GRASS Grid 3D Modules

Raster	General	Sites	Scripts
r3.in.ascii	g3.region	s.to.rast3	g3.createwind
r3.in.grid3		s.vol.idw	g3.list
r3.in.v5d		s.vol.rst	g3.remove
r3.info			g3.rename
r3.mapcalc			g3.setregion
r3.mask			
r3.mkdspf			
r3.null			
r3.out.ascii			
r3.out.v5d			
r3.showdspf			
r3.timestamp			
r3.to.sites			

Other Commands

help home, database, display, drivers, general, grid3d, imagery, import, misc, models, paint, photo, postscript, raster, scripts, sites, vector

NAME

g3.region − Allows interactive creation and modification of the current 3D window and 3D windows stored in window databases. It can use existing 2D and 3D windows, 2D and 3D raster maps, 2D vector maps, and 3D view files to set from.
(GRASS 3D Program)

SYNOPSIS

g3.region

DESCRIPTION

For a command line version of *g3.region* see *g3.setregion*.

SEE ALSO

g.region, *g3.setregion*

AUTHORS

Roman Waupotitsch, Michael Shapiro, Helena Mitasova, Bill Brown, Lubos Mitas, Jaro Hofierka

Last changed: $Date: 2002/01/25 05:45:33 $

NAME

g3.setregion − Allows command line creation and modification of the current 3D window and 3D windows stored in window databases. It can use existing 2D and 3D windows, 2D and 3D raster maps, 2D vector maps, and 3D view files to set from.
(GRASS 3D Program)

SYNOPSIS

g3.setregion

DESCRIPTION

This module is a command line version of *g3.region*.

SEE ALSO

g3.region

AUTHOR

Markus Neteler

Last changed: $Date: 2002/01/25 05:45:33 $

NAME

r3.in.ascii − Imports files in 3D ASCII format into G3D.
(GRASS 3D Program)

SYNOPSIS

r3.in.ascii [type = *default* | *double* | *float***] [precision =** *default* | *max* | *0 − 52***] [compression =** *default* | *rle* | *none***] [tiledimension =** *XxYxZ***] [nv =** none | *double***] input =** *ascii−file* **output =** *g3d−map*

DESCRIPTION

Parameters:

type
: Data type used in the output file
Options: default, double, float

precision
: Precision used in the output file
Options: default, max, 0−52

compression
: Note that the *none* option only specifies that neither LZW nor RLE is used for compression. It does not turn off the compression all together. G3D does not support non−compressed files.
Options: default, rle, none

tiledimension
: The dimension of the tiles used in the output file. The format is: XxYxZ

nv
: Specifies which value to convert to NULL−value. If the specified value is *none*, no conversion is performed. Default is *none*.

input
: Path and name of ASCII file to be imported

output
: Name of the G3D output map

NOTES

The format for the ASCII file:

```
north: floating point
south: floating point
east: floating point
west: floating point
top: floating point
```

```
bottom: floating point
rows: integer
cols: integer
levels: integer
```

This header is followed by the cell values in *floating point* format organized in rows with constant *col* and *level* coordinate. The rows are organized by constant *level* coordinate. Individual cell values are separated by *space* or *CR*.

NOTE: Currently, after the file has been imported, the stored values are compared with the original data. This feature is used to find bugs in the library at an early stage and will be turned off as soon as confidence has built up.

AUTHORS

Roman Waupotitsch, Michael Shapiro, Helena Mitasova, Bill Brown, Lubos Mitas, Jaro Hofierka

SEE ALSO

r3.out.ascii, s.to.rast3, GRASS ASCII formats

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.info − Displays information about *map*.
(GRASS 3D Program)

SYNOPSIS

r3.info
r3.info [grid3 = *g3d−map***]**

DESCRIPTION

Command line and interactive versions of *r3.info* are supported.

Parameter:

grid3
Name of the G3D map that the user seeks information on

SEE ALSO

r.info

AUTHORS

Roman Waupotitsch, Michael Shapiro, Helena Mitasova, Bill Brown, Lubos Mitas, Jaro Hofierka

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.in.grid3 − Imports files in Bill Brown's *grid3* format into G3D format.
(GRASS 3D Program)

SYNOPSIS

r3.in.grid3 [type = *default* | *double* | *float***] [precision =** *default* | *max* | *0 − 52***] [compression =** *default* | *rle* | *none***] [tiledimension =** *XxYxZ***] [nv =** *yes* | *no***] input =** *grid3−file* **output =** *g3d−map*

DESCRIPTION

Parameters:

type
: The default cell−type is float.
Options: default, double, float

precision
: Options: default, max, 0−52

compression
: Note that the *none* option only specifies that neither LZW nor RLE is used for compression. It does not turn off the compression all together. G3D does not support non−compressed files.
Options: default, rle, none

tiledimension
: Format: XxYxZ

nv
: Specifies whether zeros in the grid3 file are converted to NULL−values. Conversion is performed if the value is *yes*. Default is *no*.

input
: Path and name of grid3 file to be imported

output
: Name of the G3D output map

NOTES

r3.in.grid3 uses *float* as default cell−*type*.

Currently, after the file has been imported, the stored values are compared with the original data. This feature is used to find bugs in the library at an early stage and will be turned off as soon as confidence has built up.

