1 IDL (11.10.2005)

IDL on tulkattava ohjelmointikieli, jonka vahvuuksia ovat:

-Voidaan käyttää interaktiivisesti tai rakentamalla helppokäyttöisiä ohjelmia.

-IDL:n operaattorit vaikuttavat kokonaisiin taulukoihin, jolloin esimerkiksi silmukoiden tarve on vähäistä ja ohjelmat voidaan kirjoittaa muotoon, joka on sekä lyhyt, että havainnollinen.

-IDL:ssä on suuri määrä rutiineja datan esittämiseksi graafisessa muodossa, samoin suuri määrä numeerisia rutiineja.

-IDL hallitsee useita tiedostoformaatteja.

-IDL voi kutsua FORTRANilla tai C:llä kirjoitettuja rutiineja.

-IDL koodi on hyvin siirrettävää ympäristöstä toiseen, jos välttää käyttöjärjestelmäkutsuja

-Toisaalta IDL-ohjelmien toimivuus edellyttää sitä, että koneeseen on asennettuna IDL, binäärikoodia ei ole mahdollista luoda.

2 IDL:n perusominaisuuksia

IDL käynnistetään Unix-koneessa kirjoittamalla komentorivillä `idl`. Jos ei avaa ikkunoita, niin IDL:lää voi käyttää myös tekstipääteyhteydessä.

Kun IDL on käynnistynyt, niin komentoja pääsee antamaan IDL:n komentoriville:

```
IDL>
```

2.1 Muuttujatyypit

Kuten monissa muissakin ohjelmointikielissä, niin IDL:ssäkin on olemassa erilaisia muuttujatyyppejä. IDL:n tärkeimmät muuttujatyypit ovat:

Integer: Kokonaisluvut väillä $[-32768, +32768]$.

Long: Kokonaisluvut välillä $[-2^{31}, 2^{31}-1]$

Floating-Point: Yksinkertaisen tarkkuuden liukuluvut, itseisarvot välillä $[10^{-38}, 10^{38}]$. Kuusi merkitsevää numeroa.

Double-Precision: Kaksinkertaisen tarkkuuden liukuluvut, 16 merkitsevää numeroa.

Complex: Reaaliosa-imaginääriosa pari liukulukuja.

String: Merkkijono, 0 - 32767 merkkiä.

IDL:ää voi käyttää "taskulaskimena" print-komennon avulla:

```
IDL> print,5.5+8.3
      13.8000
```

Muuttujien luominen IDL:ssä on helppoa, esim.

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

Kuten viimeisestä komennosta käy ilmi, IDL:n komennoille ja ohjelmille voidaan syöttää argumentteina myös laskutoimituksia!

Muuttujan tyypin voi tarkistaa help-komennolla:

```
IDL> help,b
B               FLOAT     =       8.30000
```

Muuttuja B on siis yksinkertaisen tarkkuuden liukuluku. Huom! IDL:ssä pienet ja isot kirjaimet ovat samanarvoisia.

Muuttujatyypeillä tehtävistä laskutoimituksista kannattaa huomata se, että kokonaislukujen laskutoimituksista tuloksena on kokonaislukuja jne. Sekatoimituksissa tulos on "tarkimman" tekijän tyyppiä, esim. kokonaisluku + liukuluku = liukuluku. Seuraava esimerkki havainnollistaa eroja:

```
IDL> print,3/2
       1
IDL> print,3/2.0
      1.50000
```

2.2 Matemaattiset operaatiot

IDL:n matemaattiset operaatiot:

$\hat{}$potenssiin korotus
$*, /$ kerto- ja jakolasku
mod modulo
$+, -$ yhteen- ja vähennyslasku
$<, >$ pienempi kuin, suurempi kuin

2.3 Matemaattiset funktiot

IDL sisältää suuren joukon matemaattisia funktioita. Yksittäisten lukujen lisäksi argumentit voivat olla taulukoita.

$sin(x), cos(x), tan(x)$, missä x 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, jne.

2.4 Taulukot

Edellisistä muuttujatyypeistä voidaan helposti muodostaa taulukoita.

