

Csillagászati Laboratórium II.

7. óra: Psf fotometria

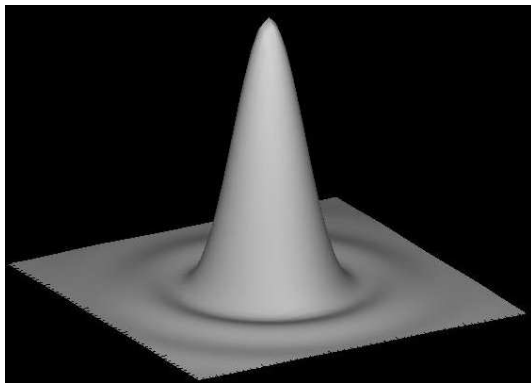
1. A psf fotometria elmélete

A csillagok köztudottan pontszerű fényforrások, mégis a felvételeken egy bizonyos félértékszélességgel rendelkező Gauss függvény formájú az intenzitáseloszlásuk. Ez több dolognak tudható be. Ha ideális észlelési körülmények között tudnánk dolgozni (azaz légkör felett, tökéletes optikai rendszerrel), akkor a csillagok fénye csak a távcsövek diffrakciós képét adná, azaz egy Airy-féle korong és gyűrűk rendszerét. Innen is a kifejezés, hogy „diffrakció limitált” leképezés, azaz, hogy tökéletes leképezés esetén mekkora szögátmérőjű objektumok lennének még megkülönböztethetőek a felvételeken.

Bevezetés a Csillagászatba gyakorlaton a távcsövek felbontóképességének is ezt tanultátok, ez a jól ismert:

$$\Theta = 1,22 \cdot \frac{\lambda}{D}$$

képlet. Ez megadja, hogy mekkora szögnél van az Airy-féle elhajlási képnél az első minimum. Ha egy másik csillagnak a maximuma ide esik, akkor azt már különálló objektumként látnánk.



Egy csillagnak, mint pontszerű objektumnak a létrejövő intenzitáseloszlását nevezzük *PSF*-nek (**P**oint **S**pread **F**unction), azaz „pont eloszlás függvénynek”. Általában a távcsövek nem ideális műszerek, így a létrejövő intenzitáseloszlások nem csak az Airy-féle képből állnak, hanem egyéb zavaró tagokat is tartalmaznak. Viszont nagyon fontos tulajdonsága a rendszernek, hogy az összfluxus állandó marad, csak „szétkenődik” egy kisebb területen. Fontos feltevézése a psf-fotometriának, hogy a psf a fókuszsíkban nem változik (jelentősen), hanem állandó marad. Természetesen az adaptív optikai rendszereknél, illetve a torzult (nagy látómezős égboltfelméréseket végző műszerek) látómezővel rendelkező távcsöveknél ez nem mindig igaz, ekkor korrekciós egyenleteket szoktak alkalmazni.

Természetesen nem csak az optikai rendszer van hatással a létrejövő psf-re. A földi észleléseknél a légkör nagyobb zavaró hatással van a kialakuló psf-ben, mint az optikai rendszer maga. A légköri mikroturbulenciákat felfoghatjuk úgy, mint random fázisú aberrációkat a rendszerben, melyek ráadásul az időben is random mód változnak, igen gyorsan. Mivel a csillagászati felvételek nem egy μs alatt készülnek, ezért ezen random változások kiegyenlítődnek, és a csillagok psf-jének egy általános kiszélesedéséhez vezetnek.

A psf-fotometria során ezen intenzitáseloszlást próbáljuk illeszteni modellfüggvényekkel. Az illesztéshez gondosan válogatott mintacsillagokra van szükségünk a látómezőkből. Az analitikus függvényeknek az eltérését az átlagos eloszlásokhoz képest megmérjük, így jön létre képenként egy „psf” kép. A fényesebb csillagok psf-je csak a függvény amplitúdójában különbözik a halványokétól, így ha van egy kezdő fényességértékünk (mondjuk apertúra fotometriából), akkor minden csillagra „be lehet iterálni” az adott képhez tartozó psf-et. Az illesztett függvénynek ki kell számolni a térfogatát, mely ekkor arányos lesz a csillag instrumentális fényességével.