AUTHORS

Roman Waupotitsch, Michael Shapiro, Helena Mitasova, Bill Brown, Lubos Mitas, Jaro Hofierka

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.in.v5d − Imports files in v5d format to G3D format.
(GRASS G3D Program)

SYNOPSIS

r3.in.v5d input=*name* **output=***name* **[nv=***name* **] [type=***name* **] [precision=***name* **] [compression=***name* **] [tiledimension=***name* **]**

DESCRIPTION

Vis5D is a system for interactive visualization of large 5−D gridded data sets such as those produced by numerical weather models. One can make isosurfaces, contour line slices, colored slices, volume renderings, etc of data in a 3−D grid, then rotate and animate the images in real time. There's also a feature for wind trajectory tracing, a way to make text anotations for publications, support for interactive data analysis, etc.
r3.in.v5d imports 3−dimensional files (i.e. the v5d file with 1 variable and 1 time step). Otherwise, only first variable and timestep from 4/5D v5d file will be imported.

Parameters:

input
Path and name of v5d file to be imported

output
Name of the G3D output raster map

nv
String representing NULL value data cell (use 'none' if no such value)
Default: none

type
Data type used in the output file
Options: default, double, float
Default: default

precision
Precision used in the output file
Options: default, max, 0−52
Default: default

compression
The compression method used in the output file
Options: default, rle, none
Default: default

tiledimension
The dimension of the tiles used in the output file
Default: default

SEE ALSO

r3.out.v5d

AUTHOR

Jaro Hofierka, GeoModel s.r.o., Slovakia

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.mapcalc − G3D grid volume data calculator.
(GRASS 3D Program)

SYNOPSIS

r3.mapcalc [**result=***expression*]

DESCRIPTION

r3.mapcalc performs arithmetic on 3D grid volume data. New 3D grids can be created which are arithmetic expressions involving existing 3D grids, floating point constants, and functions.

PROGRAM USE

If used without command line arguments, *r3.mapcalc* will read its input, one line at a time, from standard input (which is the keyboard, unless directed from a file or across a pipe). Otherwise, the expression on the command line is evaluated. *r3.mapcalc* expects its input to have the form:

result=*expression*

where *result* is the name of a 3D grid to contain the result of the calculation and *expression* is any legal arithmetic expression involving existing 3D grid, floating point constants, and functions known to the calculator. Parentheses are allowed in the expression and may be nested to any depth. *result* will be created in the user's current mapset.

The formula entered to *r3.mapcalc* by the user is recorded both in the *result* grid title (which appears in the category file for *result*) and in the history file for *result*.

Some characters have special meaning to the command shell. If the user is entering input to *r.mapcalc* on the command line, expressions should be enclosed within single quotes. See NOTES, below.

OPERATORS AND ORDER OF PRECEDENCE

The following operators are supported:

```
Operator   Meaning                              Type        Precedence
-----------------------------------------------------------------------
%          modulus (remainder upon division)   Arithmetic  4
/          division                             Arithmetic  4
*          multiplication                       Arithmetic  4
+          addition                             Arithmetic  3
```

```
-         subtraction                    Arithmetic  3
==        equal                          Logical     2
!=        not equal                      Logical     2
>         greater than                   Logical     2
>=        greater than or equal          Logical     2
<         less than                      Logical     2
<=        less than or equal             Logical     2
&&        and                            Logical     1
||        or                             Logical     1
```

```
The operators are applied from left to right, with those of higher precedence
applied before those with lower precedence. Division by 0 and modulus by
0 are acceptable and give a 0 result. The logical operators give a 1 result
if the comparison is true, 0 otherwise.
```

3D GRID NAMES

Anything in the expression which is not a number, operator, or function name is taken to be a 3D grid name. Examples:

volume
x3
3d.his

Most GRASS raster map layers and 3D grids meet this naming convention. However, if a 3D grid has a name which conflicts with the above rule, it should be quoted. For example, the expression

x = a−b

would be interpreted as: x equals a minus b, whereas

x = "a−b"

would be interpreted as: x equals 3D grid named *a−b*

Also

x = 3107

would create *x* filled with the number 3107, while

x = "3107"

would copy the 3D grid *3107* to the 3D grid *x*.

Quotes are not required unless the 3D grid names look like numbers or contain operators, OR unless the program is run non−interactively. Examples given here assume the program is run interactively. See NOTES, below.

r3.mapcalc will look for the 3D grids according to the user's current mapset search path. It is possible to override the search path and specify the mapset from which to select the 3D grid. This is done by specifying the 3D grid name in the form:

name@mapset

For example, the following is a legal expression:

result = x@PERMANENT / y@SOILS

The mapset specified does not have to be in the mapset search path. (This method of overriding the mapset search path is common to all GRASS commands, not just *r3.mapcalc*.)

