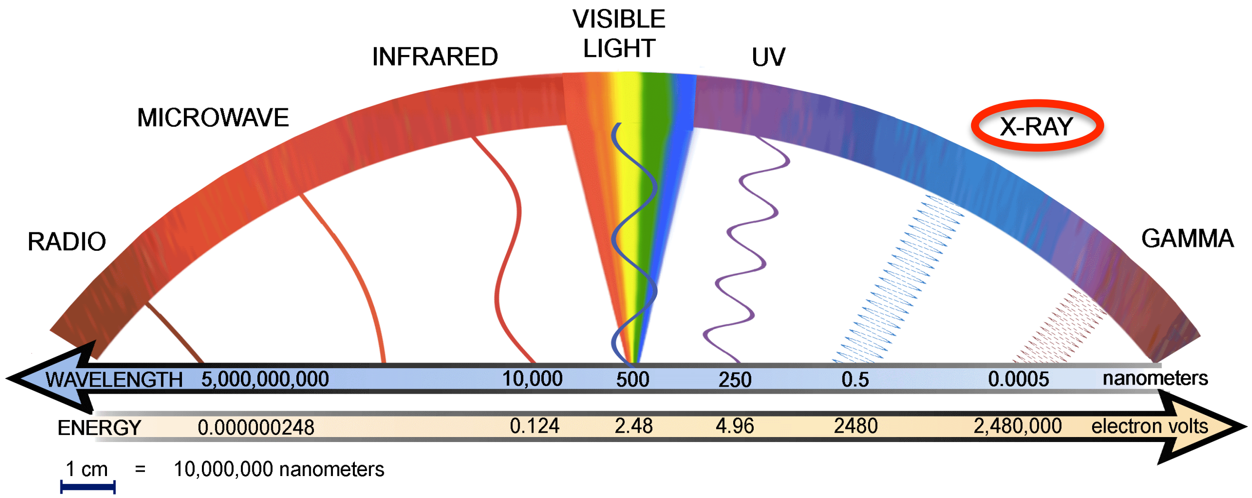
Data Analysis II: High Energy



$$E = hf = \frac{hc}{\lambda} = \frac{1240\ \mathrm{nm}}{\lambda}\ \mathrm{eV}$$

1 nm = 10^{-9}m= 10 Angstrom =0.001μ

1 eV = 1.6 10^{-12} erg

1 keV=10^3 eV

