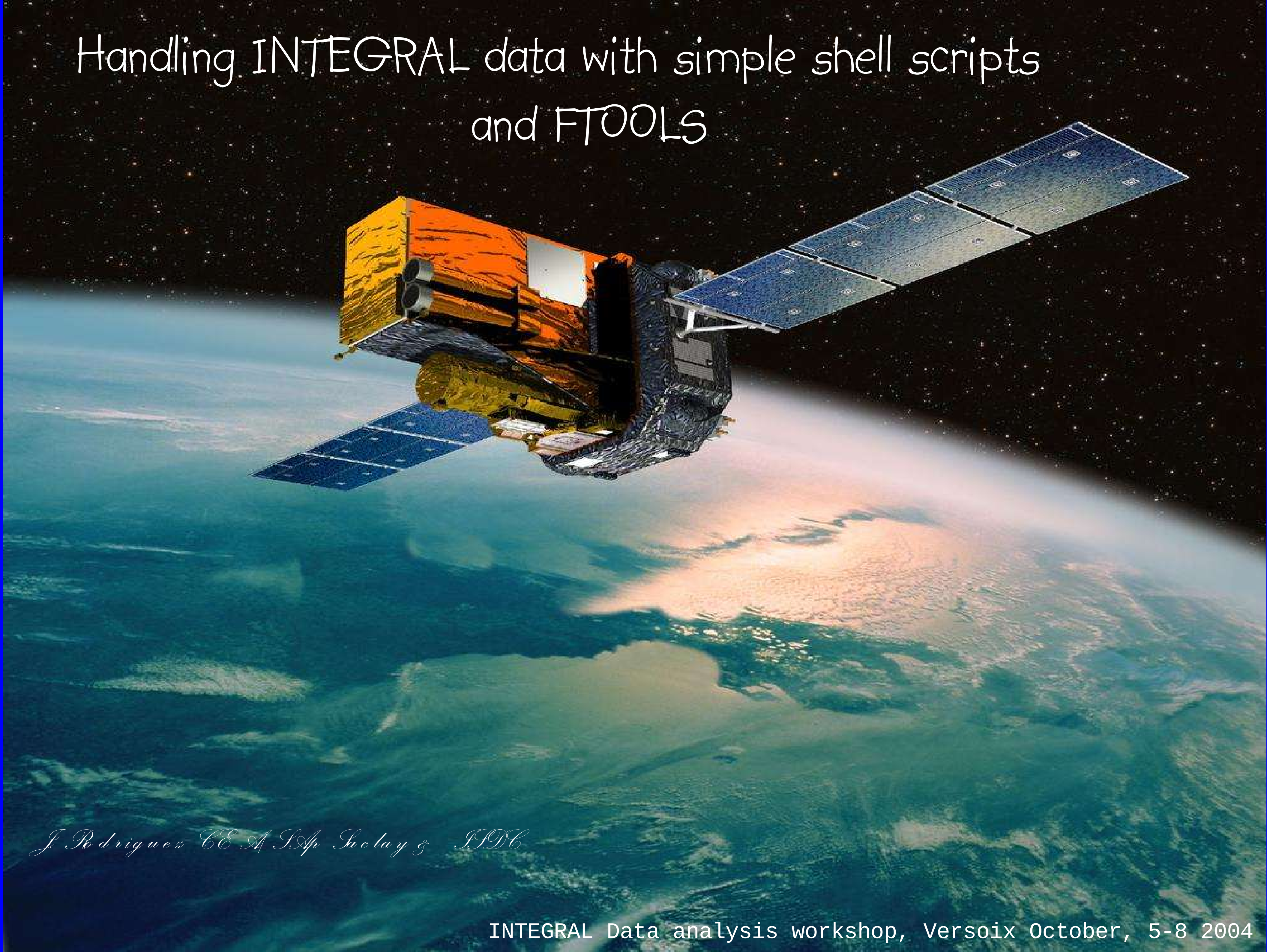
Handling INTEGRAL data with simple shell scripts and FTOOLS