`b=fltarr(100)` luo 100 alkioisen liukulukutaulukon, arvot alustettu nolliksi. Vastaavasti `bytarr, intarr, lonarr, dblarr, complexarr ...`

Myös merkkijonotaulukoita voi luoda: `lista=strarr(100)`. Alkiot on alustettu tyhjiksi merkkijonoiksi.

Määriteltyjen taulukoiden arvoja voidaan muuttaa esimerkiksi antamalla taulukon tiettyjä arvoja näppäimistöltä, esim.

```
IDL> b=fltarr(100)
IDL> b(10)=5.5
```

muuttaa määritellyn liukulukutaulukon alkion numero 10 arvoksi 5.5. Huom, taulukon ensimmäisen alkion indeksi on IDL:ssä 0! Toisin sanoen edellisen esimerkin taulukon b viimeisen alkion indeksi on 99, ei 100.

Usein halutaan luoda taulukkoja, joiden arvo muuttuu säännöllisesti. Tämä tapahtuu seuraavasti:

`a=indgen(100)` luo kokonaislukutaulukon, alkioiden arvot 1-99. Vastaavasti `indgen, lindgen, findgen,....`

Useampiulotteisten taulukoiden luonti on helppoa: esim. `a=fltarr(200,200)` luo kaksiulotteisen taulukon jossa on yhteensä 40 000 alkiota.

Taulukon voi luoda myös seuraavasti komentoriviltä:

```
IDL> x=[0.5,-0.1,8.6,11.0,-7.2,-4.6,0.77,-18.1]
IDL> help,x
X               FLOAT     = Array(8)
```

Eli esimerkissä luotiin kahdeksanalkioinen liukulukutaulukko.

Koska sijoituskomennon kopiovat numeeristen arvojen lisäksi myös muuttujatyypin, niin olemassaolevia taulukoita voidaan käyttää helposti uusienluomiseen, Esim käyttämällä edellisen esimerkin taulukkoa x

```
IDL> y=x*0.
IDL> help,y
y               FLOAT     = Array(8)
```

Jos taulukon alkiot haluaa järjestää suuruusjärjestykseen, niin siinä voi käyttää apuna `sort`-käskyä. Se toimii eri tavalla kuin esimerkiksi Unixin sort, sillä se ei järjestä itsessään taulukkoa, vaan palauttaa taulukon suuruusjärjestystä vastaavat indeksit (pienimmästä suurimpaan):

```
IDL> ind=sort(x)
IDL> print,ind
           7           4           5           1           0
           6           2           3
```

Jos nyt halutaan sitten muokata taulukko suuruusjärjestykseen, niin käytetään äsken määriteltyä indeksitaulukkoa: `x=x(ind)`. Tämän olisi voinut tehdä myös suoraan kirjoittamalla `x=x(sort(x))`.

Jos taas taulukon haluaa järjestää kääntäen suurimmasta pienimpään, niin voi käyttää edellisten komentojen jälkeen `reverse` -käskyä: `x=reverse(x)`.

Taulukon maksimi- ja minimiarvot saadaan komennoilla `max` ja `max`:

```
IDL> print,max(x),min(x)
      11.0000     -18.1000
```

Usein taulukoista halutaan etsiä niitä alkioita, jotka täyttävät tietyn ehdon. Sen voi tehdä käyttämällä `where`-komentoa:

```
IDL> ind2=where(x>0.0)
IDL> y=x(ind2)
IDL> print,y
     0.500000     0.770000      8.60000      11.0000
```

Edellisessä esimerkissä sijoitetaan taulukkoon ind2 ne taulukon x indeksit, joita vastaavat arvot ovat suurempia kuin 0.0. Tämän jälkeen määritellään taulukko y, joka sisältää näitä indeksejä vastaavat alkiot.

3 Datan graafinen esittäminen

IDL tarjoaa useita erilaisia mahdollisuuksia esittää dataa graafisessa muodossa. Voidaan piirtää käyriä, kontuurikäyriä, pintoja, kuvia jne. Graafinen esitys voidaan tehdä kuvaruudulla erilliseen ikkunaan tai kirjoittaa postscript-tiedostoon printtaamista varten.