V-band@5500 Angstrom≈2.3 eV

J-band @1.25μ ≈1 eV

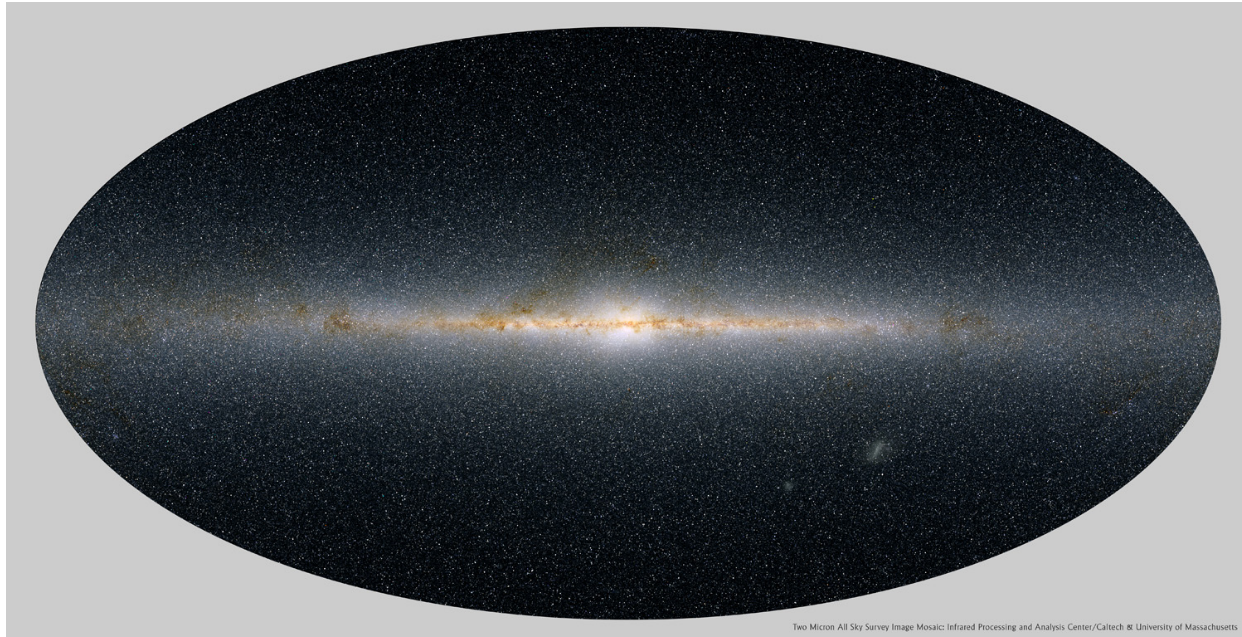
Soft X-rays: 0.1 – 1 keV

Hard X-rays: 1-100 keV

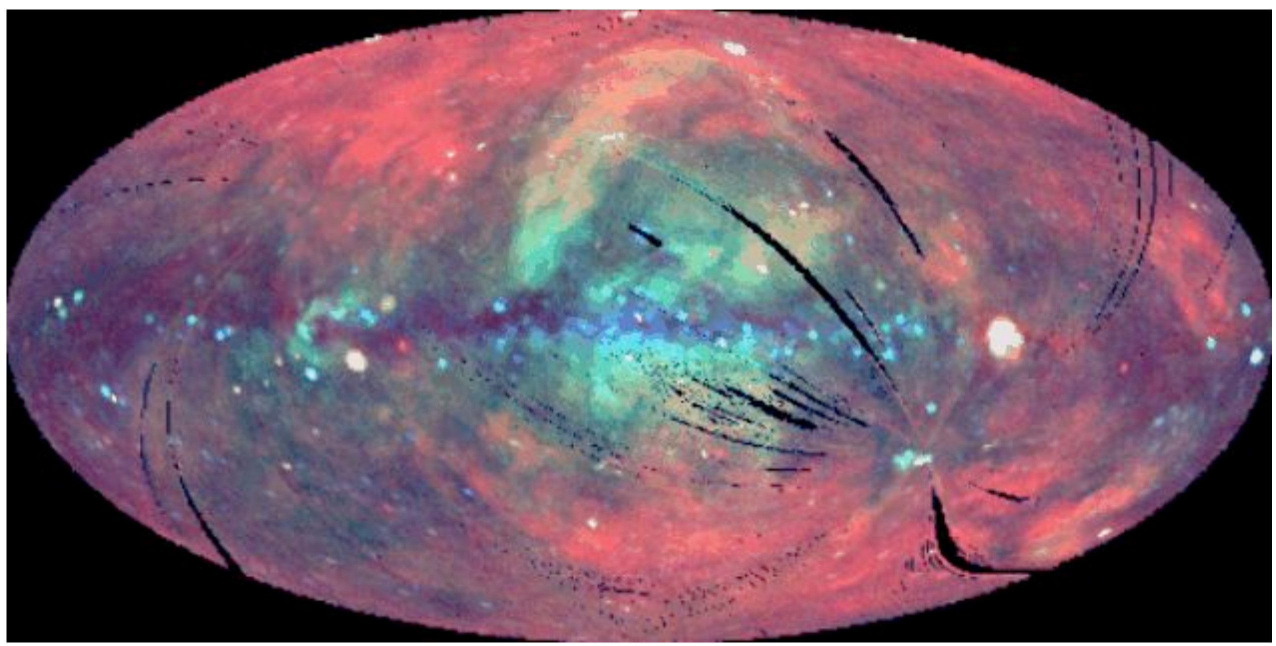
1 keV =1.6 10^{-9} ≅ 440 km/sec (E=1/2 m_pv2)

≅ 1.2 10^7 K (E=kT)

