

7. Post-processing

- [Introduction](#)
- [pcouple](#)
- [pconvert](#)
- [pgrads](#)
- [psample](#)
- [pstat](#)

[Edit](#)

7. Post-processing

Introduction

The output file of mcarats codes is in a unique format. Several tools are provided to process output data files from mcarats codes. The user can easily make a new tool using subroutines in `./src/libmcarats/`. If you made a useful tool, you can contribute to the project by providing it.

- `pcouple` to do a simple calculation with a couple of mcarats output data files
- `pconvert` to convert a mcarats output data file with a simple equation
- `pgrads` to visualize a mcarats output data set using GrADS
- `psample` to (re)sample data from mcarats output file
- `pstat` to calculate spatial statistics of mcarats output data

pcouple

This is used to do a simple calculation with a couple of mcarats output data files.

```
usage: pcouple method thresh infile1 infile2 > outfile
```

- method:

```
0 (sum)           : in1 + in2
1 (difference)    : in1 - in2
2 (product)       : in1 * in2
3 (ratio)         : in1 / in2
4 (relative diff. 1) : (in1 - in2) / in2
5 (relative diff. 2) : 2*(in1 - in2) / (in1 + in2)
6 (square diff.)  : in1*in1 - in2*in2
7 (root square diff.): sqrt(in1*in1 - in2*in2)
8 (diff. square)  : (in1 - in2)*(in1 - in2)
9 (root diff.)    : sqrt(in1) - sqrt(in2)
```

- `thresh`: threshold of the denominator when `method=3-5`. If the denominator is less than the threshold, then the output value is reset at a very small value (`-1.0E+30`).
- `infile1` : Input data file 1.
- `infile2` : Input data file 2.
- `outfile` : Output data file.

The output file is in the same format as the input.

pconvert

This is used to convert a mcarats output data file with a simple equation.

```
usage: pconvert nx ny method A B < infile > outfile
```

- `nx` : # of x-pixels for output.
 - `ny` : # of y-pixels for output.
- If `nx*ny=0`, then these are reset as defined in the input file. Otherwise the averaging is operated.

- `method=0` : $v = u$ (does nothing)

```
1 : v = A + B * u      (linear)
2 : v = A * u^B        (power-law)
3 : v = exp(A + B*u)   (exponential)
4 : v = ln(A + B*u)    (logarithmic)
5 : v = A + (1 + B*G)*u (random noises are superimposed)
   where G is a normalized Gaussian noise (average: 0; variance:1).
```

The output file is in the same format as the input, but if nx or ny is smaller than the input ("nx or ny; nxr or nyr; nxc or nyc; npot or nqpot" defined in infile itself), then the amount of output data is reduced.

pgrads

To quickly visualize output data from mcarats codes, a tool, pgrads, is provided. This converts mcarats output data file to GrADS format.

```
usage: pgrads mach q zset infile (outfile) (< zlevels)
```

■ mach : machine parameter as follows:

```
=0 : little endian, record length for a real*4 datum = 4 (x86+g77)
=1 : little endian, record len = 1 (Alpha/x86+ifort)
=2 : big    endian, record len = 4 (PPC+g77)
=3 : big    endian, record len = 1
```

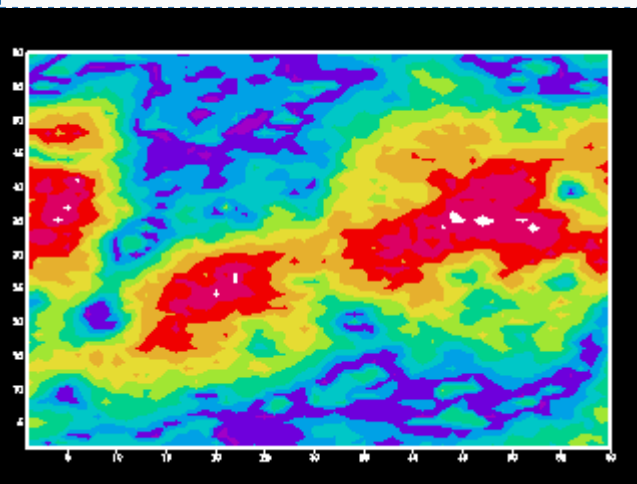
For example

```
little endian: x86, Alpha,...
big    endian: PowerPC, Sun,...
record length=4: g77
record length=1: ifort
```

