GDL – GNU Data Language

presented by Sylwester Arabas
(slayoo@igf.fuw.edu.pl)

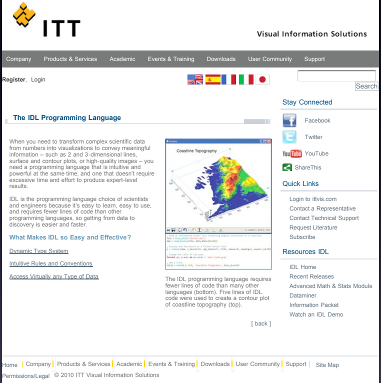
The GDL team:
Marc Schellens, Alain Coulais, Joel Gales, Sylwester Arabas,
and many, many more volunteers around the world!

(Marc is the primary author and the maintainer of GDL)

Free and Open Source Developers' European Meeting
Brussels, February 5th 2011

What's GDL (and IDL/PV-WAVE)

http://www.ittvis.com/



The IDL Programming Language

When you need to transform complex scientific data from numbers into visualizations to convey meaningful information – such as 2 and 3-dimensional lines, surface and contour plots, or high-quality images – you need a programming language that is intuitive and powerful at the same time, and one that doesn't require excessive time and effort to produce expert-level results.

IDL is the programming language choice of scientists and engineers because it's easy to learn, easy to use, and requires fewer lines of code than other programming languages, so getting from data to discovery is easier and faster.

What Makes IDL so Easy and Effective?

Dynamic Type System

Intuitive Rules and Conventions

Access Virtually any Type of Data

The IDL programming language requires fewer lines of code than many other languages (bottom). Five lines of IDL code were used to create a contour plot of coastline topography (top).

[back]

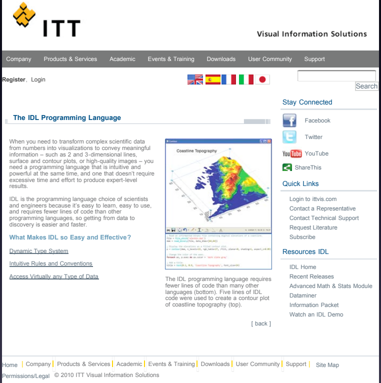
Stay Connected
Facebook
Twitter
YouTube
ShareThis
Quick Links
Login to ittvis.com
Contact a Representative
Contact Technical Support
Request Literature
Subscribe
Resources IDL
IDL Home
Recent Releases
Advanced Math & Stats Module
Dataminer
Information Packet
Watch an IDL Demo
Home | Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support | Site Map
Permissions/Legal © 2010 ITT Visual Information Solutions

- GDL[a] is developed with the aim of providing a free/libre/open-source drop-in replacement for IDL®
- IDL (ITT VIS Interactive Data Language):
  - is a tool for data analysis and visualisation
  - is a programming language ('77) (cf. archives of comp.lang.idl-pvwave)
  - is a popular software package in astrophysics, atmospheric physics, hyperspectral and medical imaging (in some cases a de facto standard)
  - is proprietary and expensive
  - is related with GDL as Matlab with Octave/Scilab, etc.

[a] despite its name, GDL is not an official GNU package yet

What's GDL (and IDL/PV-WAVE)

http://www.ittvis.com/



ITT Visual Information Solutions

Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support

Register, Login

Search

The IDL Programming Language

When you need to transform complex scientific data from numbers into visualizations to convey meaningful information – such as 2 and 3-dimensional lines, surface and contour plots, or high-quality images – you need a programming language that is intuitive and powerful at the same time, and one that doesn't require excessive time and effort to produce expert-level results.

IDL is the programming language choice of scientists and engineers because it's easy to learn, easy to use, and requires fewer lines of code than other programming languages, so getting from data to discovery is easier and faster.

The IDL programming language requires fewer lines of code than many other languages (bottom). Five lines of IDL code were used to create a contour plot of coastline topography (top).

What Makes IDL so Easy and Effective?

Dynamic Type System

Intuitive Rules and Conventions

Access Virtually any Type of Data

[back]

Stay Connected

Facebook

Twitter

YouTube

ShareThis

Quick Links

Login to ittvis.com

Contact a Representative

Contact Technical Support

Request Literature

Subscribe

Resources IDL

IDL Home

Recent Releases

Advanced Math & Stats Module

Dataminer

Information Packet

Watch an IDL Demo

Home | Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support | Site Map

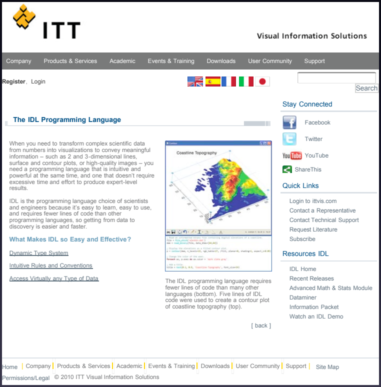
Permissions/Legal © 2010 ITT Visual Information Solutions

- GDL[a] is developed with the aim of providing a free/libre/open-source drop-in replacement for IDL®
- IDL (ITT VIS Interactive Data Language):
  - is a tool for data analysis and visualisation
  - is a programming language ('77) (cf. archives of comp.lang.idl-pvwave)
  - is a popular software package in astrophysics, atmospheric physics, hyperspectral and medical imaging (in some cases a de facto standard)
  - is proprietary and expensive
  - is related with GDL as MatLab with Octave/Scilab, etc

[a] despite its name, GDL is not an official GNU package yet

What's GDL (and IDL/PV-WAVE)

http://www.ittvis.com/


ITT
Visual Information Solutions
Company Products & Services Academic Events & Training Downloads User Community Support
Register, Login
Search
Facebook
Twitter
YouTube
ShareThis
Coastline Topography
[back]

The IDL Programming Language

When you need to transform complex scientific data from numbers into visualizations to convey meaningful information – such as 2 and 3-dimensional lines, surface and contour plots, or high-quality images – you need a programming language that is intuitive and powerful at the same time, and one that doesn't require excessive time and effort to produce expert-level results.

IDL is the programming language choice of scientists and engineers because it's easy to learn, easy to use, and requires fewer lines of code than other programming languages, so getting from data to discovery is easier and faster.

What Makes IDL so Easy and Effective?

Dynamic Type System

Intuitive Rules and Conventions

Access Virtually any Type of Data

The IDL programming language requires fewer lines of code than many other languages (bottom). Five lines of IDL code were used to create a contour plot of coastline topography (top).