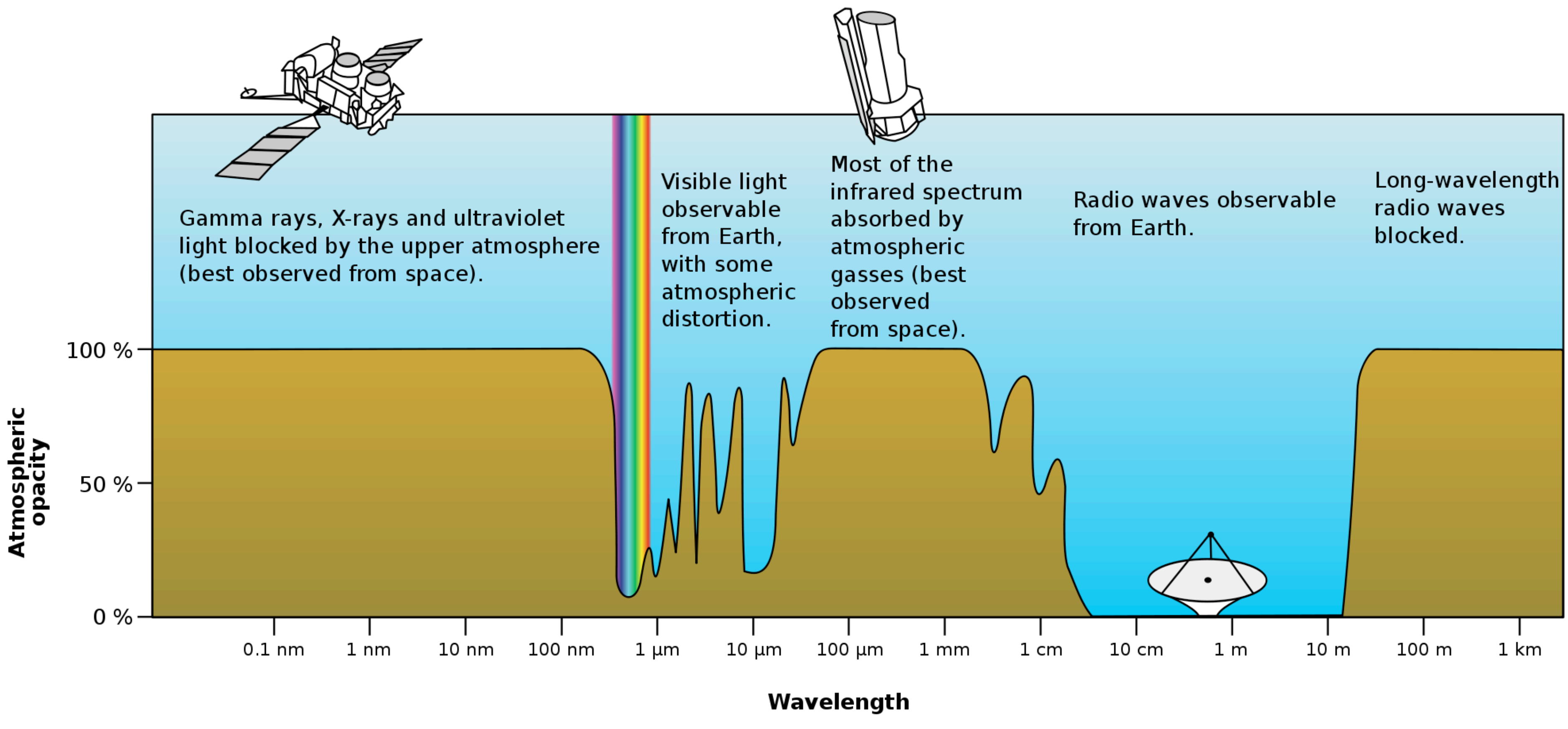
The sky in the IR (2MASS)…



…. and in X-rays (RASS)



Due to the opacity of the atmosphere, observations at high photon energies (Far UV, X-ray, γ-ray) need to be performed from either stratospheric balloon flights or from space (sounding rockets, satellites).

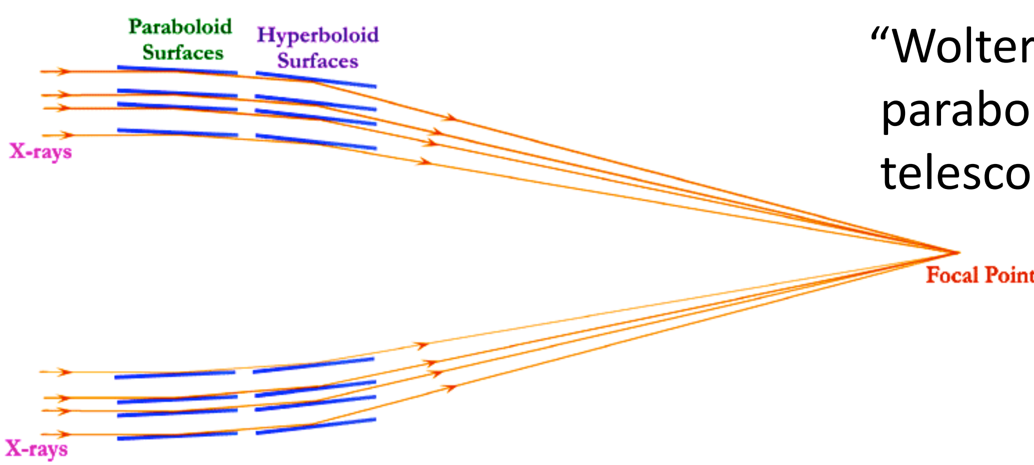


X-ray Optics: Focusing X-rays

It is tricky to focus X-rays because they pass through or are absorbed by matter at all but the smallest angles.

X-rays can be reflected off smooth metallic surfaces at very shallow angles smaller than some *critical angle* that *decreases* with *increasing energy*. Such ***grazing incidence*** reflections are particularly efficient for metals with high density, such as gold, platinum or iridium.

X-rays may be achromatically focused using *two* reflections.



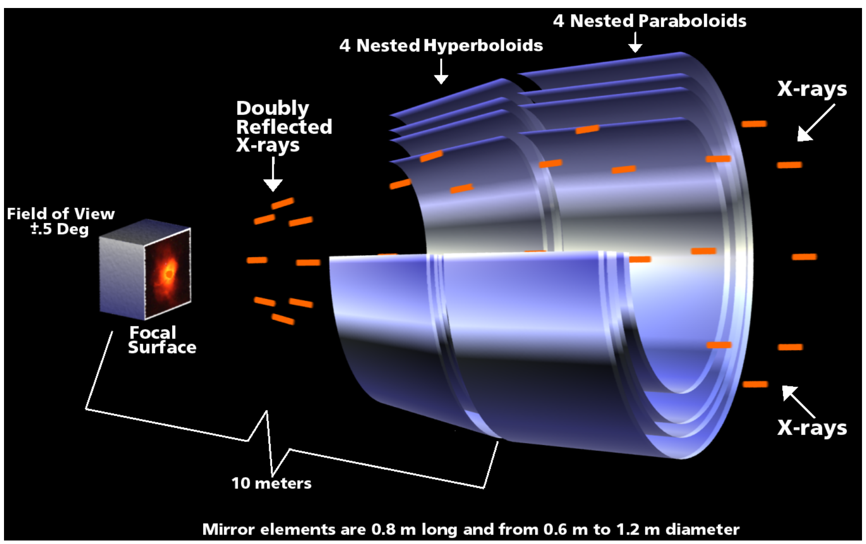
"Wolter Type I" parabolic-hyperbolic telescope

The collecting area is boosted by the nesting of mirror shells.