2. Psf fotometria az *IRAF*-fal

Az előző órán már kiválogattatok a *psfselect* nevű *TASK*-kal egy adag csillagot, mely jó jelölt lehet arra, hogy a képenkénti psf-et segítségükkel határozzuk meg. Az *IRAF* ezen *TASK*-ja nem a lehető legjobb, így majd a listákat szemmel is végignézzük a *psf* nevű *TASK*-jának interaktív futtatása mellett. Mindenek előtt nézzük végig a *psf TASK* paraméterlistáját:

```
PACKAGE = daophot
TASK = psf
```

```
image = Input image(s) for which to build PSF
photfile= default Input photometry file(s)
pstfile = Input psf star list(s) (default: image.pst?)
psfimage= default Output PSF image(s) (default: image.psf.?)
opstfile= default Output PSF star list(s) (default:image.pst.?)
groupfil= default Output PSF star group file(s)
(plotfil= ) Output plot metacode file
(datapar= ) Data dependent parameters
(daopars= ) Psf fitting parameters
(matchby= yes) Match psf star list to photometry file(s)
(interac= yes) Compute the psf interactively ?
(mkstars= no) Mark deleted and accepted psf stars ?
(showplo= yes) Show plots of PSF stars ?
(plottyp= mesh) Default plot type (mesh|contour|radial)
```

...

A legtöbb paraméter a szokásos, már jól ismert paraméter. A photfile default-ja a phot által létrehozott mag fájlok. A pstfile pedig a psteselect által létrehozott pst fájl. Az output fájlokat nyugodtan lehet hagyni default-on. A datapars illetve daopars paraméterfájlokat a korábbiakhoz hasonlóan a „:e” lenyomásával tudjuk állítgatni. A matchby funkciót hagyjuk az alapbeállításon, azaz yes-en. Mivel interaktívan szeretnénk futtatni a psf-et, ezért ezt is hagyjuk yes-en.

A honlapról letölthető run2.sh nevű fájlt lefuttatva az M67-es képekre létrejön minden egyes képhez egy *.c12 nevű IRAF script (majd az NGC 2126-os képekhez kicsikét át kell alakítanunk). Mivel a psf TASK-ot interaktívan nem lehet több képre egyszerre egy scripttel meghívni, ezért más eljáráshoz nyúlunk. A c12 fájlokat egyenként betöltjük az IRAF-nak, és mindegyik betöltés után elindítjuk a psf TASK-ot. Valahogy így:

```
da> c1 < tft332.fit.c12
```

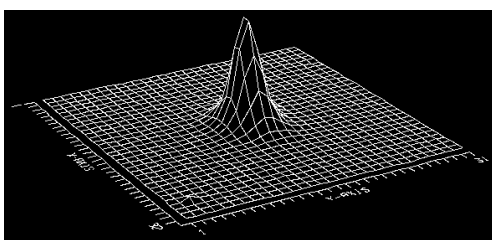
Ennek hatására betölti az aktuális képet, illetve beállítja a imex.pars alapján, mint előző órán, a képhez tartozó paramétereket. Ezek után indítsuk el a psf-et:

```
da> psf
```

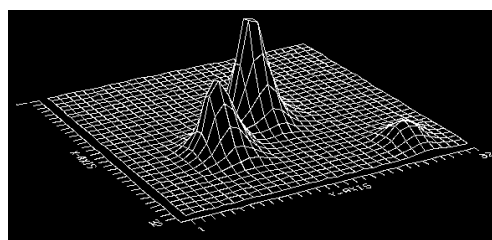
Itt a TASK újra rákérdez a paraméterekre (nyomjunk Entereket), majd a pstselect által kiválogatott csillagok mesh plotjait megmutatja nekünk. Lesz köztük olyan, mely jól reprezentálja azt, hogy hogyan is kéne kinéznie egy csillag psf-jének, s lesz olyan, mely nem alkalmas. Ezekre mutatok példát az 1. ábra képein. Itt több dolgot tudtok tenni. A „c” billentyű lenyomásával contour ábrát láthattok. Az eredeti mash plotot az „m” lenyomásával kapjátok vissza. Ha egy csillagot alkalmasnak találtok, akkor az „a” lenyomásával (add) adhatjátok hozzá a psf listához. Ha viszont nem jó, akkor „d” lenyomásával törölhetitek. Ha végigmentetek a pstselect által kiválogatott listán, akkor az „f” lenyomásával illeszthetitek a psf profilt, és „w”-vel írhatjátok ki az eredményt. Ha egy képpel végeztetek, akkor „q”-val léphettek ki a TASK-ból. Ugyanezt az algoritmust kell végig csinálnotok mindegyik képre. Az összesen 20 darab képre ez nem kéne, hogy sok időt vegyen el. Az NGC 2126-os képekhez a run2.sh-nak kicsit megváltoztatott változatára lesz szükség, a szintén a honlapon található run2_var.sh nevű fájlra.

