Subject: Re: howto make a fits cube in IDL?
Posted by wlandsman on Wed, 07 Nov 2007 19:54:06 GMT
View Forum Message <> Reply to Message

On Nov 7, 10:45 am, Tigran Khanzadyan <khanzad...@gmail.com> wrote:
> Dear All,
>
> I wish to create a FITS-CUBE from a model outcome in order to directly
> compare with the radio observations. So the task is to actually write
> on disk a Fits file which would have x,y and a spectra on each x,y
> position. A spectra itself has two dimensions - temperature and
> velocity.
>
> I was wondering if someone could point me to the right direction. I
> know that there is a "writefits" in NASA IDL ASTRO LIBRARY, but I cant
> quite find a task or routine to write FITS-CUBE.

This looks like a case for using the "Binary table" FITS format.
The routines MWRFITS and MRDFITS give a nice mapping between IDL
structures and FITS binary tables. So say you have 200 X,Y points and
that at each X,Y you have a temperature and velocity spectrum with 60
points.

IDL> str = {x:0, y:0, temperature:fltarr(60), velocity:fltarr(60) }
IDL> str = replicate(str, 200)

(If the number of points are not all the same in each spectra then you
could use pointers rather than fixed arrays.)

Then you fill the structure with the actual values, and finally write
the structure to a FITS file

IDL> mwrfits,str,'myfile.fits'

--Wayne

Subject: Re: howto make a fits cube in IDL?
Posted by David Fanning on Wed, 07 Nov 2007 20:26:06 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> This looks like a case for using the "Binary table" FITS format.
> The routines MWRFITS and MRDFITS give a nice mapping between IDL
> structures and FITS binary tables. So say you have 200 X,Y points and
> that at each X,Y you have a temperature and velocity spectrum with 60
> points.

>
> IDL> str = {x:0, y:0, temperature:fltarr(60), velocity:fltarr(60) }
> IDL> str = replicate(str, 200)
>
> (If the number of points are not all the same in each spectra then you
> could use pointers rather than fixed arrays.)
>
> Then you fill the structure with the actual values, and finally write
> the structure to a FITS file
>
> IDL> mwrfits,str,'myfile.fits'

Get out!!

I had a similar question, but I was too embarrassed to ask here.
But this is WAY too easy. I was hoping to bill about 40 hours
on this part of the code. :-(

I better go read up on FITS files.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: howto make a fits cube in IDL?
Posted by Tigran Khanzadyan on Thu, 08 Nov 2007 06:24:33 GMT
View Forum Message <> Reply to Message

On Nov 7, 7:54 pm, wlandsman <wlands...@gmail.com> wrote:
> On Nov 7, 10:45 am, Tigran Khanzadyan <khanzad...@gmail.com> wrote:
>
>> Dear All,
>
>> I wish to create a FITS-CUBE from a model outcome in order to directly
>> compare with the radio observations. So the task is to actually write
>> on disk a Fits file which would have x,y and a spectra on each x,y
>> position. A spectra itself has two dimensions - temperature and
>> velocity.
>
>> I was wondering if someone could point me to the right direction. I

>> know that there is a "writefits" in NASA IDL ASTRO LIBRARY, but I cant
>> quite find a task or routine to write FITS-CUBE.
>
> This looks like a case for using the "Binary table" FITS format.
> The routines MWRFITS and MRDFITS give a nice mapping between IDL
> structures and FITS binary tables. So say you have 200 X,Y points and
> that at each X,Y you have a temperature and velocity spectrum with 60
> points.
>
> IDL> str = {x:0, y:0, temperature:fltarr(60), velocity:fltarr(60) }
> IDL> str = replicate(str, 200)
>
> (If the number of points are not all the same in each spectra then you
> could use pointers rather than fixed arrays.)
>
> Then you fill the structure with the actual values, and finally write
> the structure to a FITS file
>
> IDL> mwrfits,str,'myfile.fits'
>
> --Wayne

Wow! that was quick.

They say if you don't know you better ask:)
Thanks Wayne, I think this will do the job.

Tigran