X-ray Telescopes: Two Designs

(1) X-ray mirrors are *highly* polished glass ceramic surface with coated with an X-ray reflective surface in the Wolter I configuration.

- Advantages: sharpest image (1" resolution)
- Disadvantages: preserve image quality ➔few shells➔ limits collecting area; difficult to fabricate, heavy (expensive to make and launch)



Chandra

smoothness of a few atoms, alignment to ≈1μ

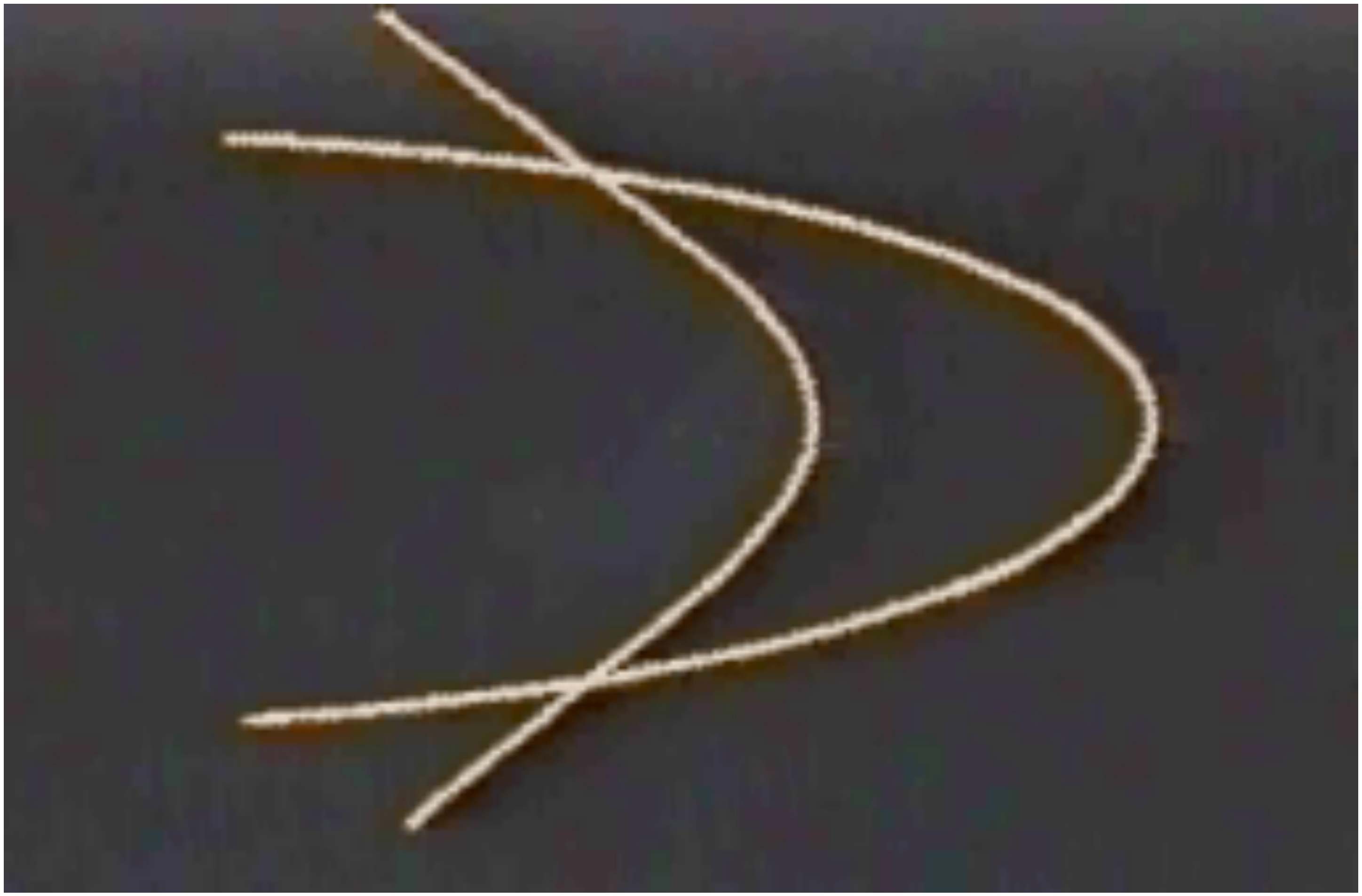
X-ray Telescopes: Two Designs

(2) Mirrors consist of *many thin foils*, coated with an X-ray reflective surface shaped into a conical approximation to the Wolter I geometry.

