenhang muss ich mir klar machen!!). Dann fange ich an, Ulysses/SWICS zu erklären und was bisher mit der Trajektorie etc. gemacht wurde.

Do – 02. Mai

- Rausgefunden, warum in polarplot() die Masken schon gesetzt waren: In diy→ gibts das Skript `load_pui_ulysses.py`, das genau diese Masken setzt.

- Detektorproblem: Dort werden Detektoren 0,1,2 geladen, d.h. Nr. 3 ist evtl. der, der über ist?
- PATHPROB: Zum Ausführen von `load_pui_ulysses.py` muss ich `uswipha.py` laden, das füge ich dafür zum `python-sys-Path` hinzu. Dasselbe mit `pui_ulysses.py`
- Habe jetzt endlich alle alten Sachen (`AnimatePolarplot`) zum Laufen bekommen. Evtl. könnte man die Animation auf Lars' Kollimator umbauen. Aber nur ganz evtl.

Di – 04. Juni

Dokument aufgesetzt, Bibdatei angelegt, Zusammenschrieb auch damit verbunden. Für Biblatex entschieden.

```
\usepackage[backend=biber,style=authoryear,maxbibnames=2,natbib=true]{biblatex}
```

Dann mit `citep` (alles in Parenthesis) oder `citet` (im Fließtext) zitieren.

Weiter: Infos von losen Zetteln in Zusammenschrieb schreiben.

Mi – 12. Juni

Python3 zusätzlich zu Python2 installiert (weil Max' Programm in Python3 geschrieben ist):

- Python 2.7 und Python3 irgendwie vorhanden.
- `sudo apt-get install python3-pip`
- Jetzt kann ich mit `pip3` Python3 erreichen und mit `pip2` Python2 (Nur `pip` bezieht sich auch auf 3)
- Problem: `ipython` öffnet immer `ipython` für Python2. `pip3 install ipython` hat nicht geholfen.

- Sebastian hat geholfen: Ipython3 ganz normal installieren: *sudo apt install ipython3* . Jetzt kann ich entweder *ipython* oder *ipython3* aufrufen! Jetzt noch alle Pakete mit *pip3* installieren

Anderes:

- pip immer nur lokal installieren, also ohne *sudo*!
- TK: GUI-Programm in C++ geschrieben, für Python über Interface tkinter. Nicht über pip installierbar, denn pip soll nur pure Python Pakete installieren. Also installieren mit *sudo apt install python3-tk*

Fr – 12. Juli

Bei Lars gemeldet. Er hat den Kollimator und Dist3D weiter entwickelt.

- Zeigt Ergebnisplot (das ist auch mein Ziel mit Ulysses), bei dem entlang der x-Richtung in die Verteilungskugel geguckt wird (schnittweise). Man erkennt innen was in der Mitte (Innere Quelle?) und außen die Schale!
- phasenraumgewichtet (wie oft habe ich überhaupt in eine Richtung geguckt, d.h., wie groß ist die Wahrscheinlichkeit, dass ich ein Teilchen hätte messen können?) und normiert (auf max)
- Das sind ACE-Daten, Langzeit, alle vsw. Ziel ist, wenn alles läuft, nach unterschiedlichen Magnetfeldkonfigurationen und vsw zu filtern und evtl. Unterschiede zu finden.
- Bsp: Bfeld einschränken auf theta: kP und phi: -90° bis -70° und 90° bis 110° . Zeichnung s. Kladde S. 83. Erwartung wird irgendwie nicht richtig erfüllt. Filter auf kleine vsw: schon eher.
- Evtl. gibt es noch ein Problem mit dem Aspect Angle, denn die Population ist manchmal wohl auf der falschen Seite (nämlich links)
- das sind 12-min-Daten. Eigentlich müsste man Minutendaten nehmen, denn: Gyroperiode der Teilchen ist etwa eine Minute. Alle Änderungen des B-Feldes, die auf kürzeren Zeitskalen stattfinden: PA-Bestimmung nicht sinnvoll, weil keine volle Gyrobahn. Deutlich längere Zeitskalen: auch nicht sinnvoll, weil Verteilung mitdreht (?)

Zu den Daten:

- Etwas veränderte Ordnerstruktur. Ich muss neu auschecken:
`svn+ssh://asterix/data/subversion/etph/ACE` (Instrumentelles, die Dist3D die hier liegt ist eine alte Version) bzw. `.../etph/PUIs` (datenspezifisches, neuere Dist3D, die zu PUIs passt)
- ACE-Datensatz zu Probieren auf `data/etph/berger/mq...`
Dort liegen HE2+-gefilterte Daten aus 12 Jahren. Ich muss zusätzlich wHE einschränken auf 0.9 bis 10 (das ist w ganz einfach in 1D, Lars hat auch gezeigt, wie das in 3D abgebildet wird, Todo: überlegen!) und eine Maske von Hand legen (s. Zeichnung Kladde S. 83)
- Datensatz umfasst ca 800000 PHAs+. Nach Maskierung sollten etwa 400000 übrig bleiben.
- Pi mal Daumen ist das eine Messung pro 12-Minuten-Zyklus! (120 Cycles pro Tag mal 365 Tage mal 10 Jahre = 400000 Cycles)
- → Wir können nur nach Signaturen suchen, die sich langfristig oder regelmäßig ändern. Aussagen von Zyklus zu Zyklus sind überhaupt nicht sinnvoll!
- Dist3D nimmt neue Parameter:
 - `nrs_perp` ist Anzahl der Punkte an der kurzen Seite (da reicht eigentlich einer, denn die Breite ist klein gegen den Sektorsweep! Wäre nur praktisch, wenn man eine hübsche Abbildung vom Kollimator machen will)
 - `nrs_perp`: 3
 - `nrs_sec`: wie viele Steps in einem 45-Grad-Step
 - `nrs_epq`: wie viele Einzelsteps innerhalb der Breite eines EpQ-Steps (nominelles Geschwindigkeitintervall)
- Lars' Rechner kam mit `nrs_epq = 3` schon an seine Grenzen. Das liegt an `calc_w`, weil da noch eine zusätzliche Dimension (Sonnenwindgeschwindigkeiten) dazu kommt. Das will er jetzt umbauen.

Weiter: Neu auschecken, Kopien anlegen und alles zum Laufen bekommen

Do – 18. Juli

- SWICS_collimator durchgeschaut, alles verstanden
- Jetzt Dist3D durchgucken. Zum Laden von Lars' Probedatensatz: .../PUIs/ace_swics_3d.py

Do – 18. Juli

- Nachtrag zur Pip2-Installation (pycwt):
pip install -user pycwt

Do – 25. Juli

- Nils: Schnell mal was plotten:
`figure(), plot(d.data["swt"])`
Histogrammieren:
`figure(), hist(d.data["swt"], arange(0, 300, 1))`
- Bin bei Funktion `def _calc_phspeff_wgt(self):`.
Phasenraumvolumen nicht verstanden. Warum komme ich mit Integration nicht auf dasselbe Ergebnis, warum ist es nicht egal, welchen Winkel ich als welche Integrationsvariable nehme...?
- Geometriefaktor auch nicht verstanden. Sullivan-Paper: woher kommt der Kosinus?

todo

Geometriefaktor ...

Fr – 26. Juli

- `_calc_sw_bulk_properties()` nicht gecheckt, warum passiert das da mit der Alfven-Geschwindigkeit und warum findet man das nirgendwo? Evtl. bei Marsch nachlesen...?

Alfvengeschwindigkeit

Allerdings ist die Funktion auch nicht so wichtig atm, glaube ich, wird nicht benutzt?

Lars:


- Datenprodukte ACE:
 - dsw* – Sonnenwinddichte
 - tsw* – Temperatur Sonnenwind
 - swt* – sector weight, ähnlich wie Base Weight Rate. Erklärung im Zusammenschrieb
- Funktionenwald in `dist3D_pui`: Die aktuelle, wichtige ist `calc_w3dspecs`! Die funktioniert auch!
 - Erklärung in 3 Schritten:
 - Unten ganz normal histogrammieren mit diversen Weights
 - `twts`: ? (`wgts*swgt`)
 - Loops darüber: Wie oft habe ich in eine Richtung geguckt. Um den Dichtegang rauszunehmen, der daraus resultiert, dass man häufig entlang der Radialen guckt und zu zeigen, dass darunter ein ähnlicher Dichtegang liegt, der nichts damit zu tun hat
 - zurückgegeben werden dann `norm_arr` (proportional zur absoluten Messzeit, „wie oft hat das Instrument in eine Richtung geguckt“; Erwartung: häufig entlang der Spinachse, also radial) und Histogramm ungewichtet.
 - Plot erzeugen:

```
norm_arr[norm_arr==0]=1. Null-Werte auf 1 setzen (warum?)
H2 = H/norm_arr Normierung durchführen
wbins = arange(-2.-2.01.0.2)
pcolormesh(wbins,wbins,H2[10,:,:].T) 10. Slice. Bei 11 geht
man weiter nach vorne usw... ( rausfinden, in welche Richtung –
DONE s. Anfang September
```
- Die anderen eher sekundär (funktionieren auch nicht, weil die noch auf einem anderen Kollimator arbeiten?): Da hat L. nur mal schnell was ausprobiert oder der Versuch bzw. das auf 1D runterzubrechen (`calc_wpecs2()`) oder für jedes w die PA-Verteilung zu zeigen (`calc_muspec()`)
- Plan: Quick and Dirty für Ulysses umschreiben, solange Lars im Urlaub ist. Lars denkt, dass fast nur der Sec0-Versatz zum Sunpulse-Analyser geändert werden muss. Hiefür soll ich diesen Versatz als variabel lassen, sodass man den übergeben kann bei Funktionsaufruf und

Slices

ein bisschen ausprobieren kann. Man erwartet den Protonen-Bulk (Bereich ziemlich starke Intensität) relativ radial. Auf dem Wege kann man schon einiges ausschließen. Das kann ich auch schon ohne die Gewichtung sehen.

Mi – 31. Juli

- Habe jetzt Dist3D für ACE durchgeguckt und das meiste gecheckt.
- Umbau für Ulysses. Neue Ordnerstruktur. Bisher liegt alles in `old_stuff` 
- Muss ich an `swics_collimator` was ändern?