THE NEIGHBORHOOD MODIFIER

3D grids are data base files stored in voxel format, i.e., three−dimensional matrices of float/double values. In *r3.mapcalc*, 3D grids may be followed by a *neighborhood* modifier that specifies a relative offset from the current cell being evaluated. The format is *map[r,c,d]*, where *r* is the row offset, *c* is the column offset and *d* is the depth offset. For example, *map[1,2,3]* refers to the cell one row below, two columns to the right and 3 levels below of the current cell, *map[−3,−2,−1]* refers to the cell three rows above, two columns to the left and one level below of the current cell, and *map[0,1,0]* refers to the cell one column to the right of the current cell. This syntax permits the development of neighborhood−type filters within a single 3D grid or across multiple 3D grids.

FUNCTIONS

The functions currently supported are listed in the table below.

```
function                description
---------------------------------------------------------------------------
abs(x)                  return absolute value of x
atan(x)                 inverse tangent of x (result is in degrees)
cos(x)                  cosine of x (x is in degrees)
col()                   return current column
depth()                 return current depth
eval([x,y,...,]z)       evaluate values of listed expr, pass results to z
exp(x)                  exponential function of x
exp(x,y)                x to the power y
ewres()                 east−west resolution from WIND3D
if                      decision options:
if(x)                   1 if x not zero, 0 otherwise
if(x,a)                 a if x not zero, 0 otherwise
if(x,a,b)               a if x not zero, b otherwise
if(x,a,b,c)             a if x > 0, b if x is zero, c if x < 0
isnull(x)               1 if x not zero, 0 otherwise
log(x)                  natural log of x
log(x,b)                log of x base b
max(x,y[,z...])         largest value of those listed
median(x,y[,z...])      median value of those listed
min(x,y[,z...])         smallest value of those listed
mode(x,y[,z...])        most frequently value of those listed
null()                  return 0
nsres()                 north−south resolution from WIND3D
rand(x,y)               random value between x and y
round(x)                round x
row()                   return current row
sin(x)                  sine of x (x is in degrees)
```

```
sqrt(x)                 square root of x
tan(x)                  tangent of x (x is in degrees)
tbres()                 top-bottom resolution from WIND3D
x()                     return current x value
y()                     return current y value
z()                     return current z value
```

Note, that the row(), col() and depth() indexing starts with 1.

EXAMPLES

To compute the average of two 3D grids *a* and *b*:

```
ave = (a + b)/2
```

To form a weighted average:

```
ave = (5*a + 3*b)/8.0
```

To produce a binary representation of 3D grid *a* so that category 0 remains 0 and all other categories become 1:

```
mask = a/a
```

This could also be accomplished by:

```
mask = if(a)
```

To mask 3D grid *b* by 3D grid *a*:

```
result = if(a,b)
```

REGION/MASK

The user must be aware of the current geographic region and current mask settings when using *r3.mapcalc*. All 3D grids are read into the current geographic region masked by the current mask. If it is desired to modify an existing 3D grid without involving other 3D grids, the geographic region should be set to agree with the cell header for the 3D grid. For example, suppose it is determined that the *volume* 3D grid must have each category value increased by 10 meters. The following expression is legal and will do the job:

```
new_volume = volume + 10
```

Since a category value of 0 is used in GRASS for locations which do not exist in 3D grid, the new 3D grid will contain the category value 10 in the locations that did not exist in the original volume. Therefore, in this example, it is essential that the boundaries of the geographic region be set to agree with the cell header.

However, if there is a current mask, then the resultant 3D grid is masked when it is written; i.e., 0 category values in the mask force zero values in the output.

NOTES

Extra care must be taken if the expression is given on the command line. Some characters have special meaning to the UNIX shell. These include, among others:

* ( ) > & |

It is advisable to put single quotes around the expression; e.g.:

```
result = 'elevation * 2'
```

Without the quotes, the *, which has special meaning to the UNIX shell, would be altered and *r3.mapcalc* would see something other than the *.

If the input comes directly from the keyboard and the *result* 3D grid exists, the user will be asked if it can be overwritten. Otherwise, the *result* 3D grid will automatically be overwritten if it exists.

Quoting *result* is not allowed. However, it is never necessary to quote *result* since it is always taken to be a 3D grid name.

For formulas that the user enters from standard input (rather than from the command line), a line continuation feature now exists. If the user adds \e to the end of an input line, *r3.mapcalc* assumes that the formula being entered by the user continues on to the next input line. There is no limit to the possible number of input lines or to the length of a formula.

If the *r3.mapcalc* formula entered by the user is very long, the map title will contain only some of it, but most (if not all) of the formula will be placed into the history file for the *result* map.

When the user enters input to *r3.mapcalc* non−interactively on the command line, the program will not warn the user not to overwrite existing 3D grids. Users should therefore take care to assign program outputs 3D grid file names that do not yet exist in their current mapsets.

BUGS

Continuation lines must end with a \ and have NO trailing white space (blanks or tabs). If the user does leave white space at the end of continuation lines, the error messages produced by *r3.mapcalc* will be meaningless and the equation will not work as the user intended.

Error messages produced by *r3.mapcalc* are almost useless. In future, *r3.mapcalc* should make some attempt to point the user to the offending section of the equation, e.g.:

```
x = a * b ++ c

ERROR: somewhere in line 1: ...  b ++ c ...
```

Currently, there is no comment mechanism in *r3.mapcalc*. Perhaps adding a capability that would cause the entire line to be ignored when the user inserted a # at the start of a line as if it were not present, would do the trick.