J. Rodriguez CEA SAp Saclay & ISDC

INTEGRAL Data analysis workshop, Versoix October, 5-8 2004

INTRO: data structure

Results from run with OSA

=> $REP_BASE_PROD = my rep. where I launched OSA

=> main data res = $REP_BASE_PROD/obs/myobs

=> indiv. results = $REP_BASE_PROD/obs/myobs/scw

=> indiv. results/instruments = $REP_BASE_PROD/obs/myobs/scw/instru/res

=> mosaics =$REP_BASE_PROD/obs/myobs/instru/res

Note: scw name will have the structure

rev#0scw#0i0(.00j)

where

rev# is the revolution number (4 digits)

scw# is the science window number (4 digits)

i indicates number of scw performed without slew between them (usually 1)

00j “revision” version 0

Getting started : csh loops

Processing large amount of data (lot of scw) => looping

Foreach:

```
~rodrigue> foreach myvar
   (varname)

foreach?> do instruction 1

foreach?> do instruction 2

foreach?> end
```

Myvar =dir, file

=> perform instruction 1 and 2 over each file/dir

While :

```
~rodrigue> while (condition)

while?> do instruction 1

while?> do instruction 2

while?> end
```

Perform instruction 1 and 2 while condition is fulfilled

may need an increement (i->i+1)

Example

Create region files of detected sources with cat2ds9

All dataset:

```
obs/myobs/scw> foreach dir (rev#*)
foreach?> cd $dir/ibis/res
foreach?> cat2ds9\
? catDOL=isgri_sky_res.fits+2\
? fileName=isgchan1.reg
foreach?> cd ../../../
foreach?> end
```

Subset of data:

```
obs/myobs/scw> set i = 0
obs/myobs/scw>while  ($i<9)
while?> @ i ++
while?> cd rev#000{$i}00.001/ibis/res
while?> cat2ds9\
? catDOL=isgri_sky_res.fits+2\
? fileName=isgchan1.reg
while?> cd ../../../
while?> end
```

Mathematical operations in csh

Usefull to perform mathematical expression on a variable

Example : take the sqrt of the variance value at position x,y

$var = var (x,y)

```
Set err = ` echo $var | awk '{print sqrt($1)}'`
```

Example : sum the cr of two subsequent scw

```
Set cr = 0

Foreach scw (scw1 scw2)

I go in scw/ibis/res and read the count rate @ position x,y -> $crtemp

Set cr = ` echo $cr $crtemp | awk '{print ($1 + $2)}'`

end
```

FTOOLS: most important for us

Fdump: dumps the content of fits file

cmmpha: converts phaII to a phaI pha file (JEM-X)

arf2arf1: converts a type 2 ARF file to a type 1 ARF (JEM-X)

fmodhead: modify the header of a fits file (see also faddparkey -> add keyword(s) to the header)

fcreate: create fits file from ascii “ templates ”

fimgdmp: dumps the content of an image

fextract: extract content of extension number j of a fits file and write it to another fits file

Example

Extract my source spectrum and move it.

My source spectrum = extension $j in each indiviual spectrum

```
Obs/myobs/scw> foreach dir (rev#*)

foreach?> set name = `echo $dir | cut -b '6-8'`

foreach?> cd $dir/ibis/res

foreach?> fextract isgri_spectrum.fits+{$j} spectrum_mysource.pha

foreach?> mv spectrum_mysource.pha\ $REP_BASE_PROD/obs/myobs/spec_mysource_
   {$name}.pha

foreach?> cd ../../../

foreach?> end
```