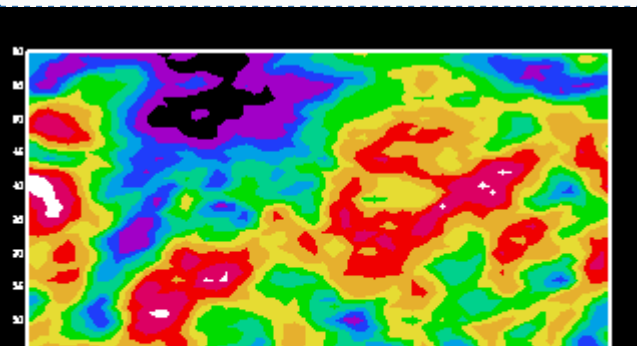
- q : determines which data sets are to be output (0:flux&HR, 1:flux, 2:HR, 3:radiance, 4:camera, 5:local flux)
- zset : method to set z-levels (0: default; 1: read from stdin)
- infile : input file (MCARaTS output)
- outfile : output file (GrADS format)
- zlevels : z-levels when zset=1

If GrADS is installed in your system, you can try the followings (in the "examples/case3" directory):

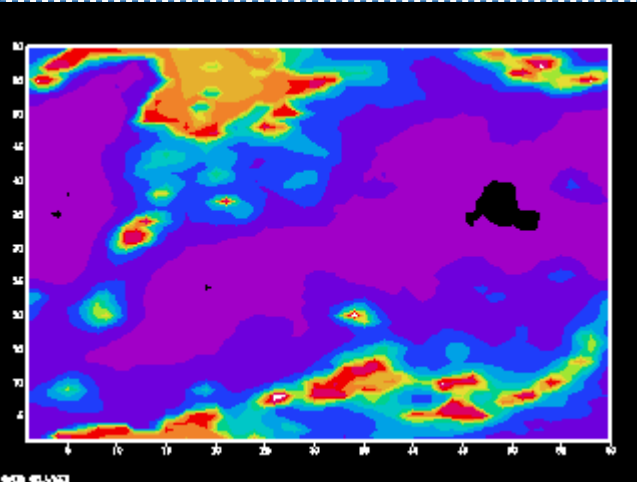
```
% ../../bin/pgrads 1 3 0 out_rs0067-f3d
% grads -l
...
ga-> open out_rs0067-f3d.gd.ctl
...
ga-> set mproj off
ga-> set gxout shaded
ga-> d rad
```



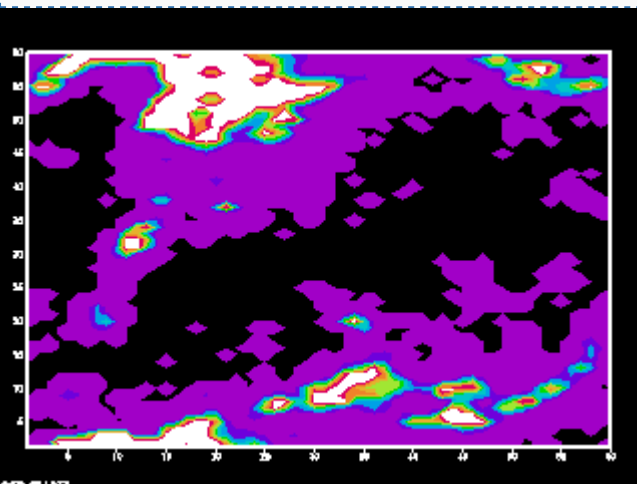
```
ga-> printim rs0067-f3d-t1.png x400 y300
ga-> set t 2
Time values set: 2001:1:0:0 2001:1:0:0
ga-> d rad
```



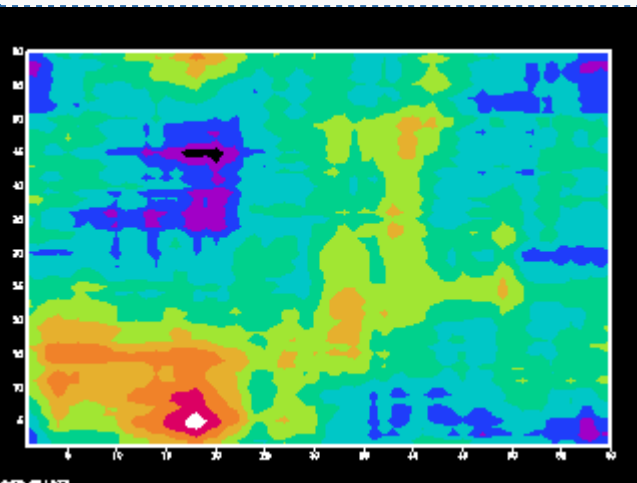
```
ga-> printim rs0067-f3d-t2.png x400 y300
ga-> quit
% ../bin/pgrads 1 0 0 out_hr0213-f3d
% grads -l
...
ga-> open out_rs0213-f3d.gd.ctl
...
ga-> set mproj off
ga-> set gxout shaded
ga-> d fd
```



```
ga-> printim hr0213-f3d-t1-fd.png x400 y300
ga-> d f0
```