Stay Connected

Quick Links

- Login to ittvis.com
- Contact a Representative
- Contact Technical Support
- Request Literature
- Subscribe

Resources IDL

- IDL Home
- Recent Releases
- Advanced Math & Stats Module
- Dataminer
- Information Packet
- Watch an IDL Demo

Home | Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support | Site Map

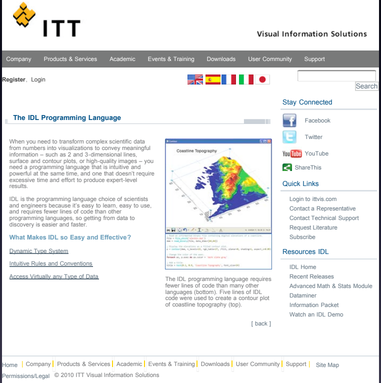
Permissions/Legal © 2010 ITT Visual Information Solutions

- GDL[a] is developed with the aim of providing a free/libre/open-source drop-in replacement for IDL®
- IDL (ITT VIS Interactive Data Language):
  - is a tool for data analysis and visualisation
  - is a programming language ('77)
    (cf. archives of comp.lang.idl-pvwave)
  - is a popular software package
    in astrophysics, atmospheric physics,
    hyperspectral and medical imaging
    (in some cases a de facto standard)
  - is proprietary and expensive
  - is related with GDL as
    Matlab with Octave/Scilab, etc.

[a] despite its name, GDL is not an official GNU package yet

What's GDL (and IDL/PV-WAVE)

http://www.ittvis.com/



ITT Visual Information Solutions

Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support

Register, Login

Search

The IDL Programming Language

When you need to transform complex scientific data from numbers into visualizations to convey meaningful information – such as 2 and 3-dimensional lines, surface and contour plots, or high-quality images – you need a programming language that is intuitive and powerful at the same time, and one that doesn't require excessive time and effort to produce expert-level results.

IDL is the programming language choice of scientists and engineers because it's easy to learn, easy to use, and requires fewer lines of code than many other programming languages, so getting from data to discovery is easier and faster.

What Makes IDL so Easy and Effective?

Dynamic Type System

Intuitive Rules and Conventions

Access Virtually any Type of Data

The IDL programming language requires fewer lines of code than many other languages (bottom). Five lines of IDL code were used to create a contour plot of coastline topography (top).

[back]

Stay Connected

Facebook

Twitter

YouTube

ShareThis

Quick Links

Login to ittvis.com

Contact a Representative

Contact Technical Support

Request Literature

Subscribe

Resources IDL

IDL Home

Recent Releases

Advanced Math & Stats Module

Dataminer

Information Packet

Watch an IDL Demo

Home | Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support | Site Map

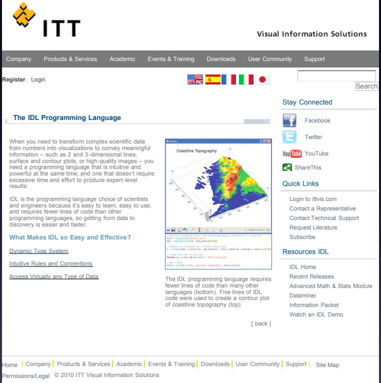
Permissions/Legal © 2010 ITT Visual Information Solutions

- GDL[a] is developed with the aim of providing a free/libre/open-source drop-in replacement for IDL®
- IDL (ITT VIS Interactive Data Language):
  - is a tool for data analysis and visualisation
  - is a programming language ('77) (cf. archives of comp.lang.idl-pvwave)
  - is a popular software package in astrophysics, atmospheric physics, hyperspectral and medical imaging (in some cases a de facto standard)
  - is proprietary and expensive
  - is related with GDL as Matlab with Octave/Scilab, etc

[a] despite its name, GDL is not an official GNU package yet

What's GDL (and IDL/PV-WAVE)

http://www.ittvis.com/



ITT

Visual Information Solutions

Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support

Register, Login

Search

The IDL Programming Language

When you need to transform complex scientific data from numbers into visualizations to convey meaningful information – such as 2 and 3-dimensional lines, surface and contour plots, or high-quality images – you need a programming language that is intuitive and powerful at the same time, and one that doesn't require excessive time and effort to produce expert-level results.

IDL is the programming language choice of scientists and engineers because it's easy to learn, easy to use, and requires fewer lines of code than other programming languages, so getting from data to discovery is easier and faster.

What Makes IDL so Easy and Effective?

Dynamic Type System

Intuitive Rules and Conventions

Access Virtually any Type of Data

The IDL programming language requires fewer lines of code than many other languages (bottom). Five lines of IDL code were used to create a contour plot of coastline topography (top).

[back]

Stay Connected

- Facebook
- Twitter
- YouTube
- ShareThis

Quick Links

- Login to ittvis.com
- Contact a Representative
- Contact Technical Support
- Request Literature
- Subscribe

Resources IDL

- IDL Home
- Recent Releases
- Advanced Math & Stats Module
- Dataminer
- Information Packet
- Watch an IDL Demo

Home | Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support | Site Map

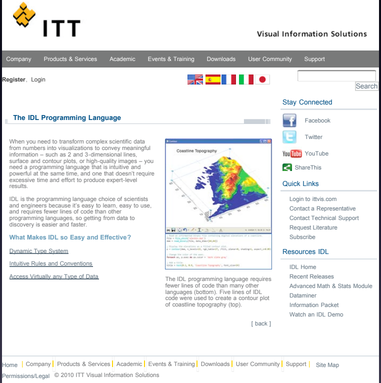
Permissions/Legal © 2010 ITT Visual Information Solutions

- GDL[a] is developed with the aim of providing a free/libre/open-source drop-in replacement for IDL®
- IDL (ITT VIS Interactive Data Language):
  - is a tool for data analysis and visualisation
  - is a programming language ('77) (cf. archives of comp.lang.idl-pvwave)
  - is a popular software package in astrophysics, atmospheric physics, hyperspectral and medical imaging (in some cases a de facto standard)
  - is proprietary and expensive
  - is related with GDL as Matlab with Octave/Scilab, etc

[a] despite its name, GDL is not an official GNU package yet

What's GDL (and IDL/PV-WAVE)

http://www.ittvis.com/



The IDL Programming Language

When you need to transform complex scientific data from numbers into visualizations to convey meaningful information – such as 2 and 3-dimensional lines, surface and contour plots, or high-quality images – you need a programming language that is intuitive and powerful at the same time, and one that doesn't require excessive time and effort to produce expert-level results.

IDL is the programming language choice of scientists and engineers because it's easy to learn, easy to use, and requires fewer lines of code than other programming languages, so getting from data to discovery is easier and faster.

What Makes IDL so Easy and Effective?

Dynamic Type System

Intuitive Rules and Conventions

Access Virtually any Type of Data

The IDL programming language requires fewer lines of code than many other languages (bottom). Five lines of IDL code were used to create a contour plot of coastline topography (top).

[back]

Stay Connected

Facebook
Twitter
YouTube
ShareThis

Quick Links

Login to ittvis.com
Contact a Representative
Contact Technical Support
Request Literature
Subscribe

Resources IDL

IDL Home
Recent Releases
Advanced Math & Stats Module
Dataminer
Information Packet
Watch an IDL Demo

Home | Company | Products & Services | Academic | Events & Training | Downloads | User Community | Support | Site Map

Permissions/Legal © 2010 ITT Visual Information Solutions

- GDL[a] is developed with the aim of providing a free/libre/open-source drop-in replacement for IDL®
- IDL (ITT VIS Interactive Data Language):
  - is a tool for data analysis and visualisation
  - is a programming language ('77) (cf. archives of comp.lang.idl-pvwave)
  - is a popular software package in astrophysics, atmospheric physics, hyperspectral and medical imaging (in some cases a de facto standard)
  - is proprietary and expensive
  - is related with GDL as Matlab with Octave/Scilab, etc.

