ATK tähtitieteessä

Osa 2 - IDL perusominaisuudet

12. syyskuuta 2014

IDL - *Interactive Data Language*

- IDL on tulkattava ohjelmointikieli, jonka vahvuuksia ovat:
  - Yksinkertainen, johdonmukainen komentosyntaksi.
  - Voidaan käyttää interaktiivisesti, tai rakentamalla ohjelmia.
  - Erilaisten datamuotojen samankaltainen käsittely (skalaarit, vektorit, jne.):
    - Operaattorit vaikuttavat kokonaisiin taulukoihin, jolloin esimerkiksi silmukoiden tarve on vähäistä.
  - Suuri määrä rutiineja datan esittämiseksi graafisesti, samoin suuri määrä numeerisia rutiineja.
  - Hallitsee useita tiedostoformaatteja. Monipuoliset INPUT/OUTPUT komennot.
  - Voi kutsua muun muassa FORTRANilla ja C:llä kirjoitettuja rutiineja.
  - Koodi on hyvin siirrettävää eri käyttöjärjestelmien välillä, jos välttää järjestelmäkutsuja.
  - Suuri määrä tähtitieteellisiä rutiineja ladattavissa. (mm. NASA IDL Astronomy User's Library: `http://idlastro.gsfc.nasa.gov/`)

IDL - *Interactive Data Language*

- Vastaavasti heikkouksia ovat:
  - Ohjelmien toimivuus edellyttää sitä, että koneeseen on asennettu IDL. Binäärikoodia ei ole mahdollista luoda.
  - Hinta.
    - Lisenssi on hyvin kallis ja monet instituutitkin ovat jättäneet sen ostamatta. Esim. Python on ilmaisena varteenotettava vaihtoehto.
    - Myös ilmaisia IDL-variantteja. Ovat yleensä pari versionumeroa perässä virallista. Paras tällä hetkellä GDL: `http://gnudatalanguage.sourceforge.net/`

IDL - käynnistys

- Kaksi tapaa:
  - Komentorivillä käsky:
    - `idl` – avaa IDL:n komentotilan.
  - IDL:n omaa sovelluskehitystympäristöä voi myös käyttää. Tämä tapahtuu komennolla:
    - `idlde`
- Voitte valita kumpaa tapaa käytätte. IDL:n komentorivi on kurssin luonteen vuoksi suositeltava tapa.
- HUOM! Jos et ole tehnyt ensimmäisen harjoituksen tehtäviä 3. ja 4. tee ne ennen kuin alat käyttämään IDL:ää!
  - Näissä tehtävissä luodaan IDL:n käyttöä helpoittavia käynnistysparametreja.

IDL - manuaali

- Tärkein ominaisuus varsinkin aluksi on IDL:n oma manuaali.
- Manuaalin saa avattua IDL:n komentorivillä komennolla ?.
- Hyvin kattava. Sisältää helposti ymmärrettävät ohjeet jokaisen perusrutiinin ja proseduurin käytöstä.
- Harjoituksissa tutustutaan tarkemmin.
- Luonnollisesti myös Google tarjoaa apua moniin ongelmiin. Lisäksi on kattava sivusto `http://www.idlcoyote.com/`, josta voi löytää monia hyödyllisiä vinkkejä varsinkin edistyneemmälle käytölle.

IDL - komentotila

- Komentotilaa voidaan käyttää interaktiivisesti, jolloin se toimii esimerkiksi taskulaskimena:

```
IDL> print,3.+.5
      3.50000
```

- Monien komentojen syntaksi on seuraava:

```
KOMENTO, parametri1, parametri1, ..., keyword1=value, keyword2=value, ...
```

- Harjoituksissa on paljon esimerkkejä interaktiivisesta käytöstä, jotka käytte läpi.
- Voit myös ajaa normaaleja komentorivin komentoja lisäämällä $-merkin alkuun (esim. `$ls`).
- IDL ei tee eroa isojen ja pienten kirjainten välillä kuten jotkin muut ohjelmointikielet, tai Linux-terminaali yleensä.