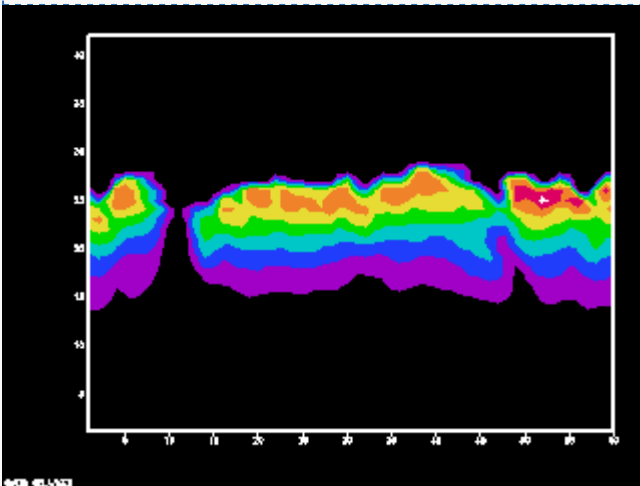
```
ga-> printim hr0213-f3d-t1-f0.png x400 y300
ga-> set z 2
LEV set to 2 2
ga-> d fu
```



```

ga-> printim hr0213-f3d-t1-fu.png x400 y300
ga-> set z 1 42
LEV set to 1 42
ga-> set y 30
LAT set to 30 30
ga-> c
ga-> d hr

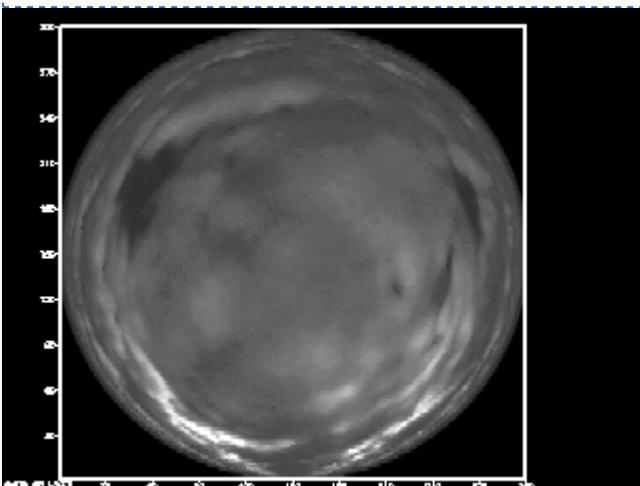
```



```

ga-> printim hr0213-f3d-t1-hr.png x400 y300
ga-> quit
% ../bin/pgrads 1 4 0 out_ci0067-f3d
% grads -l
...
ga-> open out_ci0055-f3d.gd.cti
ga-> set mproj off
ga-> set parea 1 9 0.2 8.2
ga-> set gxout shaded
ga-> set display grey
ga-> set grid off
grid is off
ga-> set t 1
Time values set: 2001:1:0:0 2001:1:0:0
ga-> set clevs 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4
Number of clevs = 15
ga-> d cam

```



```

ga-> printim ci0055-f3d-t1.png x400 y400
ga-> quit

```

psample

This is used to resample data from a mcarats output data file. This is useful to extract partial data from the file.

```
usage: psample ixi ixn xxi iyn ny < infile > outfile
```

- ixi, iyn : initial indexes for X and Y

ixs, iys : step of X and Y indexes

-
- nx, ny : # of X and Y pixels for output

An example is here: at [examples/case3/](#)

```
% ../../bin/psample 2 2 30 2 2 30 < out_rs0067-f3d > sample_rs0067-f3d
```

In this case, original data for 60x60 pixels are available, and output result from the above is for 30x30 pixels.

pstat

This is used to calculate basic statistics of a text data sequence.

```
usage: pstat N func ithresh (threshold1 threshold2) < infile > outfile
```

- N : # of data
- func : 0, raw data; 1, ln(dat); 2, exp(dat); The conversion is performed before the statistics computation.
- ithresh

```
=0 : no threshold,  
=1 : dat >= thresh1,  
=2 : dat <= thresh1,  
=3 : thresh1 <= dat <= thresh2,  
=4 : dat <= thresh1 or dat >= thresh2
```

Output data file reports fraction of data filtered by the threshold(s), minimum, maximum, average, standard deviation, root-mean-square (rms), and skewness. For example,

```
% bin/pstat 3600 0 0 < examples/case3/out_rs0067-f3d  
#frac, min, max, average, stdev, rms, skewness  
1.0000 3.3890E-01 6.4152E-01 4.7632E-01 7.4342E-02 4.8209E-01 2.0645E-01  
1.0000 2.0148E+00 2.0329E+00 2.0212E+00 2.4843E-03 2.0212E+00 8.3968E-01  
1.0000 1.3267E-01 6.6358E-01 3.6322E-01 1.1587E-01 3.8126E-01 -1.3765E-01  
1.0000 3.0360E+00 3.1539E+00 3.0686E+00 1.9938E-02 3.0687E+00 1.1165E+00
```

The output are printed to stdout. Statistics for four data sequences, each of which contains 3600 data, are reported in this example.

[Edit](#)

[Return](#)