[a] despite its name, GDL is not an official GNU package yet

Reasons behind development & use of GDL

- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:

  gsfc.nasa.gov ”The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL

  vapor.ucar.edu [VAPOR (BSD License)] ”is closely coupled with (but does not require) [...] Interactive Data Language (IDL)”

  hdfeos.org ”Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available”

  eumetsat.int ”EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ...”

  lmsal.com ”SolarSoft is ... data analysis environment for Solar Physics ... IDL based”
- Possibility to modify the source code (several published examples)
- Just for fun :)
- ...

Reasons behind development & use of GDL

- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:

  gsfc.nasa.gov ”The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL

  vapor.ucar.edu [VAPOR (BSD License)] ”is closely coupled with (but does not require) [...] Interactive Data Language (IDL)”

  hdfeos.org ”Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available”

  eumetsat.int ”EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ...”

  lmsal.com ”SolarSoft is ... data analysis environment for Solar Physics ... IDL based”
- Possibility to modify the source code (several published examples)
- Just for fun :)
- . . .

Reasons behind development & use of GDL

- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:
  - gsfc.nasa.gov ”The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL
  - vapor.ucar.edu [VAPOR (BSD License)] ”is closely coupled with (but does not require) [...] Interactive Data Language (IDL)”
  - hdfeos.org ”Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available”
  - eumetsat.int ”EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ...”
  - lmsal.com ”SolarSoft is ... data analysis environment for Solar Physics ... IDL based”
- Possibility to modify the source code (several published examples)
- Just for fun :)
- . . .

Reasons behind development & use of GDL

- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:

  gsfc.nasa.gov "The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL

  vapor.ucar.edu [VAPOR (BSD License)] "is closely coupled with (but does not require) [...] Interactive Data Language (IDL)"

  hdfeos.org "Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available"

  eumetsat.int "EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ..."

  lmsal.com "SolarSoft is ... data analysis environment for Solar Physics ... IDL based"
- Possibility to modify the source code (several published examples)
- Just for fun :)
- . . .

Reasons behind development & use of GDL

- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:

  gsfc.nasa.gov "The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL

  vapor.ucar.edu [VAPOR (BSD License)] "is closely coupled with (but does not require) [...] Interactive Data Language (IDL)"

  hdfeos.org "Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available"

  eumetsat.int "EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ..."

  lmsal.com "SolarSoft is ... data analysis environment for Solar Physics ... IDL based"
- Possibility to modify the source code (several published examples)
- Just for fun :)
- . . .

Reasons behind development & use of GDL

- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:

  gsfc.nasa.gov "The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL

  vapor.ucar.edu [VAPOR (BSD License)] "is closely coupled with (but does not require) [...] Interactive Data Language (IDL)"

  hdfeos.org "Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available"

  eumetsat.int "EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ..."

  lmsal.com "SolarSoft is ... data analysis environment for Solar Physics ... IDL based"

- Possibility to modify the source code (several published examples)
- Just for fun :)
- . . .

Reasons behind development & use of GDL

- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:

  gsfc.nasa.gov "The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL

  vapor.ucar.edu [VAPOR (BSD License)] "is closely coupled with (but does not require) [...] Interactive Data Language (IDL)"

  hdfeos.org "Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available"

  eumetsat.int "EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ..."

  lmsal.com "SolarSoft is ... data analysis environment for Solar Physics ... IDL based"

- Possibility to modify the source code (several published examples)
- Just for fun :)
- ...

Reasons behind development & use of GDL

- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:

  gsfc.nasa.gov ”The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL

  vapor.ucar.edu [VAPOR (BSD License)] ”is closely coupled with (but does not require) [...] Interactive Data Language (IDL)”

  hdfeos.org ”Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available”

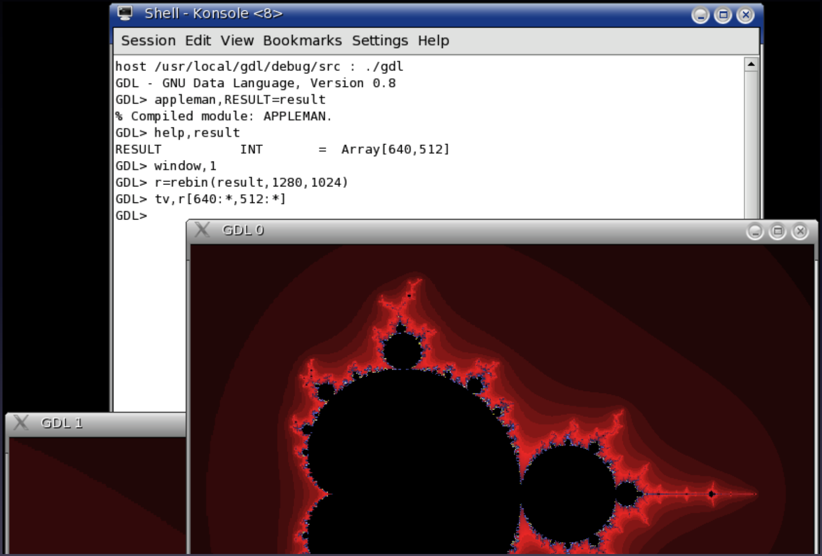
  eumetsat.int ”EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ...”

  lmsal.com ”SolarSoft is ... data analysis environment for Solar Physics ... IDL based”
- Possibility to modify the source code (several published examples)
- Just for fun :)
- ...

Reasons behind development & use of GDL

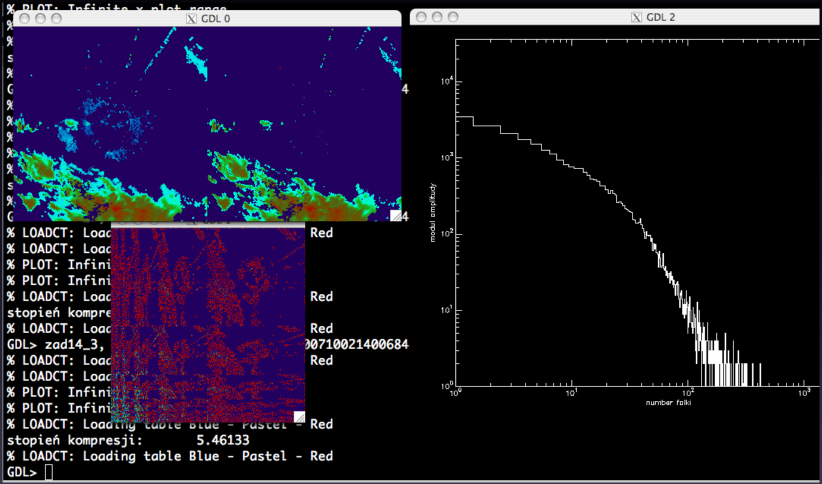
- IDL license price and limitations (e.g. number of simultaneous processes)
- Existence free/open-source scientific software that make use or rely on IDL:
  - gsfc.nasa.gov ”The IDL Astronomy Library procedures are in the public domain... [written] in the commercial language IDL
  - vapor.ucar.edu [VAPOR (BSD License)] ”is closely coupled with (but does not require) [...] Interactive Data Language (IDL)”
  - hdfeos.org ”Comprehensive NCL/IDL/MATLAB examples for many NASA HDF4 and HDF-EOS2 products are available”
  - eumetsat.int ”EUMETSAT makes available the following set of interactive tools and software programs ... IDL has been selected to read, process, and analyse the EPS products ...”
  - lmsal.com ”SolarSoft is ... data analysis environment for Solar Physics ... IDL based”
- Possibility to modify the source code (several published examples)
- Just for fun :)
- . . .