The function should require the user to type "end" or "exit" instead of simply a blank line. This would make separation of multiple scripts separable by white space.

SEE ALSO

r.mapcalc

AUTHOR

Tomas Paudits & Jaro Hofierka, funded by GeoModel s.r.o., Slovakia
tpaudits@mailbox.sk, hofierka@geomodel.sk

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.mask − Creates a new 3D−mask file.
(GRASS 3D Program)

SYNOPSIS

r3.mask
r3.mask [maskvalues=*val[−val][,val[−val],...]***] grid3d =** *g3d−map*

DESCRIPTION

File *map* is used as reference file. Cells in the mask are marked as "mask out" if the corresponding cell in *map* contains a value in the range specified with *maskvalues*.

Before a new 3d−mask can be created the exisitng mask has to be removed with *g.remove*.

Parameters:

maskvalues
: Values specified to be masked out

grid3d
: Name of G3D−map that is used as the mask reference

SEE ALSO

g.remove, *r.mask*

AUTHORS

Roman Waupotitsch, Michael Shapiro, Helena Mitasova, Bill Brown, Lubos Mitas, Jaro Hofierka

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.mkdspf − Creates a display file from an existing grid3 file according to specified threshold levels.
(GRASS 3D Program)

SYNOPSIS

r3.mkdspf [−qf] grid3=*name* **dspf=***name* **[levels=***value[,value,...]***] [min=***value***] [max=***value***] [step=***value***]**
[tnum=*value***]**

DESCRIPTION

Creates a display file from an existing grid3 file according to specified threshold levels. The display file is a display list of polygons that represent isosurfaces of the data volume. If specific *levels* are given, additional optional parameters are ignored. *Min* or *max* may be used alone or together to specify a sub−range of the data. The *step* parameter is given precedence over *tnum*.

Flags:

−q
: Suppress progress report & min/max information

−f
: Use flat shading rather than gradient

Parameters:

grid3
: Name of an existing 3dcell map

dspf
: Name of output display file

levels
: List of thresholds for isosurfaces

min
: Minimum isosurface level

max
: Maximum isosurface level

step
: Positive increment between isosurface levels

tnum
: Number of isosurface threshold levels
Default: 7

Example:

With grid3 data (*phdata*) in the range 3−7, we only want to see isosurface values for the range 4−6. Any of these commands will produce the same results:

```
r3.mkdspf phdata dspf=iso min=4.0 max=6.0 tnum=5
r3.mkdspf phdata dspf=iso levels=4.0,4.5,5.0,5.5,6.0
r3.mkdspf phdata dspf=iso min=4.0 max=6.0 step=0.5
```

NOTE

Currently the grid3 file must be in the user's mapset since the display files being created are specific to particular grid3 files and are contained in directories under them. We should create a mechanism where users may make display files from others' grid3 files without having to copy them to their mapset.

AUTHOR

Bill Brown, bbrown@gis.uiuc.edu

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.null − Modifies the NULL values of *map*.
(GRASS 3D Program)

SYNOPSIS

r3.null [setnull=*val[−val][,val[−val],...]***] [null=***double***] grid3 =** *g3d−map*

Parameters:

setnull
: The cell values specified in the range are to be set to NULL values. A range can be a single value (e.g., 2.5) or multiple values (e.g., 3.21−10.95).

null
: Value that the existing NULL values in *map* are converted to. This applies only to existing NULL values, and not to the NULLs created by the *setnull* argument.

grid3
: Name of the G3D map for which to modify null values

SEE ALSO

r.null

AUTHORS

Roman Waupotitsch, Michael Shapiro, Helena Mitasova, Bill Brown, Lubos Mitas, Jaro Hofierka

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.out.ascii − Outputs G3D maps in ASCII format.
(GRASS 3D Program)

SYNOPSIS

r3.out.ascii [−h] grid3=*name* **[output=***name***] [dp=***value***] [null=***name***]**

DESCRIPTION

Outputs *G3D* maps in *ascii* format. *map* is a valid G3D map in the current mapset. *output* is the name of an ascii file which will be written in the current working directory. If *output* is not specified then **stdout** is used. The *−h* flag may be used to suppress header information. The module is sensitive to region settings (set with g3.region).

Flag:

−h
: Suppress printing of header information

Parameters:

grid3
: G3d raster map to be converted to ascii

output
: Name for ascii output file

dp
: Number of decimal places for floats (options: 0−20)
Default: 8

null
: Char string to represent no data cell
Default: *

NOTES

The default format for the ascii file is equivalent to that required by *r3.in.ascii*. In particular, files output by *r3.out.ascii* with header information may be converted back to G3D maps with *r3.in.ascii*.

The format for the ascii file is:

The header is followed by cell values in *floating point* format. Cell values are output as a series of horizontal slices in row−major order. That is,
One level maps can be imported with r.in.ascii (Raster 2D) after removing the header lines "top", "bottom" and "levels".