- Advantages: can boost the effective area by nesting *many* reflectors; more sensitive at high energies; lightweight, "easy" to fabricate (cheap)
- Disadvantages: limited image quality (≈1' angular resoluiton)

An *Suzaku* X-Ray Telescope

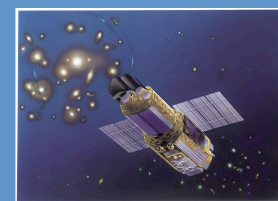
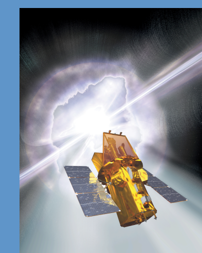
Suzaku



X-ray Detectors: CCD Cameras

- X-Ray CCD (charge coupled device) cameras are the general purpose detectors in the focal plane of X-ray Observatories currently in orbit.
- Every incident X-ray photon is recorded as an event tagged by (1) the **time** of arrival, 2) the **position** on the CCD detector, and 3) its **energy**. The result of an observation is an event list of X-ray photons.

Meet the Fleet:

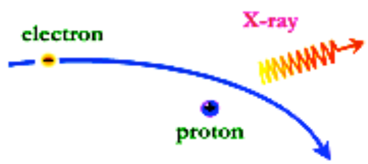
	Chandra ACIS 	XMM-Newton EPIC 	Suzaku XIS 	Swift XRT EPIC 
field of view	17’ X 17’	30’ X 30’	19’ X 19’	24’ X 24’
resolution	**1’’**	6’’	90’’	20"
bandpass	0.4 – 8 keV	0.2 – 12 keV	0.3 – 12 keV	0.2-10 keV
effective area	400 cm^2	**1200 + 2 X 900 cm^2**	400 X 3 cm^2	135 cm^2

Each of the X-ray observatories has its own data analysis and software packages: **CIAO** (*Chandra*); **SAS** (*XMM*-Newton); **Ftools** (*Suzaku* and *Swift).*

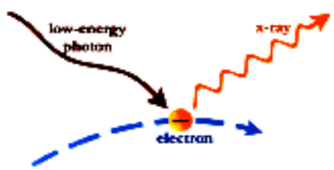
high energy particles → high energy photons

Some *X*-Ray Processes

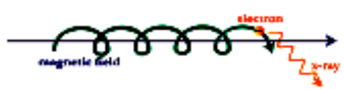
- Thermal Bremsstrahlung



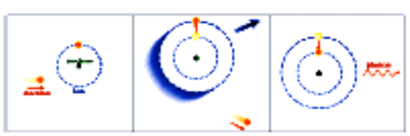
- Inverse Compton Scattering



- Synchrotron Radiation

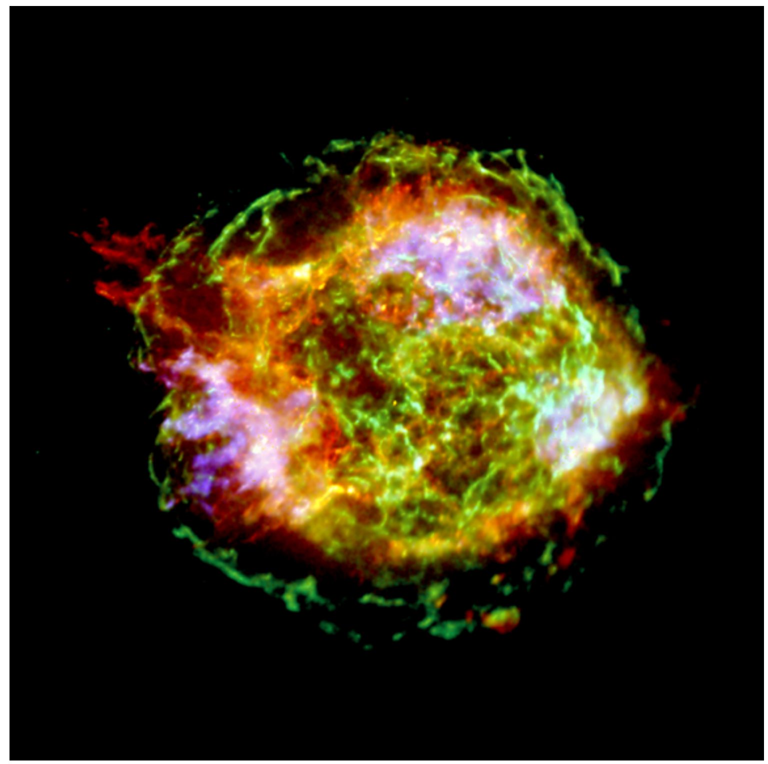


- Atomic Emission



Some X-ray Sources

- Supernova remnant, and γ-ray burst, shock waves

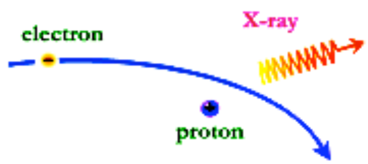


Cassiopeia A SNR

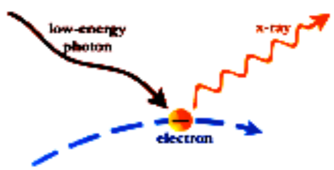
high energy particles ➔ high energy photons