GDL rendering the Mandelbrot[1] set



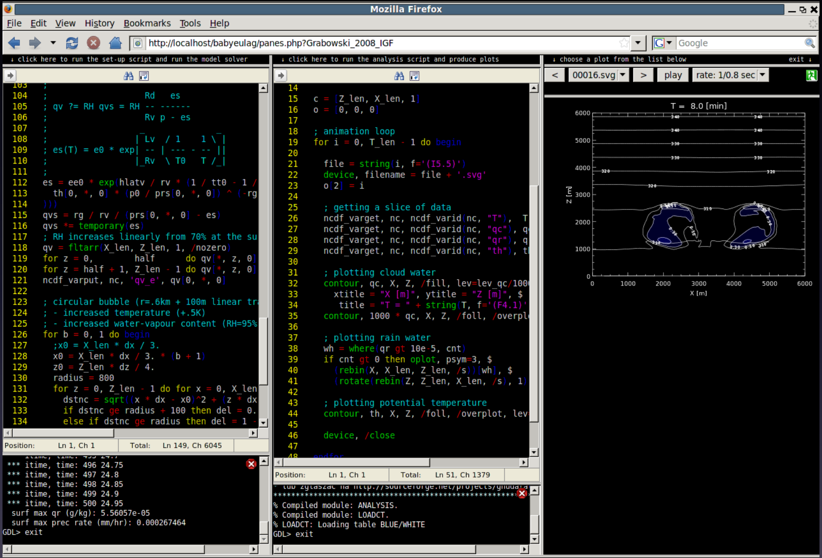
```
host /usr/local/gdl/debug/src : ./gdl
GDL - GNU Data Language, Version 0.8
GDL> appleman,RESULT=result
% Compiled module: APPLEMAN.
GDL> help,result
RESULT          INT       =  Array[640,512]
GDL> window,1
GDL> r=rebin(result,1280,1024)
GDL> tv,r[640:*,512:*]
GDL>
```

[1] Benoit B. Mandelbrot: 20 November 1924 (Warsaw, Poland) – 14 October 2010 (Cambridge, MA, USA)



```
% PLOT: Infinite x plot range
%
%
s
%
G                                                     4
%
%
%
%
%
s
%
G                                                     4
% LOADCT: Loac                                 Red
% LOADCT: Loac
% PLOT: Infini
% PLOT: Infini
% LOADCT: Loac                                 Red
stopień kompre
% LOADCT: Loac                                 Red
GDL> zad14_3,                               00710021400684
% LOADCT: Loac                                 Red
% LOADCT: Loac
% PLOT: Infini
% PLOT: Infini
% LOADCT: Loading table Blue - Pastel - Red
stopień kompresji:        5.46133
% LOADCT: Loading table Blue - Pastel - Red
GDL>
```

GDL reading weather-radar data (HDF5) & doing wavelet analysis


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/babyeulag/panes.php?Grabowski_2008_IGF
Google
↓ click here to run the set-up script and run the model solver
↓ click here to run the analysis script and produce plots
↓ choose a plot from the list below
exit ↓
<
00016.svg
>
play
rate: 1/0.8 sec
T = 8.0 [min]
Z [m]
X [m]
0
1000
2000
3000
4000
5000
6000
340
330
320
310
0.30
0.10
0.20

```
;
;                   Rd   es
; qv ?= RH qvs = RH -- ------
;                   Rv p - es
;                  _              _
;                 | Lv  / 1    1 \ |
; es(T) = e0 * exp| -- | --- - -- ||
;                 |_Rv  \ T0   T /_|
;
es = ee0 * exp(hlatv / rv * (1 / tt0 - 1 /
  th[0, *, 0] * (p0 / prs[0, *, 0]) ^ (-rg
)))
qvs = rg / rv / (prs[0, *, 0] - es)
qvs *= temporary(es)
; RH increases linearly from 70% at the su
qv = fltarr(X_len, Z_len, 1, /nozero)
for z = 0,        half      do qv[*, z, 0]
for z = half + 1, Z_len - 1 do qv[*, z, 0]
ncdf_varput, nc, 'qv_e', qv[0, *, 0]

; circular bubble (r=.6km + 100m linear tr
; - increased temperature (+.5K)
; - increased water-vapour content (RH=95%
for b = 0, 1 do begin
  ;x0 = X_len * dx / 3.
  x0 = X_len * dx / 3. * (b + 1)
  z0 = Z_len * dz / 4.
  radius = 800
  for z = 0, Z_len - 1 do for x = 0, X_len
    dstnc = sqrt((x * dx - x0)^2 + (z * dx
    if dstnc ge radius + 100 then del = 0.
    else if dstnc ge radius then del = 1 -
```

Position: Ln 1, Ch 1 Total: Ln 149, Ch 6045

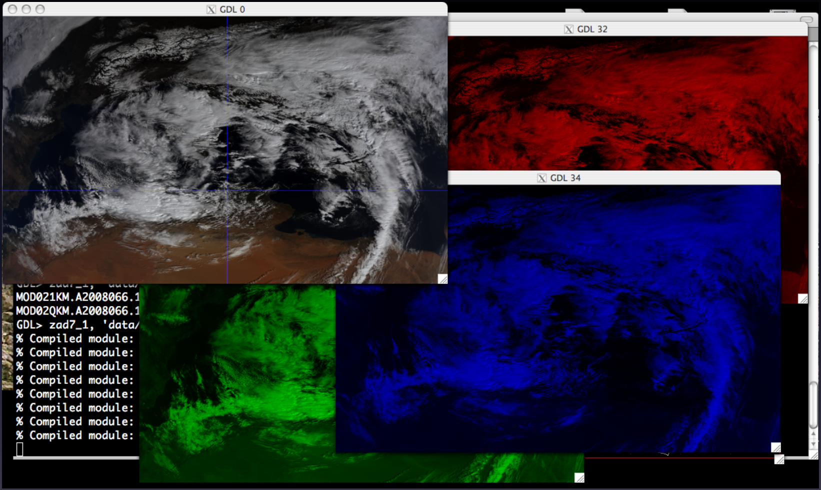
```
 itime, time: 495 24.7
*** itime, time: 496 24.75
*** itime, time: 497 24.8
*** itime, time: 498 24.85
*** itime, time: 499 24.9
*** itime, time: 500 24.95
 surf max qr (g/kg): 5.56057e-05
 surf max prec rate (mm/hr): 0.000267464
GDL> exit
```

```

c = [Z_len, X_len, 1]
o = [0, 0, 0]

; animation loop
for i = 0, T_len - 1 do begin

  file = string(i, f='(I5.5)')
  device, filename = file + '.svg'
  o[2] = i

  ; getting a slice of data
  ncdf_varget, nc, ncdf_varid(nc, "T"),  T
  ncdf_varget, nc, ncdf_varid(nc, "qc"), q
  ncdf_varget, nc, ncdf_varid(nc, "qr"), q
  ncdf_varget, nc, ncdf_varid(nc, "th"), t

  ; plotting cloud water
  contour, qc, X, Z, /fill, lev=lev_qc/100
    xtitle = "X [m]", ytitle = "Z [m]", $
     title = "T = " + string(T, f='(F4.1)'
  contour, 1000 * qc, X, Z, /foll, /overpl

  ; plotting rain water
  wh = where(qr gt 10e-5, cnt)
  if cnt gt 0 then oplot, psym=3, $
    (rebin(X, X_len, Z_len, /s))[wh], $
    (rotate(rebin(Z, Z_len, X_len, /s), 1)

  ; plotting potential temperature
  contour, th, X, Z, /foll, /overplot, lev

  device, /close

endfor
```