A psf-hez használt run2.sh nevű script működését a jegyzet végén részletezem. A futás eredményeképp létrejött egy újabb pst fájl, amely már az általunk kiválasztott új listát tartalmazza. Ezen kívül egy *.psf.1.fits nevű fájl is. Mint ahogyan a nevéből is látszik, ez egy fits kép. Ha ezt megjelenítetek, akkor láthatjátok, hogy közelítően se úgy néz ki, mint ahogyan egy átlagolt psf-nek ki kéne néznie. Ez azért van, mert ez a kép azt mutatja meg, hogy az illesztett analitikus függvénytől mennyire tér el az átlagolt psf. Ha látni is kívánjátok, hogy hogyan néz ki a számolt psf-etek, akkor azt a seepsf TASK-kal tudjátok elérni.

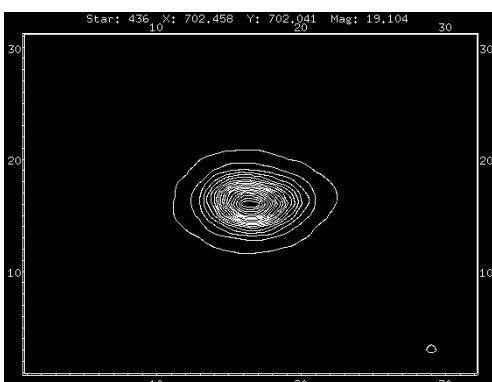
A létrejött *.psf.1.fits fájl a fejlécében tartalmazza a következő kulcsszavakat: "FUNCTION", "PSFHEIGHT", "NPARS" és "PARN". Ezek adják meg, hogy mekkora is az illesztett analitikus függvény paraméterei. A psf-ben összesen 6 darab analitikus függvény illesztésére van lehetőség: gauss, moffat15, moffat25,



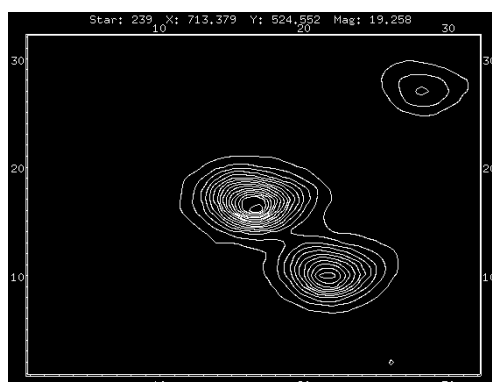
(a) Jó psf csillag mesh plot-ja



(b) Rossz psf csillag mesh plot-ja



(c) Jó psf csillag contour plot-ja



(d) Rossz psf csillag contour plot-ja

1. ábra. Jó és rossz psf csillagokra példa

penny1, penny2 és lorentz. Ha a function-t auto-ra állítjátok, akkor mindegyiket végig próbálja az *IRAF* és a legjobbat tartja meg ... természetesen ez viszonylag időigényes. A Schmidttel készült felvételek esetén általában a moffat25 szokott lenni a legjobban illeszkedő függvény.

Mivel a varorder kapcsoló 2-re van állítva, ezért az algoritmust megpróbálja egy másodrendű függvény illesztésével figyelembe venni a psf torzulását a látómezőben.

3. A számolt psf illesztése az összes csillagra

Most, hogy tudjátok, hogy hogyan is kéne kinéznie egy csillagnak az intenzitáseloszlásának képenként, megbecsülhetitek az egyes csillagoknak a fényességét. A kapott psf-t csak szimplán illeszteni kell a csillagok valódi intenzitáseloszlására. Ehhez nyújt kezdő iterációs értéket a korábban apertúra fotometriával meghatározott fényességérték. A psf-ek illesztését az allstar nevű *TASK*-kal hajthatjátok végre.

```
PACKAGE = daophot
TASK = allstar
```

```
image = Image corresponding to photometry
photfile= default Input photometry file (default: image.mag.?)
psfimage= default PSF image (default: image.psf.?)
allstarf= default Output photometry file (default: image.als.?)
rejfile = default Output rejections file (default: image.arj.?)
subimage= default Subtracted image (default: image.sub.?)
(datapar= ) Data dependent parameters
(daopars= ) Psf fitting parameters
```

...