SEE ALSO

r3.in.ascii
g3.region

AUTHORS

Roman Waupotitsch, Michael Shapiro, Helena Mitasova, Bill Brown, Lubos Mitas, Jaro Hofierka

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.out.v5d − Exports G3D grids to v5d format (VIS5D).
(GRASS 3D Program)

SYNOPSIS

r3.out.v5d [−m] grid3=*name* **[output=***name***]**

DESCRIPTION

Exports *G3D* grids to *v5d* format. *map* is a valid G3D grid in the current mapset. *output* is the name of a v5d file which will be written in the current working directory. VIS5D is a system for interactive visualization of large 5−D gridded data sets such as those produced by numerical weather models. One can make isosurfaces, contour line slices, colored slices, volume renderings, etc of data in a 3−D grid, then rotate and animate the images in real time. There's also a feature for wind trajectory tracing, a way to make text anotations for publications, support for interactive data analysis, etc.

Flags:

−m
: Use map coordinates instead of xyz coordinates

Parameters:

grid3
: G3d grid map to be converted to v5d

output
: Name for v5d output file

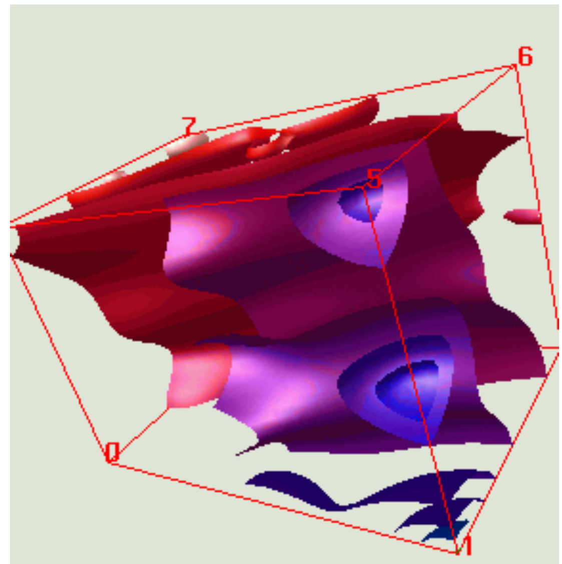
SEE ALSO

r3.in.v5d

AUTHOR

Jaro Hofierka, GeoModel s.r.o., Slovakia

Last changed: $Date: 2002/01/25 05:45:35 $



NAME

r3.showdspf − Visualization program which loads the isosurfaces previously calculated using r3.mkdspf and displays them according to commands given at the prompt.
(GRASS 3D Program)

SYNOPSIS

r3.showdspf **grid3=***name* **dspf=***name* **[color=***name***]**

Parameters:

grid3
: Name of an existing 3dcell map

dspf
: Name of an existing display file

color
: Name of existing color table

DESCRIPTION

Visualization program which loads the isosurfaces previously calculated using r3.mkdspf and displays them according to commands given at the prompt. r3.mkdspf creates a dspf file from the 3D raster and r3.showdspf uses this dspf file to draw isosurfaces and g3d file to draw planes and everything related (boxes, etc).
Upon initialization of the program, two graphics windows are opened, one for the color table and the other for

data display. The display window initially contains a red bounding box. Command options are then printed to the terminal and user is prompted for drawing instructions:

```
THE INTERACTIVE OPTIONS ARE:

?, (l #), L, (t #), (T # #), I, +, -
(x #) (y #) (z #) r (X #) (Y #) (Z #)
(B(x,y,z)#), (E(x,y,z)#), S, R, F, C, c, s, b, g, n, p[#], d, D, w, Q, h

 USAGE AND MEANING:

 ?         lists available thresholds
 l index# [index#...]  add threshold to display list
 L         Draw current display list
 t index#  reset so only this threshold is displayed
 T index# index#   show thresholds between hi & lo
 I         toggle thresholds INSIDE hi/lo or OUTSIDE hi/lo
 +(+++)    display thresholds with consecutively increasing index#
 -(---)    display thresholds with consecutively decreasing index#

 x int#  absolute rotation around x-axis in degrees(int)
 y int#  absolute rotation around y-axis in degrees(int)
 z int#  absolute rotation around z-axis in degrees(int)
 r       rotate_model
 X int#  scale model in x
 Y int#  scale model in y
 Z int#  scale model in z

 B(x,y,z)int#  begin display along (x,y,z) axis at #
 E(x,y,z)int#  end display along (x,y,z)axis #
 S int#        specular highlight control
 R   resets display along axis to show all data
 F grid3name colortablename load new color file

 C   toggles the clear flag
 c   clears the display (no thresholds)
 s   swap buffers
 b   toggles draw a box
 g   toggles grid
 n   toggle surface normal direction

 p   draw all walls
 p#  draw a wall: 1-top, 2-bottom, 3-east, 4-west, 5-north, 6-south

 d   draw (implement the option)
 D   draw a solid defined by T(isosurface + parts of walls)

 w   dump image to a file
 Q   QUIT
 h   help

 enter desired manipulations then press return
 >>
```

Hints:

- To navigate around the data, use the **r** command, then place the mouse pointer in the graphics window and drag with the left mouse to rotate the bounding box. To zoom in and out, drag right or left with the middle mouse. When satisfied with the new viewing position, click with the right mouse.

* To quickly view a series of isosurfaces, enter a series of + or − characters, i.e. `+++++++`
* Scripts using above commands on separate lines may be directed to r3.showdspf as standard input. Use the # sign as the first character on a line to indicate a comment.