Some *X*-Ray Processes

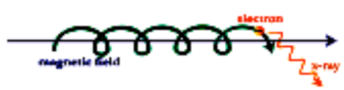
- Thermal Bremsstrahlung



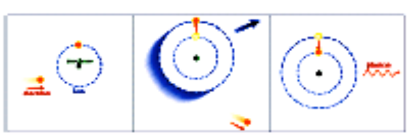
- Inverse Compton Scattering



- Synchrotron Radiation

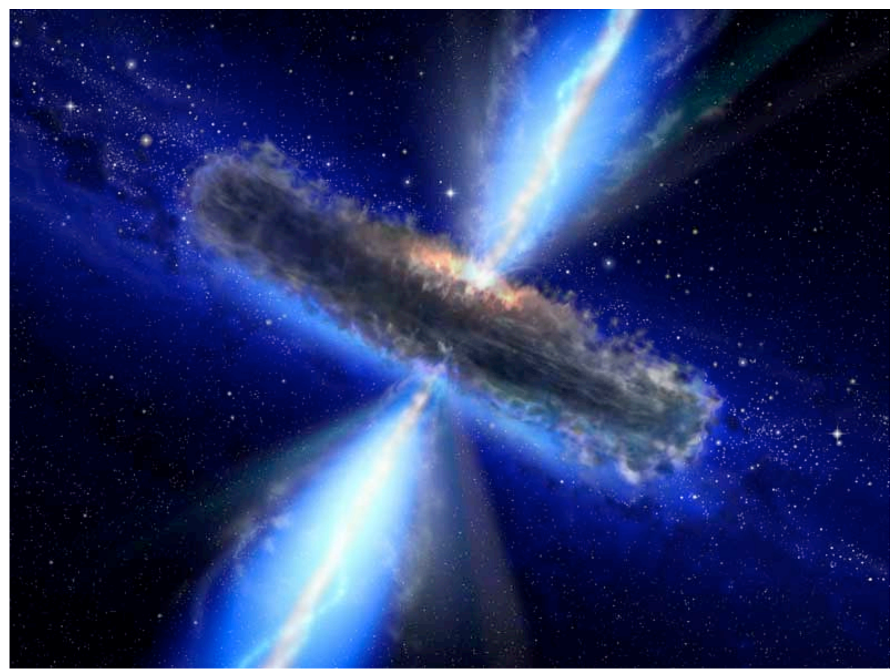


- Atomic Emission



Some X-ray Sources

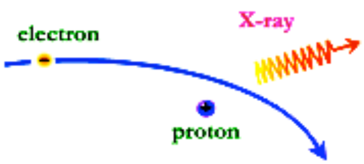
- Neutron star and (stellar and supermassive) black hole accretion disks and jets



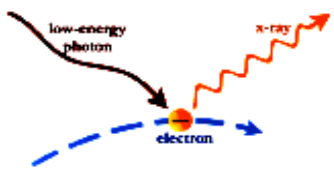
high energy particles → high energy photons

Some *X*-Ray Processes

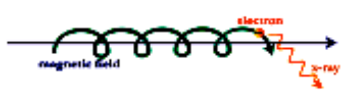
- Thermal Bremsstrahlung



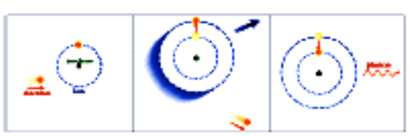
- Inverse Compton Scattering



- Synchrotron Radiation

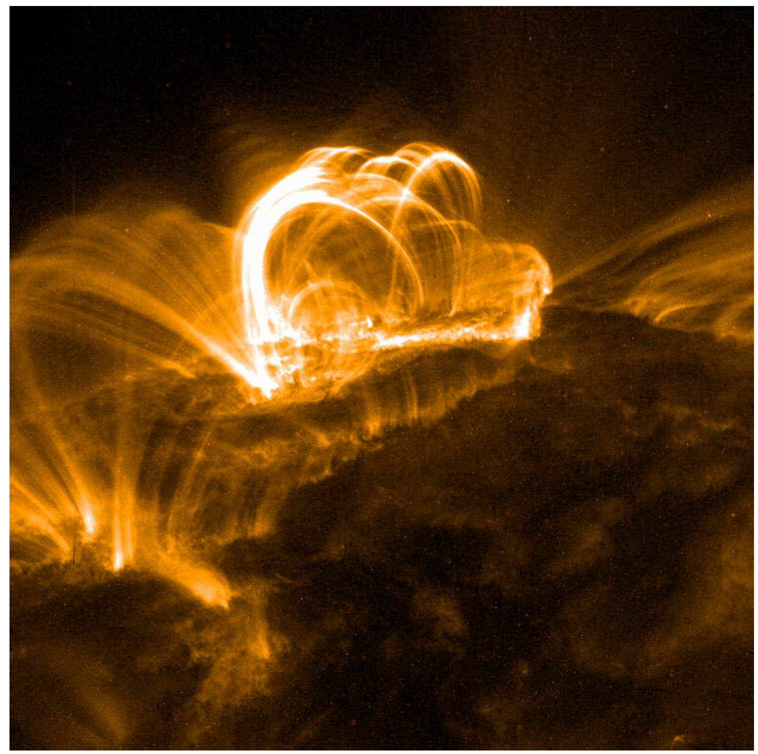


- Atomic Emission



Some X-ray Sources

- Stellar (including solar) coronae

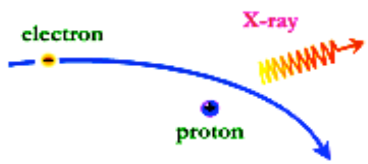


The Sun (with TRACE)