IDL - muuttujatyypit

- Kuten muissakin ohjelmointikielissä, myös IDL:ssä on olemassa erilaisia muuttujatyyppejä. Tässä tärkeimmät:
  - **Integer** – kokonaisluvut välillä [-32768, 32768].
  - **Long** – Kokonaisluvut välillä $[-2^{31}, 2^{31} - 1]$].
  - **Floating-point** – Yksinkertaisen tarkkuuden liukuluvut välillä $[-10^{38}, 10^{38}]$. Kuusi merkitsevää numeroa.
  - **Double-precision** – Kaksinkertaisen tarkkuuden liukuluvut välillä $[-10^{380}, 10^{380}]$, 16 merkitsevää numeroa.
  - **Complex** – Liukulukupari (reaaliosa-imaginääriosa).
  - **String** – Merkkijono, 0-32767 merkkiä.

IDL -muuttujien luominen

- Erilaisten muuttujien luominen on helppoa, esimerkiksi:

```
IDL> a=5.5
IDL> b=8.3
IDL> c=a+b
IDL> d='Laske a+b='
IDL> print,c
      13.8000
IDL> print,d,c
      Laske a+b= 13.8000
```

- Muuttujan tyypistä saa tietoa komennolla `help`:

```
IDL> help,c
C FLOAT = 13.8000
```

- Huomioi muuttujatyypin merkitys!

```
IDL> print,5/3
      1
IDL> print,5./3.
      1.66667
```

  - Usean muuttujatyypin laskutoimituksissa tulos annetaan "tarkimman" muuttujatyypin mukaan.

IDL -matemaattiset operaatiot

- Kaikki laskutoimitukset suoritetaan normaalissa aritmeettisessa järjestyksessä. Tässä operaatioita:
  - ^ – potenssi.
  - *****, / – kerto- ja jakolasku.
  - **mod** – modulo.
  - **+**, - – yhteen- ja vähennyslasku.
  - $<$, $>$ – pienempi kuin, suurempi kuin.
- Esimerkiksi:

```
IDL> print,a
      5.50000
IDL> print,a^2*3
      90.7500
IDL> print,a^(2*3)
      27680.6
```


IDL - matemaattiset funktiot

- IDL sisältää normaalit matemaattiset funktiot. Esimerkiksi:
  - **sin(x), cos(x), tan(x)** – missä x on radiaaneina.
  - **asin(x), acos(x), atan(x)**
  - **atan(y,x)** – palauttaa kulman α, jolle $y = sin(\alpha)$ ja $x = cos(\alpha)$.
  - **sinh(x), cosh(x), tanh(x)**
  - **exp(x), sqrt(x), abs(x)** – eksponenttifunktio, neliöjuuri ja itseisarvo.
  - **alog(x), alog10(x)** – luonnollinen ja 10-kantainen logaritmi.

IDL - taulukot

- Yksinkertaisimmillaan vektorin voi määrittää arvoineen itse:

```
IDL> a=[0,1,2,3,4]
IDL> print,a
      0 1 2 3 4
IDL> help,a
      A INT = Array[5]
```

- Tämä ei kuitenkaan ole useimminkaan kovin kätevää. Aiemmin esitellyistä muuttujatyypeistä voidaan helposti muodostaa taulukoita:
  - **a=fltarr(100)** – luo 100 alkioisen liukulukutaulukon muuttujanimelle a. Alkioiden arvot on alustettu nolliksi.
  - Vastaavasti **intarr, dblarr, complexarr, ...**
  - Myös merkkijonotaulukoita voi luoda: esimerkiksi **lista=strarr(100)**. Alkiot on alustettu tyhjiksi merkkijonoiksi.

IDL - taulukot

- Usein halutaan taulukoita, joiden arvot on alustettu valmiiksi muuttumaan tasavälisesti. Tämä tapahtuu seuraavasti:

```
IDL> a=indgen(5)
IDL> print,a
       0 1 2 3 4
IDL> help,a
      A INT = Array[5]
IDL> b=findgen(5)
IDL> print,b
       0.00000 1.00000 2.00000 3.00000 4.00000
IDL> help,b
      B FLOAT = Array[5]
```

- Edelleen lindgen, dindgen, ...