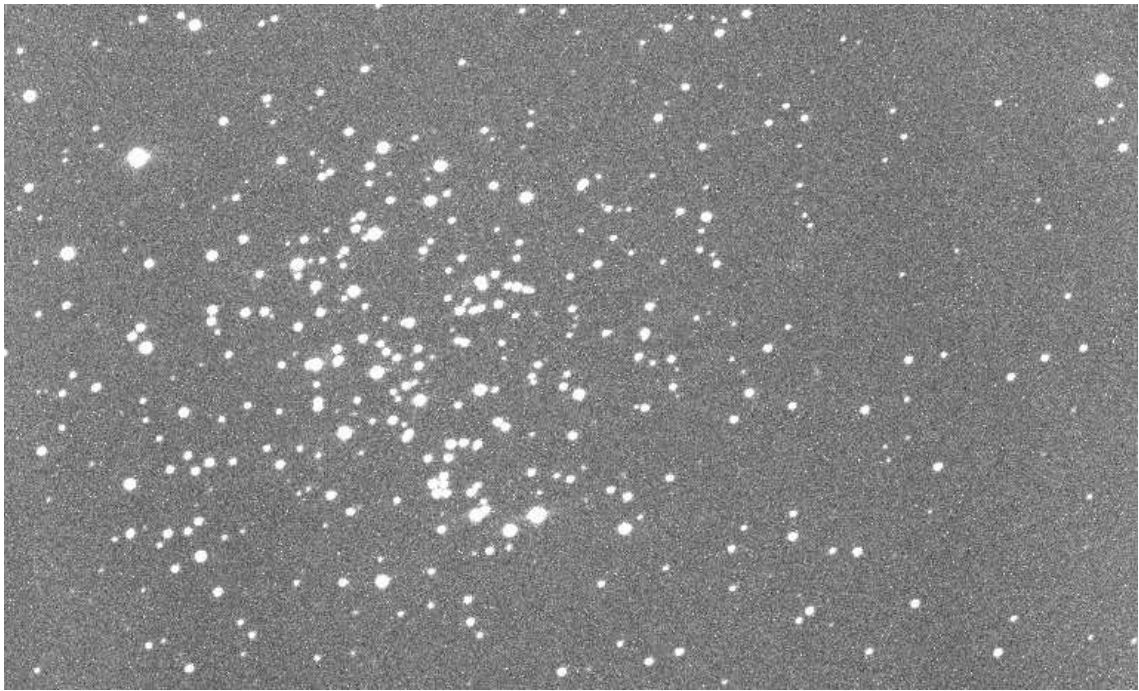
Az `allstar` paraméterfájlját végignézve láthatjátok, hogy viszonylag nem túl komplikált. Ha megadtok neki minden fájl `default`-ban, illetve jók a beállítások a `datapars` és `daopars`-ban, akkor valószínűleg minden gond nélkül lefut. A `datapars`-t és a `daopars`-t viszont már korábban is állítottátok scriptekkel, így most is ezt fogjátok tenni. Töltsétek le a honlapon található `run3.sh IRAF` scriptet generáló `bash` scriptet. Ha ezt lefuttatjátok, akkor létrejön egy `allstar.cl` nevű script. Ezt adjátok be az `IRAF`-nak.

```
da> cl < allstar.cl
```