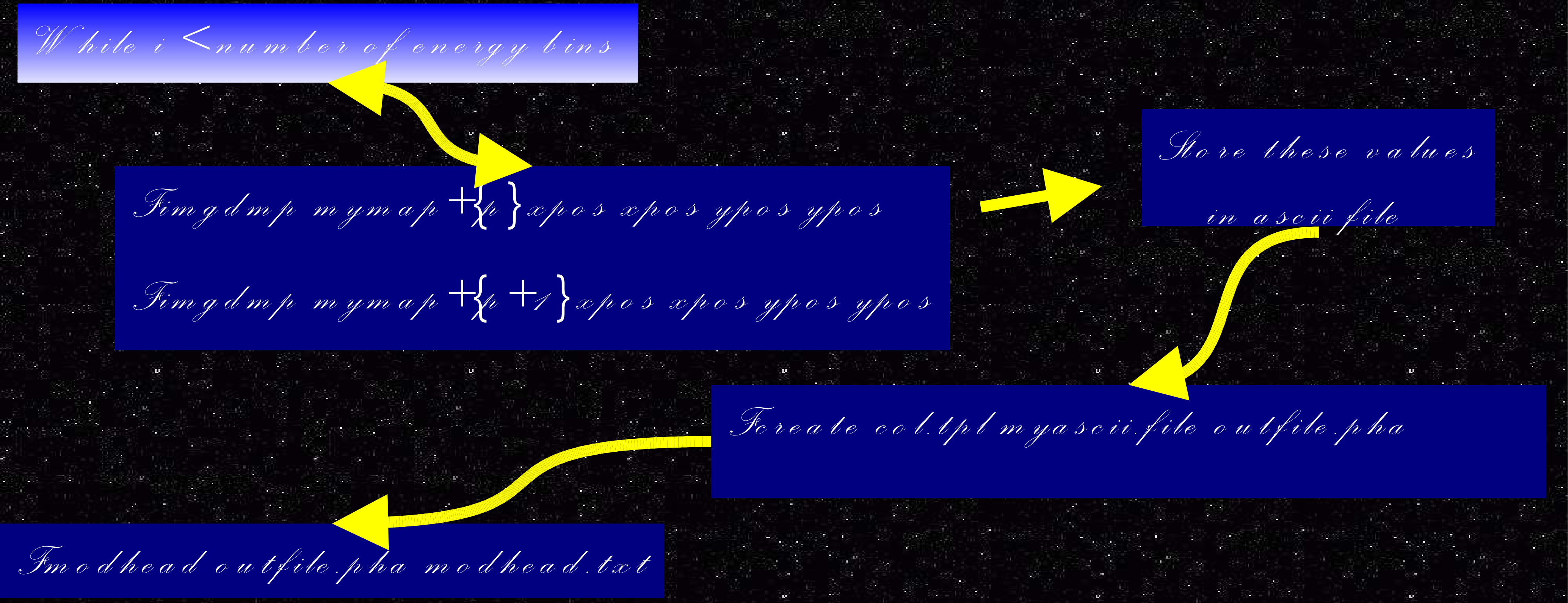
Reading flux and error in ISGRI images

Goal: build a spectrum from an ISGRImosaic performed in x energy ranges

=> x call to a dumping of fits image (intensity and variance map)

=> storage of values in a temp file

=> creation of a fits (pha) file: header, keywords etc....



While i <number of energy bins

```
Fimgdmp mymap+{p }xpos xpos ypos ypos
Fimgdmp mymap+{p +1}xpos xpos ypos ypos
```

Store these values in ascii file

```
Fcreate col.tpl myascii.file outfile.pha
```

```
Fmodhead outfile.pha modhead.txt
```

Extract JEM-X spectra and attach responses

```
foreach dir (0*)

foreach?> setenv spec $dir/jmx2/res/jmx2_srcl_spe.fits

foreach?> setenv arf $dir/jmx2/res/jmx2_srcl_arf.fits

foreach?> setenv extension `fdump {$spec}+1 mode=h prhead=no columns=ROWID\

?showcol=no | grep 'IGRJ1914' | cut -b '6-7'`

foreach?> cmppha $spec outfile={$spec}_pha1 cmpmode=expand clobber=yes

foreach?> fextract {$spec}+$ext ./{$dir}_jmx2.pha clobber=yes

foreach?> cp {$arf} ./{$dir}_jmx2.arf

foreach?> echo 'RESPFILE= '\''./jmx2_rmf_grp_0005.fits'\' '/ redistribution\

? matrix' >>modhead_{$dir}.txt

foreach?> echo 'ANCRFILE= '\'{$dir}'_jmx2.arf'\{{$extension}\} \' '/ ancillary\

? response' >>modhead_{$dir}.txt

foreach?> fmodhead ./{$dir}_jmx2.pha modhead_{$dir}.txt

foreach?> end
```