IDL - taulukot

- HUOM! IDL:ssä taulukon ensimmäisen alkion indeksi on 0 (esim. a(0)). Edellisissä viisialkioisissa taulukoissa viimeisen alkion indeksi on siis 4, eikä 5 (esim. a(4)).
- Alkioihin voi myös viitata helposti:

```
IDL> print,a
      0 1 2 3 4
IDL> print,a(0)
      0
IDL> print,a(5)
% Attempt to subscript A with <INT ( 5)> is out of range.
% Execution halted at: $MAIN$
IDL> print,a(4)
      4
```

- Alkioiden arvoja voi myös muuttaa samalla tavalla:

```
IDL> c=indgen(6)
IDL> print,c
      0 1 2 3 4 5
IDL> c(4)=1234
IDL> print,c
      0 1 2 3 1234 5
```


IDL - taulukot

- Aiemmin oli jo mainita, että operaatioita voidaan kohdistaa suoraan taulukoihin. Esimerkiksi:

```
IDL> help,a,b
     A INT = Array[5]
     B FLOAT = Array[5]
IDL> print,a,b
      0 1 2 3 4
      0.00000 1.00000 2.00000 3.00000 4.00000
IDL> print,a+b
      0.00000 2.00000 4.00000 6.00000 8.00000
IDL> print,a*b
      0.00000 1.00000 4.00000 9.00000 16.0000
```


IDL - moniuloitteiset taulukot

- Moniuloitteisten taulukoiden luominen onnistuu yksinkertaisesti esimerkiksi **a=fltarr(100,100)**. Tämä luo kaksiuloitteisen taulukon, jossa on 10 000 alkiota.
- Vastaavasti voidaan käyttää esim. **a=findgen(100,100)**.
- Yksittäisiin alkioihin viittaaminen:
  - **a(rivi,sarake)**.
  - **a(0,0)** – ensimmäisen rivin ensimmäinen sarake.
  - **a(*,0)** – koko ensimmäinen sarake.
  - **a(*,0:4)** – viisi ensimmäistä saraketta.
  - **a(0:4,0:4)** – viiden ensimmäisen rivin viisi ensimmäistä saraketta.

IDL - moniuloitteiset taulukot

- Eli jos a on 2*3 matriisi:

$$\begin{matrix} a(0,0) & a(0,1) \\ a(1,0) & a(1,1) \\ a(2,0) & a(2,1) \end{matrix}$$

- Tällöin tallennusjärjestys muistissa on: a(0,0), a(0,1), a(1,0), a(1,1), a(2,0), a(2,1).
  - Moniuloitteiseen taulukkoon voidaan viitata myös yksiuloitteisena, esimerkiksi: a(0,1) = a(1) ja a(2,1) = a(5).

IDL - taulukko-operaatioista

- Taulukon suurimman ja pienimmän arvon saa komennoilla **max** ja **min**:

```
IDL> x=findgen(10)
IDL> print,min(x),max(x)
       0.00000 9.00000
```

- Sekalaisen taulukon voi järjestää **sort**-komennon avulla, joka palauttaa taulukon suuruusjärjestystä vastaavat indeksit.

```
IDL> x=[3,7,12,7,126,8,12]
IDL> ind=sort(x)
IDL> print,ind
       0 1 3 5 2 6 4
IDL> print,x(ind)
       3 7 7 8 12 12 126
```

- Ja taulukosta voi etsiä alkioita **where** komennolla, joka palauttaa annetut ehdot täyttävät indeksit.

```
IDL> x=findgen(10)
IDL> ind=where(x lt 5. and x ge 1.)
IDL> print,x(ind)
       1.00000 2.00000 3.00000 4.00000
```


IDL - datan graafinen esittäminen plot-proseduurilla

- Komennolla **plot** voidaan esittää yksiuloitteisten taulukoiden sisältöä. Yksinkertaisimmillaan:

```
IDL> x=findgen(10)
IDL> plot,x
```

- Tai esimerkiksi:

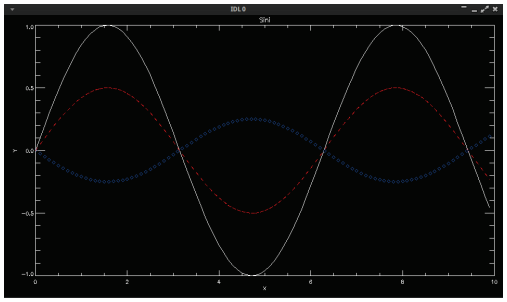
```
IDL> x=findgen(10)
IDL> y=sin(x)
IDL> plot,x,y
```

- Käytännöllisimmät lisäparametrit (keywordit):
  - **psym=x** – plottausymboli, esimerkiksi psym=2 plottaa tähdet datapisteisiin.
  - **color=x** – plotin väri. Riippuu käytetystä väripaletista.
  - **linestyle=x** – viivan tyyli, esimerkiksi linestyle=2 on katkoviiva.
  - **title='otsikko'**, **xtitle='x-akseli'**, **ytitle='y-akseli'**.