Ikkunan avaaminen tapahtuu komennolla `window`. Komennolle voidaan antaa useita erilaisia avainsanoja ja parametrejä. Esimerkiksi komento

```
IDL> window,/free,xsize=700,ysize=700
```

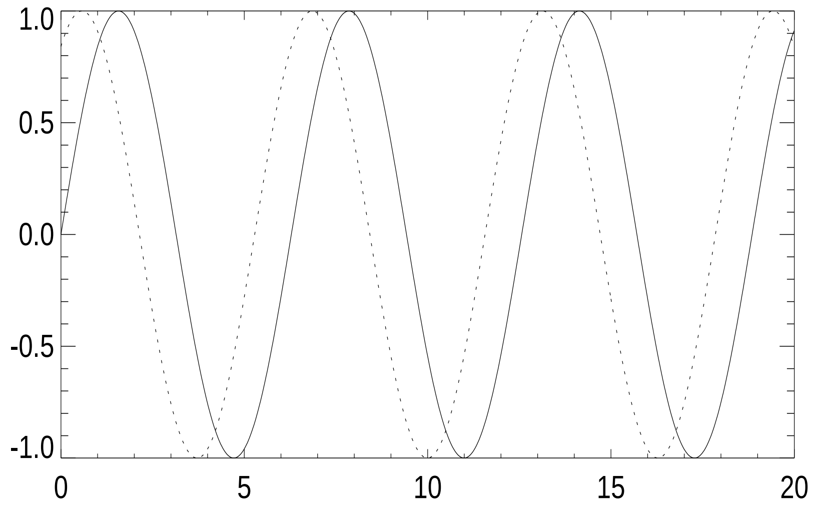
avaa ikkunan, jonka koko on 700x700 pikseliä. Avainsana `/free` tarkoittaa sitä, että IDL etsii ikkunalle vapaan ikkunaindeksin, eikä hävitä jotain olemassaolevaa ikkunaa. Joissakin käyttöympäristöissä on myös syytä käyttää avainsanaa `retain=2`, joka liittyy ikkunan sisällöstä huolehtimiseen. Jos avainsanaa ei näissä ympäristöissä käytä, niin ikkunaan tulostettu kuva katoaa jos toista ikkunaa käyttää ikkunan päällä.

Ikkunan voi tuhota komennolla `wdelete,nro` missä nro on tuhottavan ikkunan indeksin ilmaiseva numero. Jos näytöllä on useita ikkunoita, niin halutun ikkunan voi aktivoida tuhoamista varten komennolla `wset,nro`.

3.1 PLOT ja OPLOT

Yksiulotteisten taulukoiden sisältöä voidaan esittää komennolla `plot`. Luodaan aluksi taulukko, joka sisältää x:n arvot välillä 0-20 ja lasketaan sin(x), ja tulostetaan funktion y=sin(x) kuvaaja:

```
IDL> x=findgen(1001)/50.0
IDL> y=sin(x)
IDL> plot,x,y
```



Kuva 1: Sini-plotti

Usein halutaan tulostaa useita kuvaajia samaan ikkunaan. Tämä tapahtuu siten, että ensimmäinen käyrä tulostetaan samaan tapaan kuin edellä ja muut komennolla `oplot`. Käyrien erottamiseksi toisistaan, ne on syytä piirtää erilaisella viivalla. Tämä tapahtuu käyttämällä `linestyle` avainsanaa:

```
IDL> y2=sin(x+1)
IDL> oplot,x,y2,linestyle=1
```

Mikä tulostaa funktion y2=sin(x+1) kuvaajan pisteviivoituksella samaan kuvaan kuin edellä. Jos viivan piirtotyyliä ei anna, niin IDL käyttää oletusarvoa 0 = yhtenäinen viiva.

Joskus kun esimerkiksi halutaan plotata yksittäisiä mittauspisteitä joita on harvassa, on pisteitä yhdistävän käyrän sijasta mielekkäämpää plotata yksittäisiä pisteitä. Tämä tapahtuu käyttämällä `plot`- ja `oplot`-käskyissä `psym` avainsanaa. Esim. `psym=3` piirtää yksittäisiä pisteitä, `psym=2` tähtiä. Oletusarvo 0 vastaa taas pisteitä yhdistävää viivaa.