Do – 1. August

`swics_collimator` durchgeguckt. Liegt inzwischen als `swics_collimator_ulysses.py` in `./PUI/Ulysses`.

- Rausgefunden, dass das Umbrechen der Zeilen *Wrapping* heißt und es geschafft, das nachträglich in Pycharm für die Kommentare zu machen: File – Settings – Code Style – *Hard wrap* einstellen und *Wrap on typing* ankreuzen. Außerdem unter Code Style – Python – Wrapping and Braces – *Ensure right margin...* ankreuzen. Außerdem hat Edit – Fill Paragraph geholfen.
Danach reformatieren!
- In der Kollimatorklasse habe ich eigentlich nur die Dokumentation angepasst und ein paar Checks gemacht. Das KS ist bei ACE ja anders (GSE) als bei Ulysses (RTN), aber das ist ja schon bei der Berechnung des AA mit eingeflossen.
- `offset_sp` ist jetzt der variable Versatz zwischen Sunpulse-Sensor-Guckrichtung und SWICS Z-Achse. (Bei ACE sind das 135 Grad.)

Jetzt geht es an Dist3D.

- Datenloader uswipha zum Laufen bringen: Daten wurden von `/data/etph/...` nach `/data/projects/...` verschoben. Die Lösung kann so einfach sein.
- `swics_collimator_ulysses` muss schon ein `FieldOfView` zum Kollimator zufügen!
- Changes:
 - *did* → *det*. Welchen Detektor muss ich jetzt nochmal rausschmeißen? Beobachtungen auf 3 Jahren Daten (1993,94,95):
 - * Det3 wird kaum angesprochen! (von 43 Mio PHAs nur 1225 auf Det3)
 - * Wenn ich nur Ech unter Threshold nehme (26 Mio; = Doubles), landen fast alle in Det0. (Det1: 26, Det2: 88, Det3: 164)
 - * → Ich würde denken, dass Det3 einfach Müll ist.
 - * Das passt auch zum Kommentar vom 2.Mai. D.h., ich muss eine Maske setzen, bevor ich in Dist2D reingehe. Soll ich eigentlich auch Doubles benutzen, macht das Sinn? DONE: Lars: Nein, die bringen uns nichts, weil die keine Detektorinfo tragen (natürlich...)

Det-ID

Doubles

Morgen: Schauen, ob ich `pui_ulysses` benutzen kann. Sonst anderes Main-Programm bauen (Daten laden, Masken setzen, ...).

Fr – 2. August

- Pycharm-Problem vorab: Ich kann den Debugger irgendwie nicht bei meiner momentanen Datenstruktur verwende. Er findet die ganzen Pakete nicht...
- Weiter Veränderungen in Dist3D:
 - Das Skript heißt `load_pui_ulysses`. Benutze ich jetzt.
 - Funktion `getvelocity_allsteps` zum Umrechnen von EpQ-Steps in echte Energien pro Ladung bzw. in Geschwindigkeiten: s. Kommentar vom 15.04. Ist bei Ulysses SW eine andere als bei ACE. Die richtigen Werte müssen wir noch raussuchen. Ich benutze

getvelocity

jetzt erstmal die ACE-Funktion. Später muss dann auch die Anzahl der Steps angepasst werden! DONE: s. 28.8.

- Ist Delta EpQ dasselbe, 3%? DONE: Ist 5%, s. Gloeckler. Δv sind dann 2.5%, s. Kladde S. 84. Delta EpQ
- Geometriefaktor ist wahrscheinlich derselbe, oder? DONE: Joa Geomfak
- In `_calc_vspace_` wird bei ACE die Eigengeschwindigkeit nur in der y-Komponente abgezogen. Das muss bei Ulysses aber ja nicht so sein (?). DONE: Nein, geregelt. Eigengeschwindigkeit
- Ich bin gerade verwirrt von den Koordinatensystemen. GSE hat x-Achse ja vom SC zur Sonne positiv und später wird das umgedreht, damit der SW grob aus positiver Richtung kommt. RTN eigentlich andersrum. Soll ich das so beibehalten? Dann muss ich nochmal zurück zum Kollimator. Denn das FOV ist dann ja in negativer R-Richtung.
Ich hab jetzt erst einmal das RTN umgedreht (also eigentlich nur die Konvention in die Kommentare geschrieben), jetzt ist alles so wie beim GSE. Es ist ein bisschen inkonsistent, dass ich den AA ab der negativen R-Achse berechne (aber das habe ich irgendwie ja immer schon so gemacht. Jetzt auch in der `calc_asp_angles`-Funktion deutlich kommentiert). Jetzt tue ich halt so, als wär das die positive Achse, dann muss ich nichts ändern bei FoV. Aber vielleicht wird das problematisch, wenn ich die Eigengeschwindigkeit einrechne, die ja evtl. in normalen RTN-Komponenten angegeben wird? DONE: Hoffentlich richtig gemacht Koordinatensystem
- In `_add_3Dv` musste ich nochmal Datenproduktnamen ändern, außerdem *det DCR to det 0* rausnehmen. Bei ACE gibts es offensichtlich *det = 7*- Zuweisungen.
Außerdem geht der EpQ-Step über 60 hinaus. Das crasht, weil ja die `getvelocity` von ACE mit 60 Steps benutzt wird. Ich wollte Daten erstmal einfach vormaskieren, bekomme das aber nicht hin. Bei der Übergabe von `d` wird die Mastermaske ignoriert! Ich suche quasi die `subset`-Funktion aus `dbData` – nur „live“, also ohne Wegschreiben.