EXAMPLE

After generating a "dspf" control file with *r3.mkdspf* start *r3.showdspf*. Display/add the layers using "+". List available thresholds with "?". Use "l" to select isosurfaces (available number can be adjusted with *r3.mkdspf*) and "L" to display:

```
l 1 2 3 4 5
L
```

To select and display a single threshold (here: 2), use:

```
t 2
```

To select and display a range of thresholds (here: 3−5), use:

```
T 3 5
D
```

To draw a box, enter

```
p
```

the p# to plot a selected wall (here top wall):

```
p1
```

Tp draw a cut−off box, define it's position

```
Ex20
p
```

Here Ex20 defines the x coordinate of the end of the box.

In general − p draws a side of a box, E, B, define where that box starts or ends, so to make a fence diagram, the user draws sides of a series of boxes which have their starting (or ending) side shifting by a given interval. (this way the user can draw even more complex fence diagrams which have perpendicular fences, by using Ey or By). It is sufficient to use only E or B depending whether fence are drawn by using the end side or front side of a box).
To draw a fence, a sequence like this would be needed

```
Ex10
p5
Ex15
p5
Ex20
p5
Ex25
p5
```

or the same would be

```
Bx10
p6
Bx15
p6
Bx20
p6
```

The *p* is needed for the fence diagram, solids and boxes.

SEE ALSO

r3.mkdspf

AUTHORS

Bill Brown, brown@gis.uiuc.edu

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.timestamp print/add/remove a timestamp for a grid3d map
(GRASS 3D Program)

SYNOPSIS

r3.timestamp
r3.timestamp help
r3.timestamp grid3d=*name* [**date=***timestamp*],***timestamp***]

DESCRIPTION

This command has 2 modes of operation. If no date argument is supplied, then the current timestamp for the grid3d map is printed. If a date argument is specified, then the timestamp for the grid3d map is set to the specified date(s). See EXAMPLES below.

EXAMPLES

r3.timestamp grid3d=soils
Prints the timestamp for the "soils" grid3d map. If there is no timestamp for soils, nothing is printed. If there is a timestamp, one or two lines are printed, depending on if the timestamp for the map consists of a single date or two dates (ie start and end dates).

r3.timestamp grid3d=soils date='15 sep 1987'
Sets the timestamp for "soils" to the single date
"15 sep 1987"

r3.timestamp grid3d=soils date='15 sep 1987,20 feb 1988'
Sets the timestamp for "soils" to have the start date
"15 sep 1987" and the end date "20 feb 1988"

r3.timestamp grid3d=soils date=none
Removes the timestamp for the "soils" grid3d map

COMMAND LINE OPTIONS

Parameters *grid3d*: grid3d map name ***date***: date/time stamp or date1,date2 range

TIMESTAMP FORMAT

The timestamp values must use the format as described in the GRASS datetime library. The source tree for this library should have a description of the format. For convience, the formats as of Feb, 1996 are reproduced here:

There are two types of datetime values: absolute and relative. Absolute values specify exact dates and/or times. Relative values specify a span of time. Some examples will help clarify:

Absolute

The general format for absolute values is

```
day month year [bc] hour:minute:seconds timezone
```

day is 1−31
month is jan,feb,...,dec
year is 4 digit year
[bc] if present, indicates dates is BC
hour is 0−23 (24 hour clock)
mintue is 0−59
second is 0−59.9999 (fractions of second allowed)
timezone is +hhmm or −hhmm (eg, −0600)

parts can be missing

1994 [bc]
Jan 1994 [bc]
15 jan 1000 [bc]
15 jan 1994 [bc] 10 [+0000]
15 jan 1994 [bc] 10:00 [+0100]
15 jan 1994 [bc] 10:00:23.34 [−0500]

Relative There are two types of relative datetime values, year− month and day−second. The formats are:

```
[-] # years # months
[-] # days # hours # minutes # seconds
```

The words years, months, days, hours, minutes, seconds are literal words, and the # are the numeric values. Examples:

```
2 years
5 months
2 years 5 months
100 days
15 hours 25 minutes 35.34 seconds
100 days 25 minutes
1000 hours 35.34 seconds
```

The following are *illegal* because it mixes year−month and day−second (because the number of days in a month or in a year vary):

```
3 months 15 days
3 years 10 days
```

BUGS

Spaces in the timestamp value are required.

AUTHOR

Michael Pelizzari
Lockheed Martin Space Systems
based on r.timestamp by Michael Shapiro,
U.S.Army Construction Engineering Research Laboratory

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

r3.to.sites − Converts 3dcell values in a GRASS G3D volume into a GRASS site_lists file.
(GRASS Raster Program)

SYNOPSIS

r3.to.sites
r3.to.sites help
r3.to.sites grid3=*name* **sites**=*name*

DESCRIPTION

The *r3.to.sites* program extracts data from a GRASS 3dcell volume and stores output in a new GRASS *site_lists* file. The resulting sites map layer can be used with such programs as *d.sites*.

The user can run the program non−interactively by specifying the names of an existing raster input 3dcell and a new site list file to be output on the command line. The program will be run interactively if the user types *r3.to.sites* without arguments on the command line. In this case, the user will be prompted to enter parameter values through the standard user interface described in the manual entry for *parser*.