Position: Ln 1, Ch 1 Total: Ln 51, Ch 1379

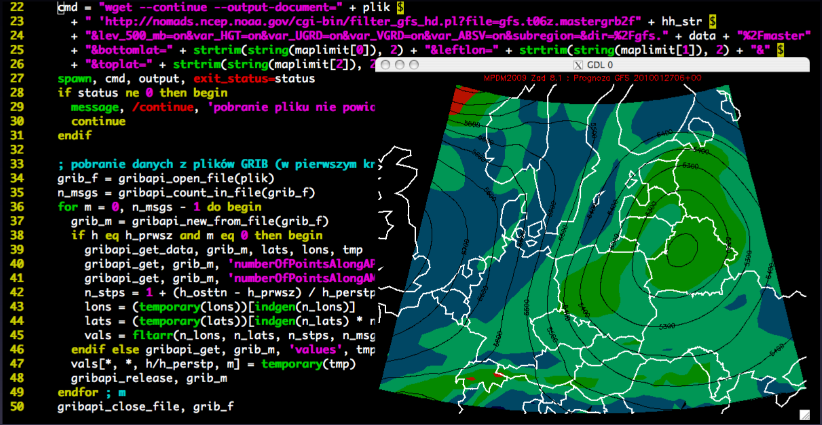
```
* lub zgłaszać na http://sourceforge.net/projects/gnudata
**********************************************************
% Compiled module: ANALYSIS.
% Compiled module: LOADCT.
% LOADCT: Loading table BLUE/WHITE
GDL> exit
```

GDL in a web interface generating SVG plots



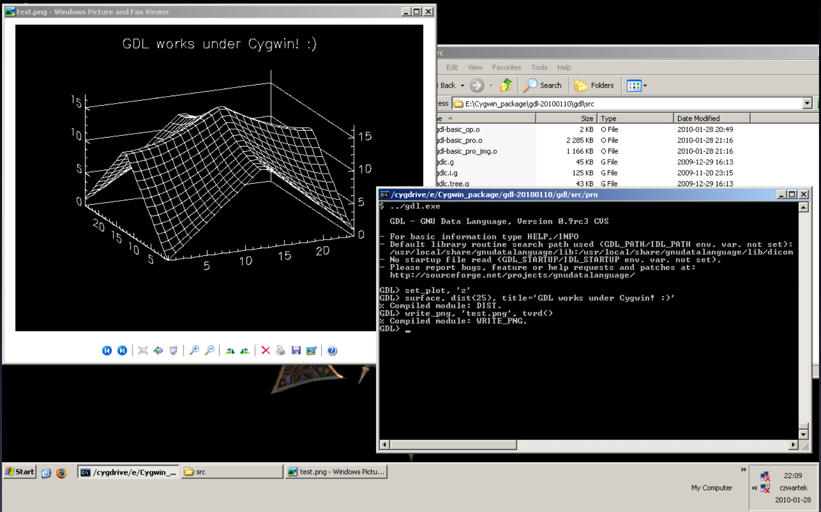
```
MOD021KM.A2008066.1
MOD02QKM.A2008066.1
GDL> zad7_1, 'data/
% Compiled module:
% Compiled module:
% Compiled module:
% Compiled module:
% Compiled module:
% Compiled module:
% Compiled module:
% Compiled module:
```

GDL plotting MODIS satellite images (reading data from HDF4)



```
cmd = "wget --continue --output-document=" + plik $
  + " 'http://nomads.ncep.noaa.gov/cgi-bin/filter_gfs_hd.pl?file=gfs.t06z.mastergrb2f" + hh_str $
  + "&lev_500_mb=on&var_HGT=on&var_UGRD=on&var_VGRD=on&var_ABSV=on&subregion=&dir=%2Fgfs." + data + "%2Fmaster"
  + "&bottomlat=" + strtrim(string(maplimit[0]), 2) + "&leftlon=" + strtrim(string(maplimit[1]), 2) + "&" $
  + "&toplat=" + strtrim(string(maplimit[2]), 2
spawn, cmd, output, exit_status=status
if status ne 0 then begin
  message, /continue, 'pobranie pliku nie powic
  continue
endif

; pobranie danych z plików GRIB (w pierwszym kr
grib_f = gribapi_open_file(plik)
n_msgs = gribapi_count_in_file(grib_f)
for m = 0, n_msgs - 1 do begin
  grib_m = gribapi_new_from_file(grib_f)
  if h eq h_prwsz and m eq 0 then begin
    gribapi_get_data, grib_m, lats, lons, tmp
    gribapi_get, grib_m, 'numberOfPointsAlongAP
    gribapi_get, grib_m, 'numberOfPointsAlongM
    n_stps = 1 + (h_osttn - h_prwsz) / h_perstp
    lons = (temporary(lons))[indgen(n_lons)]
    lats = (temporary(lats))[indgen(n_lats) * n
    vals = fltarr(n_lons, n_lats, n_stps, n_msg
  endif else gribapi_get, grib_m, 'values', tmp
  vals[*, *, h/h_perstp, m] = temporary(tmp)
  gribapi_release, grib_m
endfor ; m
gribapi_close_file, grib_f
```

GDL rendering weather forecast animation (reading from GRIB)