- Bin jetzt den Umweg gegangen über `save_subset` und `load_subset`. Damit schreibe ich ein Subset von den gefilterten Daten (Det von 0 bis 2 und epq von 0 bis 59) in eine Binärdatei und lese sie wieder ein...
Da muss es aber doch noch einen anderen Weg geben?

Subset

Nachtrag: Ich habe EpQ schließlich von 1 bis 58 gefiltert, weil die ACE-Daten auch so vorliegen. 0-59 hat für `get_velocities` funktioniert, aber nicht fürs Synchronisieren der Efficiencies.

- Efficiencies soll ich ja erstmal von ACE nehmen. Da die nur für die Triples gelten (?) filtere ich die Doubles jetzt auch mal raus:
`d.set_mask('Master','ech',12,120,reset=True)`
DONE: 05.09.

Efficiencies

- Für Ulysses gibt es das Datenprodukt *swt*, das Sectorweight, nicht. Lars fragen bzw. selbst nachgucken, evtl. wird das nur über uswi-pha nicht eingeladen. Bis dahin setzte ich alle Gewichte auf 1 (in `calc_w3dspecs`). DONE

Sectorweight

- Es läuft durch!! Ich kann das Spektrum erzeugen!! Jetzt mal mehr Daten einladen...
- Ständig Memory Error...
200 Tage sind das Maximum.
- Überlegung, warum jedes PHA-Wort überhaupt 9 FoV-Punkte hat: Weil man nicht weiß, wo der Sektor getroffen wurde. Punkt 0 hat z.B. eine ganz andere w-Signatur als Punkt 5. Alle Möglichkeiten müssen betrachtet werden.
Aber: Macht eine w-Auflösung von 0.2 überhaupt Sinn?
Die größte Differenz im Betrag von w haben immer der erste und der letzte Eintrag.
Differenzen werden kleiner für größere EpQ-Steps. Stimmt nicht ganz: Peaken bei einem mittleren EpQ-Step (abh. von vsw)
WARUM verändern sich die Differenzen für verschiedene Sektoren bei keinem AA?? → Weil irgendwie der AA doch nicht Null ist. Was geht da schief?
ToDo für morgen:
 - Methode basteln, um schnell irgendwelche Sachen zu plotten, z.B. wSpace. Das ist ja nicht mehr wirklich eine Sache in SwicsCollimator, weil ich mir ja mehrere Kollimatoren angucke.

- Methode schreiben, mit der ich durch die pcolormesh-Dinger loopen kann, um mir ein besseres Bild zu machen.
- Variablen SP-Offset durchführen und das verstehen.
- Oben weiter machen: Wollte eig. nur sicher gehen, dass sich die Diff mit den Sektoren nicht ändern und mit den Detektoren schon! (Tut es das dann?)

Di – 26. August

Habe jetzt nach langer Pause nochmal neu angefangen, Dist3D durchzugucken.

- `calc_w3dspecs()`:

- Was ist mit *d00*? Ich habe nur *d90*. Ist das egal? Hab das jetzt erstmal einfach auf *d90* geändert. Ich frage mich, wie das vorher durchlief...? DONE: Wird weiter oben erzeugt d90, d00
- Warum suche ich eigentlich nur eine Kombi pro Tag? Warum histogrammiere ich nicht alle vorkommenden Kombis?
 ⇒ Verstanden: Das sind nur die Bingrenzen. Warum die gerade so gewählt werden, ist aber irgendwie trotzdem random? (Anne, das stimmte nicht)
 AH! Alles falsch! Ich nehme die Uniquewerte, weil ich zum selben Zeitpunkt mehrere Messwerte habe. Ich will aber nur jeden Zeitpunkt einmal! Die Bingrenzen habe ich als Argument mitgegeben.
- Wozu `ivoffset`? ivoffset

- Ich verstehe immer noch nicht so ganz, was ich da oben genau verstehen wollte – warum soll sich die w-Diff. wann ändern?
 Aber kann es sein, dass ich die ersten Einträge beim AA genommen habe und das gar nicht 0 war, sondern eben der kleinste in diesem Zeitraum gemessene...?

Mi – 27. August

- *d00* wird oben berechnet! (in der `init` von `Dist3D`)
- Verwirrung: Warum sind die Werte des `wspace` (Betrag) *immer* unterschiedlich? Bei der Konstruktion werden die doch auf 1 normiert und dann mit *einer* Akzeptanzgeschwindigkeit malgenommen...(alle Sektoren, alle Detektoren)?
Gecheckt: FoV vom Collimator hat immer den Betrag 1 im Betrag.
Gecheckt: Bleibt auch so für verschiedene AA (in `Dist3D`)
Lösung gefunden: Die Unterschiede kommen rein, wenn man die Eigengeschwindigkeit betrachtet!
...Und natürlich, wenn man beim Umwandeln in `w` die SW-Geschwindigkeit nur in x-Richtung abzieht. Das ist ja dasselbe Problem, die Beträge SIND dann halt nicht mehr gleich. (Und das ist auch der Fall, wenn keine Eigengeschwindigkeit mit reinkommt!)