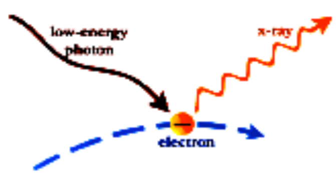
high energy particles→high energy photons

Some *X*-Ray Processes

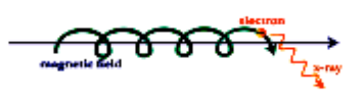
- Thermal Bremsstrahlung



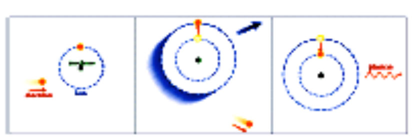
- Inverse Compton Scattering



- Synchrotron Radiation

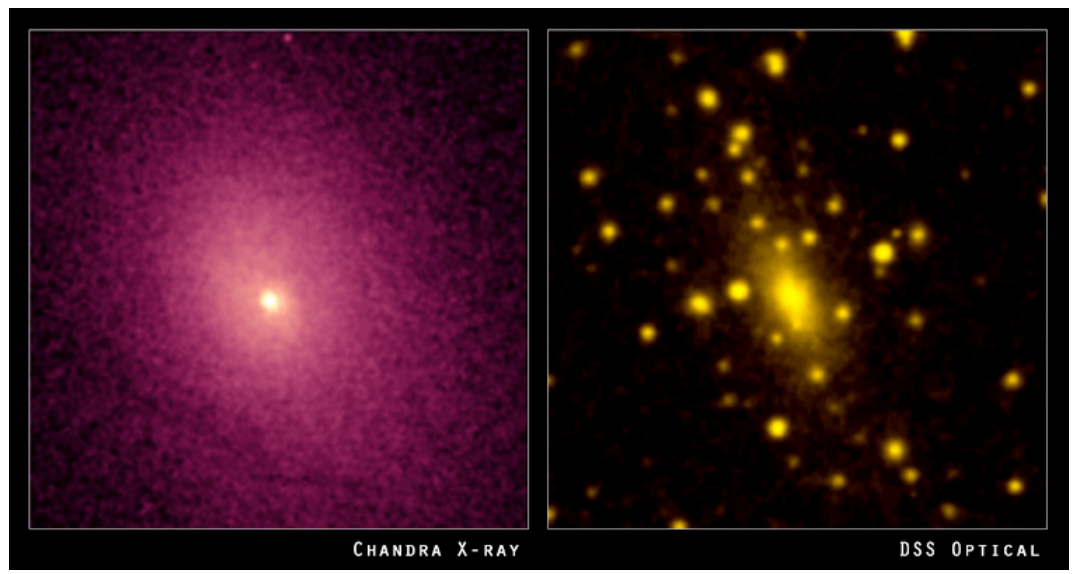


- Atomic Emission



Some X-ray Sources

- Hot thermal plasma in galaxies (including our own) and clusters of galaxies



The Abell 2029 Cluster; X-ray vs Optical

The Basics of ***FITS*** Files

- Flexible Image Transport System (***FITS***), originally developed to store and transport astronomical images, is primarily designed to store general scientific data sets in a standard way, and is self-documenting. ***FITS*** files are segmented into "Header + Data Units" (HDUs)

  (a) defined as primary or an extension, and

  (b) contain keywords describing the data, and data (tables, image arrays)

Ftools: a free suite of utility programs, distributed by NASA's High Energy Astrophysics Science Archive Research Center (HEASARC)

http://heasarc.gsfc.nasa.gov/docs/software/ftools/ftools_menu.html

General Subpackages

The following General subpackages are currently available:

Package	Description
caltools	General Calibration Tasks
fimage	General FITS Image-Manipulation Tasks
futils	General-Purpose FITS Tasks
heasarc	General Tasks for High Energy Astrophysics (**includes Xselect**)
heatools	Next-generation futils
time	Timing-Specific Tasks
xronos	The HEASARC's general timing analysis package

Mission-Specific Tasks

The following subpackages are currently available for specific missions:

- asca - ASCA Mission-Specific Tasks
- einstein - Einstein Mission-Specific Tasks
- exosat - EXOSAT Mission-Specific Tasks
- gro - CGRO Mission-Specific Tasks
- heao1 - HEAO 1 Mission-Specific Tasks
- integral - Integral Mission-Specific Tasks
- oso8 - OSO8 Mission-Specific Tasks
- rosat - ROSAT Mission-Specific Tasks
- suzaku - Suzaku Mission-Specific Tasks
- swift - Swift Mission-Specific Tasks
- vela5b - Vela 5b Mission-Specific Tasks
- xte - XTE Mission-Specific Tasks

Xselect

Xselect is a general driver which greatly simplifies many commonly performed tasks such as extracting images or spectra from events files. Xselect is distributed with the **HEASARC** package on the download page.