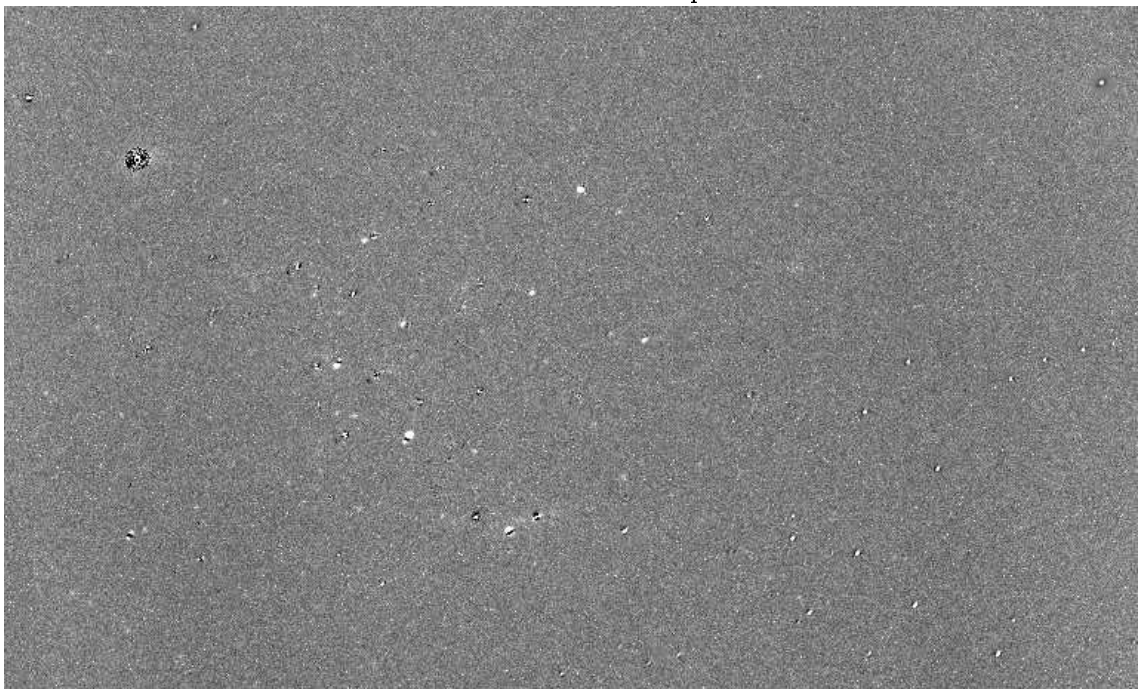
Ennek hatására a `TASK` az összes képen található összes csillagra elvégzi a meghatározott képenkénti psf illesztését a talált csillagokra. Eredményképp létrejön több fájl. Az illesztésekből meghatározott fényesség értékek (és egyéb értékek) a `*.als.1` fájlokban találhatóak. Azon csillagok, melyeket valamilyen ok folytán nem tudott illeszteni az algoritmus a `*.arj.1` fájlokban vannak. Az iterációhoz az `allstar` a csillagokat csoportosítja különböző algoritmusok alapján. A csoportok találhatóak a `*.psg.1` fájlokban...ez számunkra lényegtelen. Az `allstar.cl` scriptben a `txdump TASK`-ot is meghívjuk. Ez a `*.als.1` fájlokból kiszedi a számunkra lényeges információkat szépen oszlopokba rendezve, képenként egy különböző ASCII fájlba. Az NGC 2126-os képekhez a `run3_var.sh` fájl használjuk.

Kérdések, feladatok:

- 1. Értelmezzétek a psf fotometria elméletét a syllabusz első oldala alapján*
- 2. Mekkora lenne a piszkéstetői Schmidt távcső „diffrakció limitált” felbontása? Ehhez képest mekkora romlást jelent a 4-es „seeing”?*
- 3. Ha a CCD kamera 1536×1024 pixelből áll, és a látómező $29' \times 18'$, akkor mekkora egy pixelhez tartozó felbontás?*
- 4. Mind az NGC 2126 mind az M67-es felvételek psf fotometriáját végezzétek el a syllabusz alapján!*
- 5. A psf fotometria során az illesztésekhez tartozó illesztési hibákat írjátok fel. Ha auto-ban csináljátok a függvényillesztéseket, akkor írjátok fel azt is, hogy melyik függvényt illesztette az IRAF végülis, illetve azt, hogy hány csillagra csináltátok az illesztést.*
- 6. A kapott képeknél a létrejött sub képeket nézzétek át. Ezek azok, amelyek az eredeti képet mutatják meg a psf illesztések és levonások után. Ha a végeredmény kép egy szép homogén eloszlású kép, akkor jó a psf fotometria.*