Abschließend habe ich festgestellt, haha, dass die `w`-Werte für Detektorpunkte innerhalb einer Det-Sek-Kombi in allen(?) Fällen eine maximale Differenz unter 0.5 haben (Details: Die max. Differenz liegt bei mittleren `EpQ`, rutscht mit größeren `vsw` etwas weiter zu kleineren `EpQ`-Steps (evtl. mal überlegen, warum...), ändert sich vom Wert aber kaum).

Damit schließe ich das Thema erstmal ab, wird bestimmt bei der Fehlerbetrachtung interessant... (Da muss man dann auch die Breite im `EpQ`-Step und die Variabilität in einem einzigen Step betrachten)

Jetzt erst gesehen: Lars' `w`-Bins sind aber nur 0.2 breit... (`nrs_sec`)

Jetzt: Sunpuls-Angle variabel übergebbar einbauen. hatte ich schon gemacht.

Morgen: prüfen, warum das `w`-Spektrum nicht einfach gedreht ist, sondern komplett anders aussieht mit unterschiedlichen SP-Offset-Angles.

Do – 29. August

Kurze Funktion geschrieben, um das Plotten des `w`-Spektrums zu automatisieren.

To Do:

Colorbar und passende Range. Wie beim Polarplot muss ich überlegen, ob ich auf einzelnen Steps oder auf alle Werte normieren möchte.

colorbar

Animierte Funktion schreiben, mit der ich durch die Slices klicken kann.

Durchklicken

Duncan gefragt: Stichwort *ClickEvent*, Figure damit verbinden. Minimalbeispiel suchen: *mpl.canvas.connect*. Nur machen, wenn das einigermaßen schnell geht...

Fehlersuche, warum offsetSP das Spektrum nicht nur dreht, sondern insgesamt verändert (habe einheitliche viridis-colormap. Die Range ist auch ganz unterschiedlich...):

- Test mit offset 135 Grad und 45 Grad (90 Grad Versatz)
- v-Space ist einfach gedreht um 2 Sektoren. W-Space auch.
- Die den PHA-Worten zugeordneten v- und w-Werte unterscheiden sich, auch wenn man ein Wort mit dem entsprechenden sektor sucht: Denn dann ist ja der EpQ-Step, Detektor etc. anders.
- Überlegung: Sollte sich das über viele Daten nicht trotzdem ausgleichen und in etwa ähnlich aussehen? (Ich glaube bei 200 Tagen sieht man es scho etwas!)
- Und wie beeinflussen die Anzahlen der verschiedenen Steps (para,perp,epq,sec) das ganze? Test:
 - Nur wenige (20) Tage einladen, damit ich mit der Anzahl hochgehen kann
 - Am meisten Einfluss haben nrs_perp und nrs_para, die *weichen* das Spektrum etwas auf.

Wollte mir überlegen, wie das Spekturm bei einem falschen Offset aussähe, keine Ahnung. Wollte mir das angucken für verschiedene Offsets. Dafür die Maske wegnehmen, die auf He+ einschränkt. Das funktioniert irgendwie nicht. Das Quadmesh aus der Plotfunktion sieht immer gleich aus. Morgen untersuchen.

Außerdem: Kann es sein, dass ich das falsche w_He einschränke?

- w_He1+2 ist ein Datenprodukt, das in Dist3D zugefügt wird. Das sollte gleich sein mit dem wHe, das sowieso schon in den Ulysses-Daten ist – allerdings erst dann, wenn die richtigen EpQ-Umsetzungen da sind. D.h., man könnte die evtl. finden, wenn man guckt, womit wHe auf der uswipha-Ebene berechnet wurde.
- Habs gefunden, in uswiutils. `getvelocity`-Funktion angepasst, Steps angepasst (64 bei Ulysses), aber EpQ-Filter muss trotzdem auf 58 Steps bleiben, weil immer noch die Efficiencies und die Sektorgewichte fehlen. Delta $v / \text{self.vels}$ auch angepasst. Alles DONE
- Jetzt: Loopfunktion schreiben
- Schon fertig, das hat super geklappt! Ich habe das Spektrum zum Plotten in eine Klasse (WSpec) ausgelagert. Man kann jetzt durchslicen und auch drehen (auf Tastendruck oder über Funktion). Als nächstes möchte ich eine gute Anzeige, die angibt, wo ich gerade bin – vielleicht auch eine entsprechende Skizze...? Die Colorbar vielleicht auch noch anpassen, damit man mehr sieht und Funktion, auf einzelnen Slice zu normieren...?

efficiencies

Mo – 2. September

Irgendwie scheinen mir die Schnitte komisch. Nochmal checken, wie ich transformieren muss.