OPTIONS:

Parameters:

grid3
: Name of an existing 3dcell volume from which site data are to be extracted

sites
: Name of new sites file

SEE ALSO

[r3.out.ascii](), [r3.out.v5d](), [s.to.rast3](), [parser]()

AUTHOR

Jaro Hofierka

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

s.vol.idw − Interpolates point data to a 3D grid.
(GRASS 3D Program)

SYNOPSIS

s.vol.idw
s.vol.idw help
s.vol.idw input=*name* **output=***name* **[npoints=***count***] [field=***value***]**

DESCRIPTION

s.vol.idw fills a GRID3D raster volume matrix with interpolated values generated from a set of irregularly spaced data points using numerical approximation (weighted averaging) techniques. The interpolated value of a tile is determined by values of nearby data points and the distance of the cell from those input points. In comparison with other methods, numerical approximation allows representation of more complex volumes (particularly those with anomalous features), restricts the spatial influence of any errors, and generates the interpolated volume from the data points.

Parameters:

input
: Name of input 3D sites file

output
: Name of output 3D − G3D raster file

npoints
: Number of interpolation points
Default: 12

field
: Number of z−field attribute to use for calculation
default: 1

NOTES

If two or more sites fall into one voxel, the last site value will determine the 3dcell value (no warning yet).

SEE ALSO

s.vol.rst, *s.to.rast3*

AUTHOR

Jaro Hofierkahofierka@geomodel.sk

Last changed: $Date: 2002/01/25 05:45:35 $

NAME

s.vol.rst − Interpolates point data to a G3D grid volume using regularized spline with tension (RST) algorithm *(GRASS 3D Program)*

SYNOPSIS

s.vol.rst input=*name* **[cellinp=***name***] [field=***value***] [tension=***value***] [smooth=***value***] [devi=***name***] [maskmap=** *name***] [segmax=***value***] [dmin=***value***] [npmin=***value***] [wmult=***value***] [zmult=***value***] [cellout=***name***] [elev=***name***] [gradient=***name***] [aspect1=***name***] [aspect2=***name***] [ncurv=***name***] [gcurv=***name***] [mcurv=***name***]**

DESCRIPTION

s.vol.rst interpolates the values to 3−dimensional grid from point data (climatic stations, drill holes etc.) given in a 3D sites file named *input*. Output g3d file is *elev*. The 3−dimensional grid is given by the current 3D region. If the options *cellinp* and *cellout* are specified then the output raster file *cellout* contains crossection of interpolated volume with surface defined by input cell file . As an option, simultaneously with interpolation, geometric parameters magnitude of gradient, both aspects, change of gradient, Gauss−Kronecker curvature, or mean curvature are computed and saved as g3d file as specified by the options *gradient, aspect1, aspect2, ncurv, gcurv, mcurv* respectively.

At first, data points are checked for identical points and points that are closer to each other than given *dmin* are removed. Parameters *wmult* and *zmult* allow user to re−scale the w−values and z−values for sites (useful e.g. for transformation of elevations given in feet to meters, so that the proper values of gradient and curvatures can be computed).

Regularized spline with tension is used for the interpolation. The *tension* parameter tunes the character of the resulting volume from thin plate to membrane. Higher values of tension parameter reduce the overshoots that can appear in volumes with rapid change of gradient. For noisy data, it is possible to define a smoothing parameter, *smooth*. With the smoothing parameter set to zero (*smooth=0*) the resulting volume passes exactly through the data points. When smoothing is used, it is possible to output site file *devi* containing deviations of the resulting volume from the given data.

User can define a 2D raster file named *maskmap*, which will be used as a mask. The interpolation is skipped for 3−dimensional cells whose 2−dimensional projection has zero value in mask. Zero values will be assigned to these cells in all output g3d files.

If the number of given points is greater than 700, segmented processing is used. The region is split into 3−dimensional "box" segments, each having less than *segmax* points and interpolation is performed on each segment of the region. To ensure the smooth connection of segments the interpolation function for each segment is computed using the points in given segment and the points in its neighborhood. The minimum number of points taken for interpolation is controlled by *npmin* , the value of which must be larger than *segmax* and less than 700. This limit of 700 was selected to ensure the numerical stability and efficiency of the algorithm.

s.vol.rst uses regularized spline with tension for interpolation from point data (as described in Mitasova and Mitas, 1993). The implementation has an improved segmentation procedure based on Oct−trees which enhances the efficiency for large data sets.

Geometric parameters − magnitude of gradient (*gradient*), horizontal (*aspect1*) and vertical (*aspect2)* aspects, change of gradient (*ncurv*), Gauss−Kronecker (*gcurv*) and mean curvatures (*mcurv*) are computed directly from the interpolation function so that the important relationships between these parameters are preserved. More information on these parameters can be found in Mitasova et al., 1995 or Thorpe, 1979.

The program gives warning when significant overshoots appear and higher tension should be used. However, with tension too high the resulting volume changes its behavior to membrane(rubber sheet stretched over the data points resulting in a peak in each given point and everywhere else the volume goes rapidly to trend). With smoothing parameter greater than zero the volume will not pass through the data points and the higher the parameter the closer the volume will be to the trend. For theory on smoothing with splines see Talmi and Gilat, 1977 or Wahba, 1990.