Piirrettäviin kuviin ja niiden akseleille voidaan antaa otsikkoja käyttämällä avainsanoja `title`, `xtitle` ja `ytitle`: `plot,x,y,title='Y=SIN(X),xtitle='X',ytitle='Y'`.

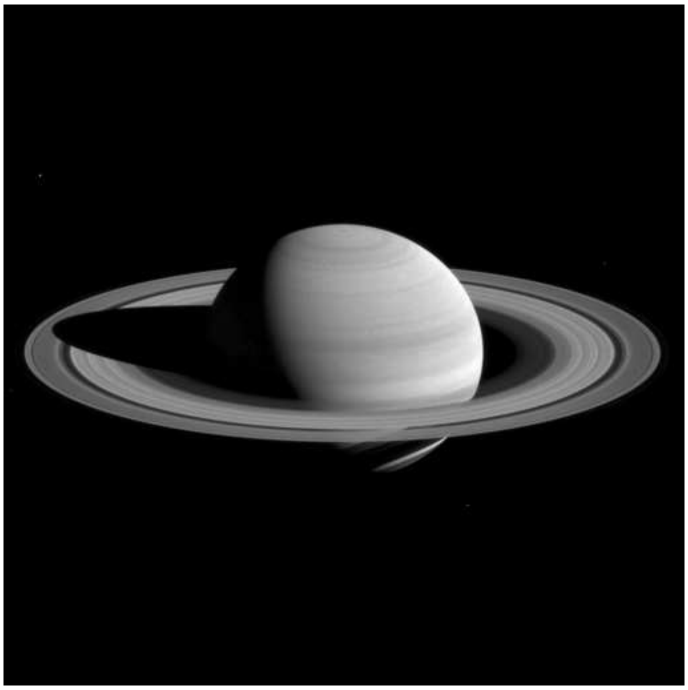
IDL skaalaa plottausalueen automaattisesti syötetyn datan perusteella. Itse plottausalueen voi määrittää käyttämällä avainsanoja `xrange` ja `yrange`:

`plot,x,y,xrange=[0,1],yrange=[0,1.5]`.

Avainsanat voidaan kirjoittaa myös lyhyemmin `xr` ja `yr`. Vaikka plottausalueen antaisikin edellä kuvatulla tavalla, niin IDL:llä on edelleen taipumus tehdä joskus esteettiseksi kokemiaan ratkaisuja ja poiketa hieman annetuista rajoista. IDL;n saa pakotettua käyttämään täsmälleen annettuja rajoja asettamalla plottauskäskyn yhteydessä

`plot,x,y,xr=[0,1.05],yr=[0,1.55],xstyle=1,ystyle=1`.

Useat plottauskäskyjen yhteydessä toimivat avainsanat toimivat myös muiden graafiseen esittämiseen liittyvien komentojen, kuten `contour` kanssa.



Kuva 2: Saturnus Cassinin kuvaamana (`tvscl,image`)

Plottaukseen liittyy joitakin systeemimuuttujia, tällainen on esimerkiksi `!p.multi`, jota voidaan käyttää useamman kuvan plottaamiseen samaa ikkunaan.

`!p.multi=[0,3,2]` määrittelee ikkunan siten, että siihen voi plotata kuusi kuvaa, vaakasarakkeita on kolme ja pystysarakkeita kaksi. Myös useita plottauksen avainsanoista voidaan määrittää systeemimuuttijina jolloin niiden antaminen `plot`-käskyn yhteydessä on tarpeetonta, esim. `!x.range=[0,1.5]` tai `!y.title='Y'`.

Plottaamisessa on myös mahdollista käyttää värejä. Tällöin on määriteltävä väripaletti. Tästä enemmän harjoituksissa.