Trafo w-Spec

DONE:

- Wenn z mit in der Ebene: transformieren (weil die *letzte* Dimension von pcolormesh standardmäßig auf die x-Achse geplottet wird. Das liegt einfach daran, in welchem Format der geslicte Array vorliegt.)
- Wenn y mit in der Ebene: x-Achse umdrehen (damit wir in positiver x-Richtung gucken)
- Die Ebenen sind jetzt:
 - x-Schnitt: oben pos. z, rechts neg. y, Blickrichtung von Sonne zu SC

- y-Schnitt: oben pos. z, rechts pos. x. Blickrichtung: rechts geht zum CS, links zur Sonne
- x-Schnitt: oben pos. x, rechts neg. y, Blick von oben auf die Ekliptik

Textfeld/Annotation unabhängig von Axes (relevant, wenn man Axes z.B. umdreht): `ax.text(..., transform = self.ax.transAxes)`

Colorbar-Range updaten: Mit `update_bruteforce`.

Aha.haha. Habe in sämtlichen Funktionen vergessen, durch den norm-Array zu teilen.

Ändert das was? Sind da immer noch komische Signaturen? He ausschneiden, Offset testen, Lars dazu holen. SW rausfiltern, Unterschied `min_wHe` angucken

Di – 3. September

Range weiter angepasst: Jetzt immer scientific Notation.

Color_norm 'all' macht echt nur Sinn, wenn man gaaanz wenige Tage einlädt, sonst ist die Range zu groß und man sieht nichts.

Mi – 4. September

Hatte ein Problem, auf `dbData` ein Histogramm zu erstellen. Problem gelöst: Colormap *spectral* wurde in meiner Matplotlib-Version abgeschafft. Ich glaube, *nipy-spectral* sieht ähnlich aus. *Spectral* ist pastelliger und mit weniger Range. Sollte Lars davon berichten. Habe im Skript `_dbData.py` in der Klasse `plot_properties` die `contourcolor` geändert. Jetzt läuft.

Besprechung Lars:

- Problem Subset: Eigentlich arbeitet Dist3D ja nur auf d, d.h. die Masken sollte alle bleiben. Wird wahrscheinlich nur überschrieben/resettet in Dist3D.

checken \Rightarrow verstanden, s. nächster Tag. Ist erstmal egal.

Masken anpassen

- Efficiencies: Gibt es nicht direkt für Ulysses. Da kann ich mit den

efficiencies

ACE-Werten vergleichen und dazwischen interpolieren – denn es geht nur um den groben Gang, wenn man EpQ-Steps an verschiedenen Enden vergleichen will

- Sektorgewichte: Gibt es wahrscheinlich nicht in den Ulysses-Daten. Base Rate Weights nehmen, die sind halt dasselbe, nur über einen gesamten Spin gemittelt. (\Rightarrow Informationsverlust, denn es ist ja schon interessant zu wissen, ob in Sektor 1 20 Teilchen gemessen wurden oder über alle verteilt...(??))
BRW sind wahrscheinlich bei meinem He-Subset immer 1, weil da eh nicht viel gemessen wurde...
DONE: einfaches BRW als Sectorweight übergeben. Später aber nochmal gucken, wie oft das nicht 1 ist.
sector weights
- Im 3D-Schnitt w in der Ecke angeben statt dem Slice
DONE: vmin ist jetzt der kleinste Wert nach 0, sowohl für *color_norm* = *all* als auch *color_norm* = *sg* fürs gesamte H-Array.
w statt Slice
- Wo kein Count ist weiß machen: *v_under*, white, vmin... DONE
Colormap Range
- Sunpulse-Offset:

Zum Backward-Engineering: SW ausschneiden in 5-min-Intervallen und dann det-sec Histogramm erstellen. Da sollte man gut erkennen, was aufleuchtet.

Robert Frage für Rudi (?) schicken.

Lars hat was im DPU gefunden. 180 Grad \pm 45 Grad...?
Testen Offset
- Bestimmung, indem ich Ableitung zwischen zwei Koordinatenangaben bestimme ($\Delta V // \Delta t$) \Rightarrow Gibt dann vx,vy und vz grob an. Mit in die Trajektoriendateien schreiben. Done, ewig am Einarbeiten in die PHAs gebastelt.
Eigengeschwindigkeit
- Leider läuft He schnell in den Threshold mit höheren ESA-Steps (dann nur noch als Doubles). Robert: Da wurden andere, ältere SSDs verwendet als bei ACE oder bei PLASTIC: Goldschicht mit ziemlich hoher Totschicht. Diesen Threshold nochmal prüfen! Denn He liegt weit unten, nicht versehentlich wichtige PHAs wegschneiden.
D.h., dass wir uns nur die äußeren Schalen und nur den schnellen SW angucken können (bzw. die Schnitte, die mittig zu entsprechend großen
ECH-Grenze

ws gehören...?) \Rightarrow Klar, denn kleinere Geschwindigkeit heißt kleinere Energie und He hat kleine Masse. Dafür haben wir da auch keinen störenden Hintergrund.

warum?