2. ábra. Az eredeti kép



3. ábra. A psf illesztés és levonás utáni maradványok

4. A scriptek értelmezése

4.1. run2.sh

```
#!/bin/bash

awk '{print "echo \"\" \"display \" $1 \" 1\\\"\" \" > \" $1\".c12\";\n
print "echo \"\" \"unlearn psf\" \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"datapars.fwhmpsf=\"$4 \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"datapars.sigma=\"$3 \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"datapars.datamin=\"$2-7*$3\"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"daopars.psfad=3.5*$4 \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"daopars.function=\\\"\\\"moffat25\\\"\\\"\" \"\\\"\" \" >> \"$1\".c12\";\n
print "echo \"\" \"daopars.varorde=2 \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"daopars.fitrad=2.8*$4 \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"daopars.wsannul=5 \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"psf.matchby=yes\" \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"psf.pstfile=\\\"\\\"default\\\"\\\"\" \" \" \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"psf.interac=yes\" \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"psf.image=\\\"\\\"$1\\\"\\\"\" \" \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"psf.verify=no\" \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"psf.update=no\" \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"psf.verbose=no\" \"\\\"\" \" >> \" $1\".c12\";\n
print "echo \"\" \"daopars.sannulu=3.8*$4 \"\\\"\" \" >> \" $1\".c12\"}'\n
imex.pars > psf.run

chmod 755 psf.run
./psf.run
rm psf.run
```

Látható, hogy ez egy bash script, mely egy awk parancsot tartalmaz. Az awk parancs az `imex.pars logfile` alapján gyárt le képenként egy `*.c12` nevű fájlt, amely tulajdonképpen egy *IRAF* script lesz. Láthatóan a paramétereket állítja be a `datapars` illetve `daopars` paraméterfájlnak. Emellett a `psf TASK`-nak is beállít egy-két dolgot.

Legelőször a képet kiteszi az 1-es frame-be. Ezek után kiadja majd az *IRAF*-ban a következő parancsot: `unlearn psf`. Ezzel a paranccsal, hogy `unlearn` lehet minden *IRAF TASK*-ban kitörölni minden korábbi beállítást és az alapbeállításra állítani mindent. Ez az előre megírt pipeline-oknál tud nagyon hasznos lenni, ugyanis nem lehet tudni, hogy a tisztelt felhasználó korábban milyen paramétereket állított be magának, ezzel viszont minden visszaáll. Az egyes további beállított paramétereket a korábban elmondottak alapján értelmezték.

A script a végén futtatási jogot ad a létrehozott `psf.run` nevű fájlnek, azt le is futtatja, illetve rögtön utána le is törli, hogy ne foglaljon extra helyet. Ezzel létrejöttek a képenkénti `*.c12 IRAF` scriptek, amiket egyesével betölthetünk. Ezt sajnos azért kell így csinálni, mert ha az *IRAF*-nak a scripteket egy másik scripttel hívnánk meg, akkor nem lehetne kilépni az egyes illesztések után... sajnos ez a hiba még a legújabb verzióban is benne van. A `run2_var.sh` ezzel teljesen megegyező, csak kijavítja azt a hibát, amit a WCS-be való betevés utáni megváltozott nomenklatúra okozna.

4.2. run3.sh

```
#!/bin/bash
awk '{print "allstar", $1, "photfile=default",\
"psfimage=default", "allstarf=default", "rejfile=default",\
"subimage=default", "fwhmpsf="$4, "sigma="$3, "datamin="$2-7*$3,\
"function=moffat"25, "varorde="2, "psfrad="3.8*$4, "fitrad="2.8*$4,\
"sannulu="3.8*$4, "wsannul="5}' imex.pars > allstar.cl

awk '{print "txdump", "textfile="$1".als.1,"$1".arj.1",\
"fields=id,xcenter,ycenter,mag,merr,chi,xairmass",\
"expr=yes" " >> " $1".dat"}' imex.pars >> allstar.cl
```

Ez a script roppant mód hasonlít a már korábban (előző óra) értelmezett run1.sh scripthez. Az alapján könnyen tudjátok magatok is értelmezni ennek a működését. Mivel az *allstar* paramétereit már értelmeztük, ezért ez nem okozhat nehézséget. A második *TASK* amit futtat a script, az a *txdump*.

Ez a script az, amely kiszedi az infókat az **.als.1* fájlokból. Értelmezése minden hallgatóra rá van bízva az *IRAF*-on belül található help fájlok alapján (help *txdump*).