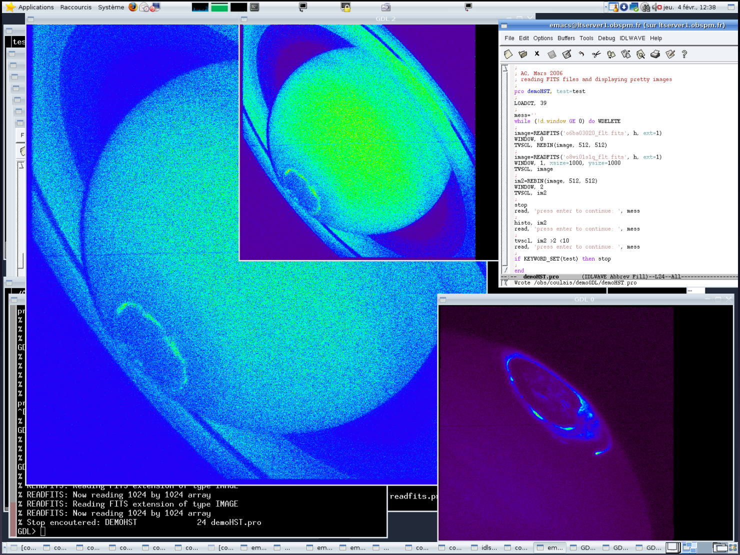
E:\Cygwin_package\gdl-20100110\gdl\src

Name	Size	Type	Date Modified
gdl-basic_op.o	2 KB	O File	2010-01-28 20:49
gdl-basic_pro.o	2 285 KB	O File	2010-01-28 21:16
gdl-basic_pro_jmg.o	1 166 KB	O File	2010-01-28 21:16
gdlc.g	45 KB	G File	2009-12-29 16:13
gdlc.i.g	125 KB	G File	2009-11-20 23:15
gdlc.tree.g	43 KB	G File	2009-12-29 16:13

/cygdrive/e/Cygwin_package/gdl-20100110/gdl/src/pro

```
$ ../gdl.exe

  GDL - GNU Data Language, Version 0.9rc3 CVS

- For basic information type HELP,/INFO
- Default library routine search path used (GDL_PATH/IDL_PATH env. var. not set):
  /usr/local/share/gnudatalanguage/lib:/usr/local/share/gnudatalanguage/lib/dicom
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not set).
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

GDL> set_plot, 'z'
GDL> surface, dist(25), title='GDL works under Cygwin! :)'
% Compiled module: DIST.
GDL> write_png, 'test.png', tvrd()
% Compiled module: WRITE_PNG.
GDL> _
```

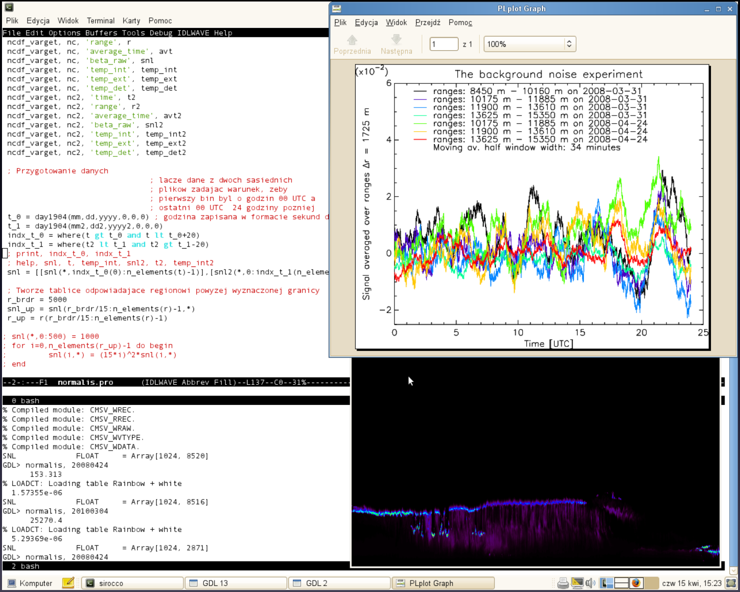
GDL writing a 3D surface plot to a PNG file under Cygwin (by Mateusz Turcza)


Applications Raccourcis Système
jeu. 4 févr., 12:38
GDL 2
GDL 0
emacs@ltserver1.obspm.fr (sur ltserver1.obspm.fr)
File Edit Options Buffers Tools Debug IDLWAVE Help

```
;
; AC, Mars 2006
; reading FITS files and displaying pretty images
;
pro demoHST, test=test
;
LOADCT, 39
;
mess=''
while (!d.window GE 0) do WDELETE
;
image=READFITS('o6ba03020_flt.fits', h, ext=1)
WINDOW, 0
TVSCL, REBIN(image, 512, 512)
;
image=READFITS('o8wi01s1q_flt.fits', h, ext=1)
WINDOW, 1, xsize=1000, ysize=1000
TVSCL, image
;
im2=REBIN(image, 512, 512)
WINDOW, 2
TVSCL, im2
;
stop
read, 'press enter to continue: ', mess
;
histo, im2
read, 'press enter to continue: ', mess
;
tvscl, im2 >2 <10
read, 'press enter to continue: ', mess
;
if KEYWORD_SET(test) then stop
;
end
--:--  demoHST.pro      (IDLWAVE Abbrev Fill)--L24--All-----------------
Wrote /obs/coulais/demoGDL/demoHST.pro
```

```
% READFITS: Now reading 1024 by 1024 array
% READFITS: Reading FITS extension of type IMAGE
% READFITS: Now reading 1024 by 1024 array
% Stop encoutered: DEMOHST              24 demoHST.pro
GDL> 
```

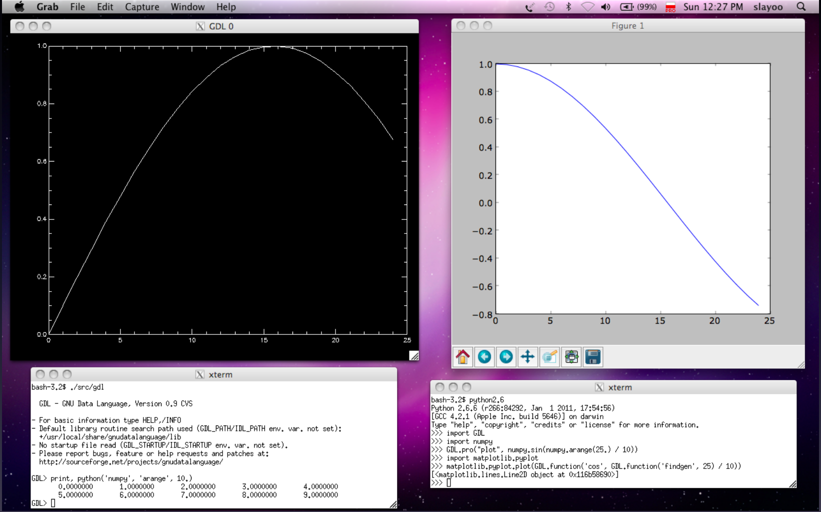
GDL rendering images of polar aurorae on Saturn (reading FITS) (by Renée Prangé & Laurent Pallier)