- He+ rausfiltern:

- erste Überlegung: Mit mpq-Algorithmus He sauber auf 4 abbilden (?). Dann liegen da noch ein paar andere drüber (oben rechts in EpQ vs ToF), die man über E vs ToF rausfiltern kann
Drüber nachdenken, aber es geht viel einfacher:

(Wie schreibe ich PHA-Worte weg? Wie liegen die eigentlich vor...?)

He ausschneiden und wegschreiben

- Range: In RNG0 sind He und Protonen, in RNG1 schwerere Ionen. Die Boxen sind einfache Rechtecke in m, mpq und dann in ToF vs EpQ Dreieck/Schlauch. D.h. als erstes nur RNG0 maskieren.
- Dann eine *wandernde Box* in E-vs-ToF-Darstellung (animate) legen: Evtl. per Mouseclick-Tracker: Bei welcher ToF ist der He-Bulk beim jeweiligen EpQ-Step? Wenn das PHA in einer Box innerhalb \pm Kanäle liegt: PHA wegschreiben, dann habe ich mein Set, auf dem ich arbeiten kann und man braucht keinen komplizierten mpq-Algorithmus. In der senkrechten kann ich das genauso mitfiltern, aber bei unseren 50 Tagen lag da bei rng0 eigentlich auch nichts.

Super, dass das so sauber ist!

- Doubles sind für uns nicht interessant, weil es für die keine Detektorinfo gibt \rightarrow keine 3D-Auflösung
- Geometriefaktor ist im Prinzip nur vom Instrument abhängig, soll ich also von ACE übernehmen
- Robert hat DPU geschickt. Mal reingucken

DPU

Do – 5. September

Heute: Nach und nach die Todos von gestern abarbeiten:

- Für die Effizienzen kurzes Skript geschrieben (`guess_ul_eff.py`), das eine Funktion an die ACE-Effizienzen fittet und den Ulysses-EpQ-Werten passende Werte zuordnet, weggeschrieben in `He1+.eff`, z.Zt. im Hauptskriptordner
- Subset-Mastermaskenproblem verstanden: An vielen Stellen werden die Daten aus `d` geholt ohne die Mastermaske aufzurufen. Das sind dann *alle* Daten. Mit dem Subset bin ich das umgangen, weil *alle* Daten dann das Subset waren. Vielleicht frage ich Lars nochmal. Ich ändere das jetzt erstmal nicht alles um, sodass die Daten der Mastermaske genommen werden, weil das ganz schön viel ist. Vielleicht brauche ich das eh nicht mehr, wenn ich mir mein He-Subset weggeschrieben habe. Bis dahin mache ich das weiter mit dem künstlichen Subset.

Fr – 6. September

Probleme beim BRW gelöst: Nuller wegmaskiert.

Probleme bei den Efficiencies: angehängte 0 für ACE übersehen, jetzt passts.

Eigengeschwindigkeit: Lars meinte, ich soll die Differenzenquotienten zwischen zwei Trajektorienpunkten bilden und das mit in die Trajektorienfiles schreiben. Ich brauche die Geschwindigkeit ja in RTN-Koordinaten. Die Trajektorienfiles liegen ja aber gar nicht in RTN vor. Könnte ich v in HG-Koord. berechnen und dann genauso transformieren?

Mo – 9. September

Ich wandle jetzt beide Positionen von HG sphärisch in RTN kartesisch um und subtrahiere 2-1. In die Liste schreibe ich dann v_x , v_y , v_z und den Betrag.

Habe den Betrag mit v aus `ulysses_daily_heliocentric...dat` verglichen: Immer leichte Abweichung. Lars meint, da soll ich mir keine Gedanken drüber machen – Rundung etc...

Hauptsachen, das SC durchläuft dieselbe Geschwindigkeit wieder im nächsten Orbit.

Komische Werte Ende 2001 gefunden: Da stehen für ca. 20 Tage nur Nullen in *helio.dat*. Die sollte ich mal interpolieren, vielleicht kann ich die auch ex- Werte helio.dat
akt berechnen aus HC...?

Done: Dafür habe ich ja eine Funktion, klappt super. Habe jetzt `fill_up_gap()` und `calc_velocities()` in `traj_pooled` geschrieben – dort wird die neue kompakte Trajektorienliste geschrieben. Die tatsächlichen etwas komplexeren Funktionen dazu stehe jetzt alle in `ul_calc_traj`.

Jetzt muss ich `Dist3D` damit verknüpfen.

Datenprodukte zugefügt und Ordnerstruktur etwas aufgeräumt.

Di – 10. September

Jetzt Riesenproblem: Wie soll ich das in das V-Space-registern integrieren? Ich kann doch nicht noch 3 Dimensionen mehr anlegen... Ich schaue mir den Verlauf der einzelnen Geschwindigkeiten jetzt erstmal an. Plotfunktion `plot_eigen_velocities()` in `pui_ulysses.pui`.

Warum wackeln die Geschwindigkeiten so? Das kommt durch Rundungen und dadurch, dass ich von AU auf km gehe. Ich versuche jetzt, den Radius in km einzuladen, das ist in einer der Files gegeben. Etwas nervig, aber gut... Einen Versuch ist es Wert.

Krass, das hat es voll gebracht. Ich sollte den Radius in km auch in die anderen Funktionen füttern, wo es geht.