IDL - data graafinen esittäminen plot-proseduurilla

- Komennolla **oplot** voi piirtää aiemman plotin päälle:

```
IDL> x=findgen(100)/10.
IDL> y=sin(x)
IDL> plot,x,y,xtitle='X',ytitle='Y',title='Sini'
IDL> oplot,x,y/2.,color=2,linestyle=2
IDL> oplot,x,-y/4.,color=4,linestyle=4,psym=4
```



IDL - datan graafinen esittäminen plot-proseduurilla

- Komennolla **window** voidaan avata ennen plottausta uusi ikkuna, johon plot-komento piirtää.
- Kun avaa uuden ikkunan vapaaseen ns. ikkunaindeksiin ei seuraava plot-komento piirrä edellisen kuvaajan päälle.
- Avainsanoilla voidaan kontrolloida esimerkiksi ikkunan kokoa pikseleissä. 800x800 pikselin kokoinen ikkuna ikkunaindeksiin 1 avautuu komennolla:
  - `window,1,xsize=800,ysize=800.`
- Vapaaseen ikkunaindeksiin voi avata ikkunan käyttämällä `/free`-avainsanaa:
  - `window,/free,xsize=800,ysize=800.`
- Aktiivisen ikkunan voi katsoa komennolla: `print, !d.window`
- Ikkunan muuttaminen aktiiviseksi komennolla:
  `wset, ikkunaindeksi`
- Ikkunan voi myös sulkea komennolla:
  `wdelete, ikkunaindeksi`

IDL - kehittyneempi plot()-funktio

- IDL:ssä on myös kehittyneempi funktio kuvaajien tekemistä varten, jonka syntaksi on hieman erilainen.
- Toisaalta tämä funktio tarjoaa interaktiivisen ikkunan kuvaajan tarkempaan muokkaamiseen.
- Peruskäsky on **plotinnimi=plot(data)**, esimerkiksi:

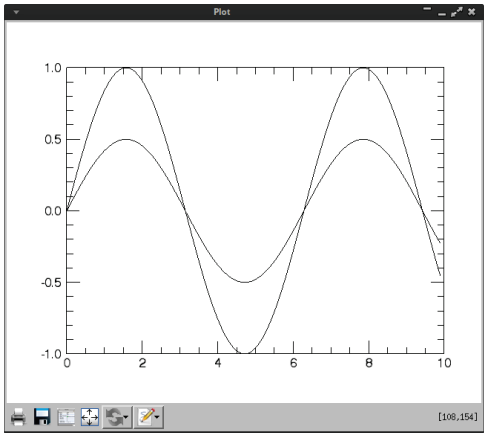
```
IDL> x=findgen(100)/10.
IDL> y=sin(x)
IDL> omaplotti=plot(x,y)
```

- Aiemman kuvaajan päälle voi piirtää käyttämällä avainsanaa **/overplot**, esim:

```
IDL> omaplotti=plot(x,y/2.,/overplot)
```

- Ikkunan valikoista voi editoida muun muassa viivatyyliä, plottaussymboleita, sekä kuvaajan voi tallentaa tai printata (mahdollista myös komentotilassa, katso manuaalista tarkemmin).

IDL - kehittyneempi plot()-funktio



IDL - kehittyneempi plot()-funktio

- Jokainen `a=plot()`-kutsu (ilman /overplot avainsanaa) avaa uuden ikkunan, toisin kuin yksinkertaisempi `plot`-proseduuri.
- `plot`-proseduuri ja `plot()`-funktio käyttävät myös täysin eri ikkunaindeksejä, joten ne eivät voi mennä sekaisin.
- Ikkunan indeksi on nyt muuttuja johon `plot()`-kutsu viittaa, ja akviinen ikkuna on uusin avattu.
- Esimerkiksi avataan kaksi plottia a ja b:
  - `a_plotti=plot(x,y)`
  - `b_plotti=plot(x,y)`
- Jos nyt halutaan esimerkiksi muuttaa aktiivinen ikkuna takaisin a-plotiksi koska haluamme piirtää sen päälle. Tämä onnistuu komennolla:
  - `a_plotti.window.setcurrent`