Plik Edycja Widok Terminal Karty Pomoc
File Edit Options Buffers Tools Debug IDLWAVE Help
PLplot Graph
Plik Edycja Widok Przejdź Pomoc
Poprzednia Następna
1 z 1 100%
The background noise experiment
(x10⁻²)
Signal averaged over ranges Δr = 1725 m
Time [UTC]
6
4
2
0
−2
0 5 10 15 20 25
ranges: 8450 m – 10160 m on 2008–03–31
ranges: 10175 m – 11885 m on 2008–03–31
ranges: 11900 m – 13610 m on 2008–03–31
ranges: 13625 m – 15350 m on 2008–03–31
ranges: 10175 m – 11885 m on 2008–04–24
ranges: 11900 m – 13610 m on 2008–04–24
ranges: 13625 m – 15350 m on 2008–04–24
Moving av. half window width: 34 minutes
Komputer
sirocco
GDL 13
GDL 2
PLplot Graph
czw 15 kwi, 15:23

```
 ncdf_varget, nc, 'range', r
 ncdf_varget, nc, 'average_time', avt
 ncdf_varget, nc, 'beta_raw', snl
 ncdf_varget, nc, 'temp_int', temp_int
 ncdf_varget, nc, 'temp_ext', temp_ext
 ncdf_varget, nc, 'temp_det', temp_det
 ncdf_varget, nc2, 'time', t2
 ncdf_varget, nc2, 'range', r2
 ncdf_varget, nc2, 'average_time', avt2
 ncdf_varget, nc2, 'beta_raw', snl2
 ncdf_varget, nc2, 'temp_int', temp_int2
 ncdf_varget, nc2, 'temp_ext', temp_ext2
 ncdf_varget, nc2, 'temp_det', temp_det2

 ; Przygotowanie danych
                                ; lacze dane z dwoch sasiednich
                                ; plikow zadajac warunek, zeby
                                ; pierwszy bin byl o godzin 00 UTC a
                                ; ostatni 00 UTC  24 godziny pozniej
 t_0 = day1904(mm,dd,yyyy,0,0,0) ; godzina zapisana w formacie sekund d
 t_1 = day1904(mm2,dd2,yyyy2,0,0,0)
 indx_t_0 = where(t gt t_0 and t lt t_0+20)
 indx_t_1 = where(t2 lt t_1 and t2 gt t_1-20)
 ; print, indx_t_0, indx_t_1
 ; help, snl, t, temp_int, snl2, t2, temp_int2
 snl = [[snl(*,indx_t_0(0):n_elements(t)-1)],[snl2(*,0:indx_t_1(n_eleme

 ; Tworze tablice odpowiadajace regionowi powyzej wyznaczonej granicy
 r_brdr = 5000
 snl_up = snl(r_brdr/15:n_elements(r)-1,*)
 r_up = r(r_brdr/15:n_elements(r)-1)

; snl(*,0:500) = 1000
; for i=0,n_elements(r_up)-1 do begin
;         snl(i,*) = (15*i)^2*snl(i,*)
; end

--2-:---F1  normalis.pro      (IDLWAVE Abbrev Fill)--L137--C0--31%---------

  0 bash
% Compiled module: CMSV_WREC.
% Compiled module: CMSV_RREC.
% Compiled module: CMSV_WRAW.
% Compiled module: CMSV_WVTYPE.
% Compiled module: CMSV_WDATA.
SNL             FLOAT     = Array[1024, 8520]
GDL> normalis, 20080424
      153.313
% LOADCT: Loading table Rainbow + white
  1.57355e-06
SNL             FLOAT     = Array[1024, 8516]
GDL> normalis, 20100304
      25270.4
% LOADCT: Loading table Rainbow + white
  5.29369e-06
SNL             FLOAT     = Array[1024, 2871]
GDL> normalis, 20080424
  2 bash
```

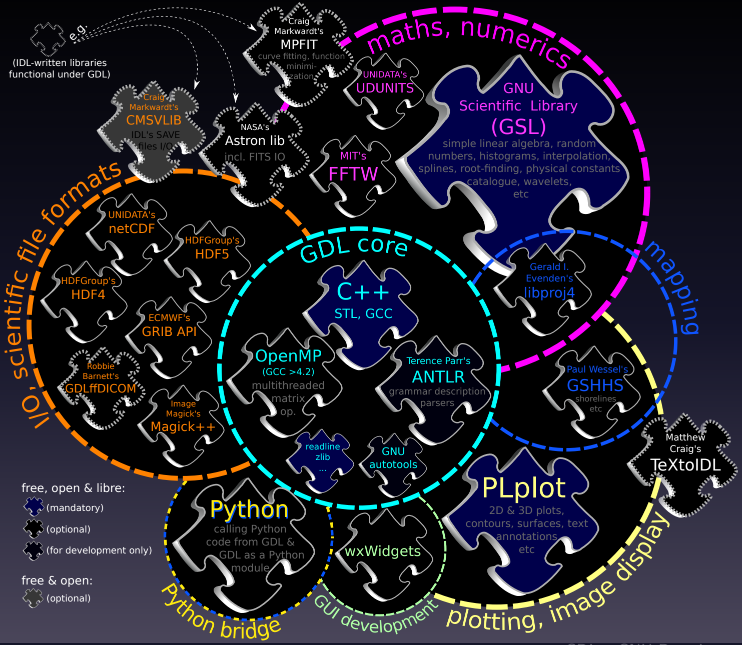
GDL & LIDAR data analysis (reading data from netCDF, by Michał Piądłowski)



```
bash-3.2$ ./src/gdl

  GDL - GNU Data Language, Version 0.9 CVS

- For basic information type HELP,/INFO
- Default library routine search path used (GDL_PATH/IDL_PATH env. var. not set):
  +/usr/local/share/gnudatalanguage/lib
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not set).
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

GDL> print, python('numpy', 'arange', 10.)
      0.0000000       1.0000000       2.0000000       3.0000000       4.0000000
      5.0000000       6.0000000       7.0000000       8.0000000       9.0000000
GDL> 
```

```
bash-3.2$ python2.6
Python 2.6.6 (r266:84292, Jan  1 2011, 17:54:56)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import GDL
>>> import numpy
>>> GDL.pro("plot", numpy.sin(numpy.arange(25.) / 10))
>>> import matplotlib.pyplot
>>> matplotlib.pyplot.plot(GDL.function('cos', GDL.function('findgen', 25) / 10))
[<matplotlib.lines.Line2D object at 0x116b58690>]
>>> 
```

Calling GDL from Python and vice versa (Numpy & matplotlib)



e.g.
(IDL-written libraries
functional under GDL)

Craig Markwardt's
MPFIT
curve fitting, function minimization

Craig Markwardt's
CMSVLIB
IDL's SAVE files I/O

NASA's
Astron lib
incl. FITS IO

maths, numerics

UNIDATA's
UDUNITS

GNU Scientific Library (GSL)
simple linear algebra, random numbers, histograms, interpolation, splines, root-finding, physical constants catalogue, wavelets, etc

MIT's
FFTW

I/O, scientific file formats

UNIDATA's
netCDF

HDFGroup's
HDF5

HDFGroup's
HDF4

ECMWF's
GRIB API

Robbie Barnett's
GDLffDICOM

Image Magick's
Magick++

GDL core

C++
STL, GCC

OpenMP
(GCC >4.2)
multithreaded matrix op.

Terence Parr's
ANTLR
grammar description parsers

readline
zlib
...