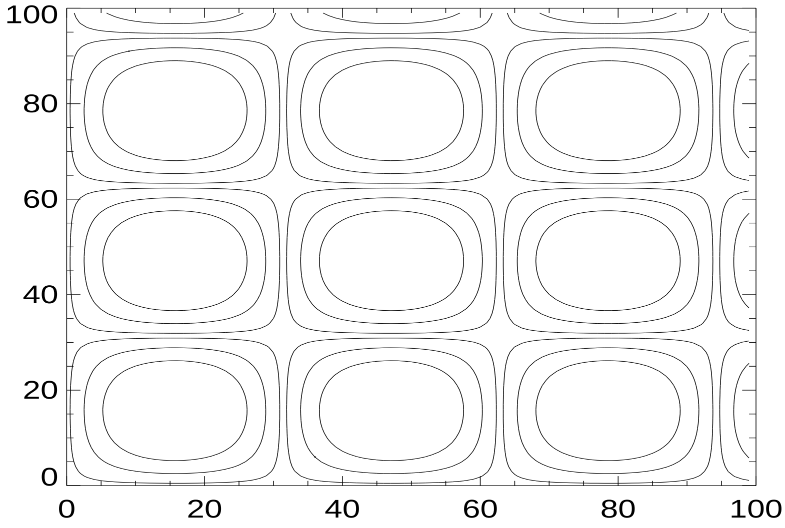
3.2 Kaksiulotteisen datan graafinen esittäminen

Usein data on kaksi- tai useampiulotteista ja `plot`-käskyllä siitä voidaan kerrallaan tutkia vain hyvin kapeaa siivua.

Yksi esimerkki kaksiulotteisesta datasta ovat harmaasävykuvat (värikuvissa saatetaan käyttää useampia ulottuvuuksia väripaletin luomiseen).

Käytetään seuraavassa IDL:n *SAVE* komennolla tiedostoon 'saturnus.save' talletettua Cassini-kuvaa. Luetaan tiedosto *RESTORE* komennolla. Tämän jälkeen voidaan `help`-käskyllä tutkia kuvan koko ja avataan sen kokoinen ikkuna. Lopuksi näytetään kuva `tvscl`-komennolla:

```
IDL> restore,'saturn.save'
IDL> help,image
image           BYTE      = Array(512, 512)
IDL> window,/free,xs=512,ys=512
IDL> tvscl,image
```



Kuva 3: Kontuurikäyrät `contour,z`

Luodaan seuraavaksi keinotekoinen kaksiulotteinen data:

```
IDL> x=findgen(100)/10.0
IDL> y=sin(x)
IDL> z=y#y
```

Tätä dataa voidaan havainnollistaa esim. kontuurikäyrillä tai pinnoilla. Tulostetaan esimerkiksi kontuurikäyrät:

```
IDL> contour,z,levels=[-0.5,-0.25,-0.05,0.05,0.25,0.5]
```

Pintojen tulostamiseen IDL:ssä on useampia komentoja. `surface`-komennolla saadaan "kanaverkko"-pinta:

Varjostettu pinta puolestaan saadaan seuraavasti:

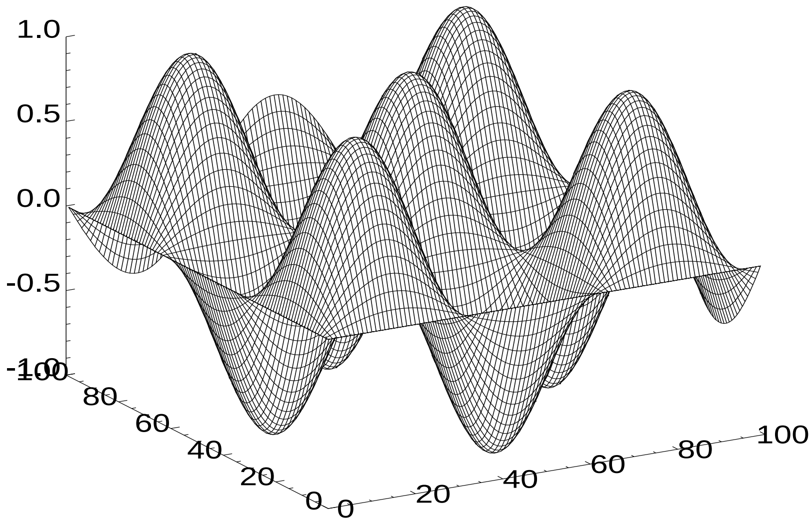
```
IDL> shade_surf,z
IDL> shade_surf,z,ax=70
```

Jälkimmäinen komento tulostaa pinnan eri suunnasta kuin ensimmäinen komento.