GUIs

Ftools Basics

- General help: *fhelp ftools*
- Specific tasks help: *fhelp taskname*
- *fdump* prints the contents of a FITS table to an ASCII file (e.g., STDOUT)
- *fv* does this in an interactive gui
- *ftlist* prints the contents of a FITS file in flexible way.
- *fstruct* lists a description of the structure of a FITS file
- *ftsat, fimgstat* compute statistics (max, min, etc.)

Ftools Basics

- ***Ftools*** has required parameters that must be specified, and optional/hidden parameters that are preset but can be overridden.
- *plist* taskname specifies these paramters.
- The required parameters can be specified (in order), or one can let ***Ftools*** prompt for these.
- The hidden parameters must be specified on the command line, if you wish to set these.
- ***Xselect***: Process an event list and extract data products (images, spectra, light curves).

For example…

plist fimgstat

```
Parameters for /Volumes/Apps_and_Docs/loew/pfiles/fimgstat.par
     infile =              Input image file:
   threshlo = INDEF        Lower threshold value:
   threshup = INDEF        Upper threshold value:
   (outfile = STDOUT)      Output image file:
      (sum = )             Sum of the included pixel values
     ...
```

```
fimgstat
Input image file:[flat_field_img.fits]
Lower threshold value:[INDEF] 1
Upper threshold value:[INDEF]
```

is equivalent to

```
fimgstat flat_field_img.fits 1 INDEF
```

```
fimgstat flat_field_img.fits 1 INDEF outfile=out.dat
```

is equivalent to

```
fimgstat outfile=out.dat
Input image file:[flat_field_img.fits]
Lower threshold value:[1]
Upper threshold value:[INDEF]
```

Yet More IDL Basics

Fitsio:

- readfits: Read a FITS file into IDL data and header variables.
- writefits: Write IDL array and header variables to a disk FITS file; if not specified a minimal fits header is created.
- sxpar: Obtain the value of a parameter in a FITS header.

see examples on next two pages…

IDL Fitsio examples:

```
IDL> image=readfits("ngc4406_suzaku_8bin_img.fits",hdr)
% Compiled module: READFITS.
% Compiled module: SXPAR.
% Compiled module: GETTOK.
% Compiled module: VALID_NUM.
% READFITS: Now reading 192 by 192 array
IDL> image=readfits("ngc4406_suzaku_8bin_img.fits",hdr)
% Compiled module: READFITS.
% Compiled module: SXPAR.
% Compiled module: GETTOK.
% Compiled module: VALID_NUM.
% READFITS: Now reading 192 by 192 array
IDL> my_name=sxpar(hdr,'OBSERVER')
IDL> print, my_name
MICHAEL LOEWENSTEIN
IDL> writefits, 'copy', image
```

```
IDL> image_copy=readfits("copy",newhdr)
% READFITS: Now reading 192 by 192 array
IDL> help,image_copy
IMAGE_COPY      LONG      = Array[192, 192]
IDL> help,newhdr
NEWHDR          STRING    = Array[10]
IDL> print,newhdr
SIMPLE  =                    T / Written by IDL:  Mon
Nov  1 11:37:29 2010
BITPIX  =                   32 / Number of bits per
data pixel
NAXIS   =                    2 / Number of data axes
NAXIS1  =                  192 /
NAXIS2  =                  192 /
EXTEND  =                    T / FITS data may contain
extensions
DATE    = '2010-11-01'         / Creation UTC
(CCCC-MM-DD) date of FITS header
COMMENT FITS (Flexible Image Transport
System) format is defined in 'Astronomy
COMMENT and Astrophysics', volume 376,
page 359; bibcode 2001A&A...376..359H
END
```

More on IDL Arrays

```
IDL> array_1=fltarr(5); create an array of zeros
IDL> print,array_1
     0.00000      0.00000      0.00000      0.00000      0.00000
IDL> array_2=findgen(5); create an array with values = index
IDL> print,array_2
     0.00000      1.00000      2.00000      3.00000      4.00000
IDL> array_3=replicate(1.0,5); create an array of ones
IDL> print,array_3
     1.00000      1.00000      1.00000      1.00000      1.00000
IDL> array_4=2.0*array_2;  multiply all array_2 elements by 2....
IDL> print,array_4
     0.00000      2.00000      4.00000      6.00000      8.00000
IDL> array_5=2.0*array_2+array_3; … and add the elements of array_3
IDL> print,array_5
     1.00000      3.00000      5.00000      7.00000      9.00000
IDL> indices=[0,2]
```

More on IDL Arrays

```
IDL> array_6=array_5(indices); subarray of 1st and 3rd  entries
IDL> print,array_6
    1.00000      5.00000
IDL> array_7=array_5; copy
IDL> print,array_5
    1.00000      3.00000      5.00000      7.00000      9.00000
IDL> array_7(indices)=0.0; zero out 1st and 3rd  entries
IDL> print,array_7
    0.00000      3.00000      0.00000      7.00000      9.00000
IDL> nonzeros=where(array_7 gt 0.0); indices where array is positive
IDL> print,nonzeros
           1           3           4
IDL> array_8=array_1; copy
IDL> array_8(nonzeros)=1.0; replace specified entries
IDL> print,array_8
    0.00000      1.00000      0.00000      1.00000      1.00000
```