GNU
autotools

mapping

Gerald I. Evenden's
libproj4

Paul Wessel's
GSHHS
shorelines
etc

Matthew Craig's
TeXtoIDL

plotting, image display

PLplot
2D & 3D plots, contours, surfaces, text annotations
etc

GUI development

wxWidgets

Python bridge

Python
calling Python code from GDL & GDL as a Python module

free, open & libre:
(mandatory)
(optional)
(for development only)

free & open:
(optional)

	Arch (AUR)	Debian	Fedora	Fink	FreeBSD	Gentoo	Hmug	MacPorts	Ubuntu
GDL version:	0.9	0.9rc3	0.9	0.9	0.9	0.9	0.9	0.9	0.9rc3
features:									
FFTW	−	+	+	+	+	+	+	+	+
GSHHS	−	−	−	−	−	−	−	+	−
GRIB_API	−	−	+	−	−	−	−	+	−
HDF4	−	+	+	+	+	+	+	+	+
HDF5	+	+	+	+	+	+	+	+	+
ImageMagick	+	+	+	+	+	+	+	+	+
libproject	−	−	−	−	−	−	−	+	−
netCDF	+	+	+	+	+	+	−	+	+
GDL→Python	+	−	+	−	+	+	+	+	−
Python→GDL	−	−	+	−	−	+	−	−	−
UDUNITS-2	−	−	+	−	−	+	−	+	−
wxWidgets	+	−	+	−	+	+	−	+	−

- Big thanks to all packagers!!!

  (incl. Juan A. Añel, Markus Dittrich, Takeshi Enomoto, Sébastien Fabbro, Orlando Garcia Feal, Gaurav Khanna, Justin Lecher, Sebastien Maret, Lea Noreskal, Orion Poplawski, Marius Schamschula, Gürkan Sengün, Thierry Thomas, . . .)

- More help and feedback needed!
  - upgrades/enhancements to existing packages (Debian/Ubutnu!)
  - new packages (OpenSUSE, Homebrew, Cygwin, Slackware, Solaris, . . .)

	Arch (AUR)	Debian	Fedora	Fink	FreeBSD	Gentoo	Hmug	MacPorts	Ubuntu
GDL version:	0.9	0.9rc3	0.9	0.9	0.9	0.9	0.9	0.9	0.9rc3
features:									
FFTW	−	+	+	+	+	+	+	+	+
GSHHS	−	−	−	−	−	−	−	+	−
GRIB_API	−	−	+	−	−	−	−	+	−
HDF4	−	+	+	+	+	+	+	+	+
HDF5	+	+	+	+	+	+	+	+	+
ImageMagick	+	+	+	+	+	+	+	+	+
libproject	−	−	−	−	−	−	−	+	−
netCDF	+	+	+	+	+	+	−	+	+
GDL→Python	+	−	+	−	+	+	+	+	−
Python→GDL	−	−	+	−	−	+	−	−	−
UDUNITS-2	−	−	+	−	−	+	−	+	−
wxWidgets	+	−	+	−	+	+	−	+	−

- Big thanks to all packagers!!!

  (incl. Juan A. Añel, Markus Dittrich, Takeshi Enomoto, Sébastien Fabbro, Orlando Garcia Feal, Gaurav Khanna, Justin Lecher, Sebastien Maret, Lea Noreskal, Orion Poplawski, Marius Schamschula, Gürkan Sengün, Thierry Thomas, . . .)

- More help and feedback needed. . .

  - upgrades/enhancements to existing packages (Debian/Ubutnu!)
  - new packages (OpenSUSE, Homebrew, Cygwin, Slackware, Solaris, . . .)

	Arch (AUR)	Debian	Fedora	Fink	FreeBSD	Gentoo	Hmug	MacPorts	Ubuntu
GDL version:	0.9	0.9rc3	0.9	0.9	0.9	0.9	0.9	0.9	0.9rc3
features:									
FFTW	−	+	+	+	+	+	+	+	+
GSHHS	−	−	−	−	−	−	−	+	−
GRIB_API	−	−	+	−	−	−	−	+	−
HDF4	−	+	+	+	+	+	+	+	+
HDF5	+	+	+	+	+	+	+	+	+
ImageMagick	+	+	+	+	+	+	+	+	+
libproject	−	−	−	−	−	−	−	+	−
netCDF	+	+	+	+	+	+	−	+	+
GDL→Python	+	−	+	−	+	+	+	+	−
Python→GDL	−	−	+	−	−	+	−	−	−
UDUNITS-2	−	−	+	−	−	+	−	+	−
wxWidgets	+	−	+	−	+	+	−	+	−

- Big thanks to all packagers!!!

  (incl. Juan A. Añel, Markus Dittrich, Takeshi Enomoto, Sébastien Fabbro, Orlando Garcia Feal, Gaurav Khanna, Justin Lecher, Sebastien Maret, Lea Noreskal, Orion Poplawski, Marius Schamschula, Gürkan Sengün, Thierry Thomas, . . .)

- More help and feedback needed...
  - upgrades/enhancements to existing packages (Debian/Ubutnu!)
  - new packages (OpenSUSE, Homebrew, Cygwin, Slackware, Solaris, . . .)

Recently added features (since 0.9rc3):

- Multithreading (multi-core) matrix operations using OpenMP
- GRIB 1/2 file format & GSHHS shoreline database support (both announced for the upcoming release of IDL!)
- New language features from IDL 8.0 (foreach, garbage collection)
- CALL_EXTERNAL (dlopen() interface by Christoph Fuchs)
- numarray → Numpy transition (thanks to Orion Poplawski, et al.)
- cmake build files for GDL (thanks to Maxime Lenoir)
- over 50 new library routines (incl. wavelet transforms)

Key TODO items (help & feedback welcome):

- documentation – currently we rely on IDL docs (on the web)
- enhance (rewrite?) the plotting code (GDL↔plplot)

Thanks for your attention!

http://gnudatalanguage.sf.net/

Recently added features (since 0.9rc3):

- Multithreading (multi-core) matrix operations using OpenMP
- GRIB 1/2 file format & GSHHS shoreline database support (both announced for the upcoming release of IDL!)
- New language features from IDL 8.0 (foreach, garbage collection)
- CALL_EXTERNAL (dlopen() interface by Christoph Fuchs)
- numarray → Numpy transition (thanks to Orion Poplawski, et al.)
- cmake build files for GDL (thanks to Maxime Lenoir)
- over 50 new library routines (incl. wavelet transforms)

Key TODO items (help & feedback welcome):

- documentation – currently we rely on IDL docs (on the web)
- enhance (rewrite?) the plotting code (GDL↔plplot)

Thanks for your attention!

http://gnudatalanguage.sf.net/

Recently added features (since 0.9rc3):

- Multithreading (multi-core) matrix operations using OpenMP
- GRIB 1/2 file format & GSHHS shoreline database support (both announced for the upcoming release of IDL!)
- New language features from IDL 8.0 (foreach, garbage collection)
- CALL_EXTERNAL (dlopen() interface by Christoph Fuchs)
- numarray → Numpy transition (thanks to Orion Poplawski, et al.)
- cmake build files for GDL (thanks to Maxime Lenoir)
- over 50 new library routines (incl. wavelet transforms)

Key TODO items (help & feedback welcome):

- documentation – currently we rely on IDL docs (on the web)
- enhance (rewrite?) the plotting code (GDL↔plplot)

Thanks for your attention!

http://gnudatalanguage.sf.net/