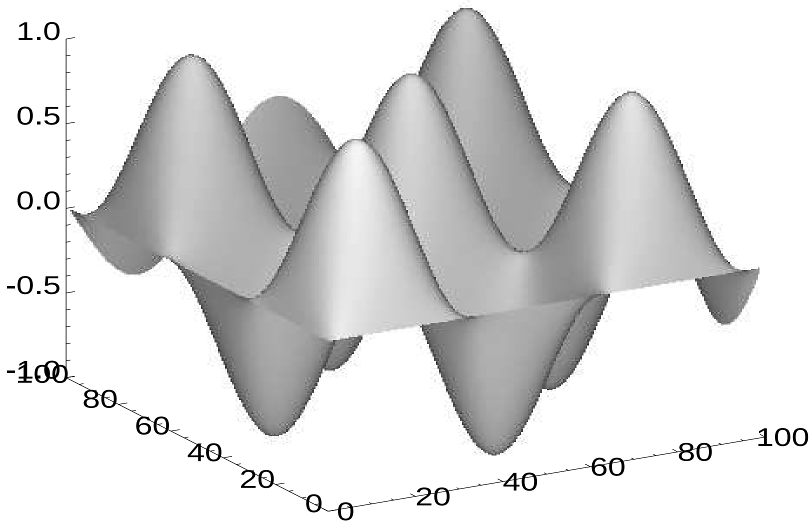
Pinnoista ja kuvista saa usein enemmän irti jos käyttää harmaasävyjen sijaan väripaletteja. Väripaletin voi valita seuraavasti:

```
loadct,3
```

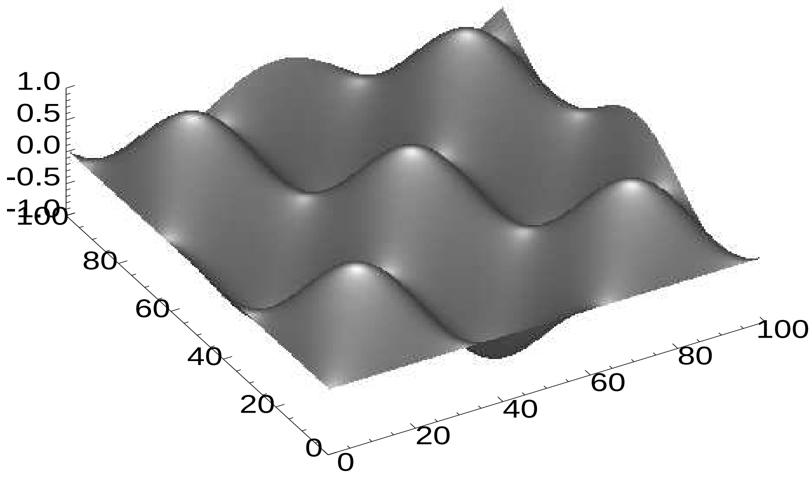
mikä ottaa käyttöön väripaletin nro 3. On myös valikkopohjainen ohjelma nimeltä `xloadct`, jolla voi valita paletteja ja vaikuttaa värien skaalauksiin jne.



Kuva 4: Pinta `surface,z`-käskyllä tulostettuna



Kuva 5: Varjostettu pinta `shade_surf,z`



Kuva 6: Varjostettu pinta toisesta suunnasta

4 Ohjelmat, aliohjelmat ja funktiot

Edellä on käsitelty IDL:n interaktiivista käyttöä. On ilmeistä, että monimutkaisten operaatioiden suorittaminen interaktiivisesti ei ole kovin käytännöllistä. Tätä varten on olemassa pääohjelmat, aliohjelmat ja funktiot. Samoin usein tehtävistä toiminnoista kannattaa kirjoittaa ohjelmat, joita voi näppärästi kutsua komentoriviltä käsin. Ohjelmien ja aliohjelmien kirjoittamisesta ja kutsumisesta enemmän erillisessä monisteessa.

Omien ohjelmien kirjoittamista harjoitellaan harjoituksissa.

5 Ohjelmakirjastot, manuaalit