Macht es irgendwie nicht:

- von HG zu HC gibt es in beiden Fällen eine Abweichung zum HC-Wert, km und AU unterscheiden sich nicht.
- von HC zu HG dito. Ich bräuhete einfach genauere Winkelwerte. Egal. Ich nehme R.km jetzt wieder aus der Pooled-Liste.

Ich bin SO verwirrt. Morgen weiter machen. In `calc_v` Zwischenwerte umrechnen oder sogar zwei verschiedene Funktionen schreiben.

Heute mit klarem Kopf nochmal angeguckt: Verwendung der Radiusangabe in km macht einen Unterschied für v_R , aber nicht für die anderen v -Komponenten, weil die Winkelangaben sowohl in HC als auch in HG zu ungenau sind. Umrechnen mit dem genauen Radius bringt nichts. Damit muss ich so leben.

Jetzt weiter mit dem echten Problem. Lars fragen...?

Nee, ich erweitere das Register jetzt einfach. Geht glaub ich eh kein Weg dran vorbei, nicht so schlimm, trägt zu meinem eigenen Verständnis bei – hoffentlich.

Wie groß sollen die Steps sein? Wenn ich abschätze, dass wir w auf 0.2 auflösen, entspricht das einer Geschwindigkeit von ca. $500 \frac{\text{km}}{\text{s}}$?!

Lars sagt, dass Translationen additiv sind, deshalb kann man auch beim Mappen der PHA-Worte mit den Geschwindigkeiten die Eigengeschwindigkeit dazuuaddieren (so wie das mit w // der SW-Geschw. passiert). Ich probiere das kurz aus, ob dasselbe rauskommt (Eigengeschwindigkeit im Grid abziehen oder händisch verschieben).

Frage nach der Auflösung:

- Beispielzeichnung w-Space mit 600 km/s: 30 km/s in y-Richtung verschoben (wie bei ACE): Teilchen auf Schale $w = 1$ (eig. 600) landet oben bei 630, $w = 1.05$ und unten bei 570, $w = 0.95$.
- Ich kann die Bins auch mal anpassen und schauen, was passiert. eigentlich wären Bins wie bei PLASTIC besser, also welche, die sich an den Phasenraum anpassen (rund). Die momentanen Rechtecke haben ja immer unterschiedlich viel des Kreises überdeckt (Frage to self: Ist das nicht irgendwie beim Phasenraum mit eingerechnet?). Das wäre ein weit entferntes ToDo...

Bin-Grid

Kommentare zum Phasenraum:

- Eigengeschwindigkeit dafür nicht mit rein, denn das ist rein instrumentenspezifisch (??).
- Beispiel Autos: in meiner Richtung Autos mit 95, aus der Gegenrichtung auch (gleichverteilt). Ich fahre mit 100. Weiter weiß ich nicht, nochmal drüber nachdenken, fragen.

Meinen Plot zur zeitaufgelösten Geschwindigkeit könnte ich mit in die MA nehmen. Zeigen, dass der Verlauf so ist, man den Fast-Latitude-Scan sieht, besonders in einer Komponente (wir schauen uns aber eher den Bereich an, wo das nicht so eine große Rolle spielt, da sind die Komponenten alle nicht so groß)

Morgen: weiter vergleichen.

Do – 12. September

Fazit: Es passt, macht also keinen Unterschied, ob ich den V-Raum im Grid anpasse oder das eigentliche Datenprodukt beim Adden zu den PHA-Worten (außer dass ich beim Grid gerundet habe ich z.B. nicht 9.5, 9.67 und 9.88 zufüge, sondern immer 10).

Jetzt: Schauen, ob die Richtungen im RTN richtig sind, yay...

Hab mir den Kopf zerdacht, s. S. 85 in Kladde. Hoffe die Richtung passt jetzt, ich bin zu blöd das zu prüfen. Muss morgen gucken, dass das auf mehrere Detektorpunkte auch klappt.

Fr – 13. September

Habe das gelöst mit der Funktion `numpy.tile`, `.T` und die Anzahl ist `self.col_dim`.

Jetzt Threshold Ech-Detektor checken. Wollte mal mehr Daten einladen, gab Problem mit `sync_traj`. Die Maske, die da gesetzt wird, ist glaube ich ziemlich sinnlos. Die war nur für `sync_swoops`, weil da ggf. durch 0 geteilt wird. Das passiert hier nicht, weil ich keinen Mittelwert bilde, weil d90 gleich ist.

Ergebnis: Stimmt wirklich. Ganz ohne Filter ist erst E-Channel 15 belegt. Manchmal gaaanz wenig in 12 oder 14 (?!).

Di – 17. September

Jetzt He⁺ filtern:

Brauchte lange, um `animate2d` aus `dbData` zu erweitern, um die Mouseclickfunktion einzupflegen. Muss immer noch manuell die Fkt. zum Setzen der Masken aufrufen, will gern wissen, wie ich das in die *init* bekomme. Die Funktion selbst ging dann aber gut. Morgen:

- Headline in gefilterte PHAs schreiben
- Überlegen, ob ich Epq noch rutnerzählen muss