If a visible connection of segments appears, the program should be rerun with higher *npmin* to get more points from the neighborhood of given segment.

If the number of points in a site file is less then 400, *segmax* should be set to 400 so that segmentation is not performed when it is not necessary.

The program gives warning when user wants to interpolate outside the "box" given by minimum and maximum coordinates in site file, zoom into the area where the points are is suggested in this case.

For large data sets (thousands of data points) it is suggested to zoom into a smaller representative area and test whether the parameters chosen (e.g. defaults) are appropriate.

The user must run *g3.region* before the program to set the region for interpolation.

Parameters:

input
: Name of the site file (format see NOTES below)

field
: decimal attribute to use for value w (1=first) options (1−100), default is 1.

cellinp
: Name of the surface cell file to use for crossection

tension
: Tension
: Default: 40

smooth
: Smoothing parameter
: Default: 0.1

devi
: Output deviations to a site file

maskmap

Name of the raster file used as mask

segmax
Max number of points in segment (=700)
Default: 50

dmin
Min distance between points (extra points ignored)
Default: Default value is set to 0.5 cell size.

npmin
Min number of points for interpolation
Default: 200

wmult
Conversion factor for w−values
Default: 1.0

zmult
Conversion factor for z−values
Default: 1.0

cellout
Name of the crossection cell file

elev
Elevation g3d−file

gradient
Gradient g3d−file

aspect1
Aspect1 g3d−file

aspect2
Aspect2 g3d−file

ncurv
Change of gradient g3d−file

gcurv
Gauss−Kronecker curvature g3d−file

mcurv
Mean curvature g3d−file

NOTES

The sites volume format is as follows:

```
x|y|z|#n %w1 %w2 %w3
```

with x,y,z (spatial coordinates), n (optional integer number) and w (data values).

SEE ALSO

g3.region, s.in.ascii, s.vol.idw, r3.mask, s.surf.rst

AUTHOR

Original version of program (in FORTRAN) and GRASS enhancements:
Lubos Mitas, NCSA, University of Illinois at Urbana−Champaign, Illinois, USA,lubos_mitas@ncsu.edu

Helena Mitasova, Department of Geography, University of Illinois at Urbana−Champaign, Champaign, Illinois, USA, hmitaso@unity.ncsu.edu

Modified program (translated to C, adapted for GRASS, new segmentation procedure):
Irina Kosinovsky, US Army CERL, Champaign, Illinois, USA
Dave Gerdes, US Army CERL, Champaign, Illinois, USA

Modifications for g3d library, geometric parameters, deviations:
Jaro Hofierka, GeoModel s.r.o., Bratislava, Slovakia, hofierka@geomodel.sk, http://www.geomodel.sk

REFERENCES

Hofierka J., Parajka J., Mitasova H., Mitas L., 2002, Multivariate Interpolation of Precipitation Using Regularized Spline with Tension. Transactions in GIS 6, pp. 135−150.

Mitas, L., Mitasova, H., 1999, Spatial Interpolation. In: P.Longley, M.F. Goodchild, D.J. Maguire, D.W.Rhind (Eds.), Geographical Information Systems: Principles, Techniques, Management and Applications, Wiley, pp.481−492

Mitas L., Brown W. M., Mitasova H., 1997, Role of dynamic cartography in simulations of landscape processes based on multi−variate fields. Computers and Geosciences, Vol. 23, No. 4, pp. 437−446 (includes CDROM and WWW: www.elsevier.nl/locate/cgvis)

Mitasova H., Mitas L., Brown W.M., D.P. Gerdes, I. Kosinovsky, Baker, T.1995, Modeling spatially and temporally distributed phenomena: New methods and tools for GRASS GIS. International Journal of GIS, 9 (4), special issue on Integrating GIS and Environmental modeling, 433−446.

Mitasova, H., Mitas, L., Brown, B., Kosinovsky, I., Baker, T., Gerdes, D. (1994): Multidimensional interpolation and visualization in GRASS GIS

Mitasova H. and Mitas L. 1993: Interpolation by Regularized Spline with Tension: I. Theory and Implementation, *Mathematical Geology* 25, 641−655.

Mitasova H. and Hofierka J. 1993: Interpolation by Regularized Spline with Tension: II. Application to Terrain Modeling and Surface Geometry Analysis, *Mathematical Geology* 25, 657−667.

Mitasova, H., 1992 : New capabilities for interpolation and topographic analysis in GRASS, GRASSclippings 6, No.2 (summer), p.13.

Wahba, G., 1990 : Spline Models for Observational Data, CNMS−NSF Regional Conference series in applied mathematics, 59, SIAM, Philadelphia, Pennsylvania.

Mitas, L., Mitasova H., 1988 : General variational approach to the interpolation problem, Computers and Mathematics with Applications 16, p. 983

Talmi, A. and Gilat, G., 1977 : Method for Smooth Approximation of Data, Journal of Computational Physics, 23, p.93−123.

Thorpe, J. A. (1979): Elementary Topics in Differential Geometry. Springer−Verlag, New York, pp. 6−94.

Last changed: $Date: 2003/08/20 08:10:13 $