Appendix A

Parameter files used in preprocessing of GIRT data

GIRT data reduction was performed through the Image Reduction and Analysis Facility (IRAF) package. This package is the compilation of different tasks under different packages. The IRAF is made available to the astronomical community by the National Optical Astronomy Observatories (NOAO), USA. Basically, preprocessing includes removal of all instrumental signatures.

A.1 Conversion of images (FITS) to individual IRAF images

This package is used to convert the FITS images to IRAF images. Here in the table, @fitslist and @imhlist are the files which contains the list of all input and output file names for the images respectively.

A.2 Resetting the image coordinate system

This task resets the coordinates of the image. Here @out is the list of input and output file where images are stored and for which the coordinates will be set.

A.3 Detecting and replacing cosmic rays

This task detects the cosmic rays in the image and will be replaced. Here @out is the list of input and output file for which the cosmicrays will be detected and replaced.

A.4 Arithmetic on the image

This task is used for arithmetic operations like addition, substraction, multiplication and division of images. Here s3, - , s5, s3m5fits are the user dependant names and operations.

Table A.1: *PACKAGE = dataio TASK = rfits*

fits_fil=	@fitslist	FITS data source
file_lis=	*	File/extensions list
iraf_fil=	@imhlist	IRAF filename
(make_im=	yes)	Create an IRAF image?
(long_he=	no)	Print FITS header cards?
(short_h=	yes)	Print short header?
(datatyp=	)	IRAF data type
(blank =	0.)	Blank value
(scale =	yes)	Scale the data?
(oldiraf=	no)	Use old IRAF name in place of iraf_file?
(offset =	0)	Tape file offset
(mode =	ql)	

Table A.2: *PACKAGE = imcoords TASK = wcsreset*

image =	@out	List of input images
wcs =	world	Name of wcs to be initialized
(verbose=	yes)	Print messages about actions taken ?
(mode =	ql)	

Table A.3: *PACKAGE = crutil TASK = cosmicrays*

input =	@out	List of images in which to detect cosmic rays
output =	@out	List of cosmic ray replaced output images (optional)
(crmasks=	)	List of bad pixel masks (optional)
(thresho=	25.)	Detection threshold above mean
(fluxrat=	2.)	Flux ratio threshold (in percent)
(npasses=	5)	Number of detection passes
(window =	5)	Size of detection window
(interac=	no)	Examine parameters interactively?
(train =	no)	Use training objects?
(objects=	)	Cursor list of training objects
(savefil=	)	File to save train objects
(plotfil=	)	Plot file
(graphic=	stdgraph)	Interactive graphics output device
(cursor =	)	Graphics cursor input
answer =	yes	Review parameters for a particular image?
(mode =	ql)	

Table A.4: *PACKAGE = imutil TASK = imarith*

operand1=	s3	Operand image or numerical constant
op =	-	Operator
operand2=	s5	Operand image or numerical constant
result =	s3m5.fits	Resultant image
(title =	)	Title for resultant image
(divzero=	0.)	Replacement value for division by zero
(hparams=	)	List of header parameters
(pixtype=	)	Pixel type for resultant image
(calctyp=	)	Calculation data type
(verbose=	no)	Print operations?
(noact =	no)	Print operations without performing them?
(mode =	ql)	

Appendix B

Parameter files used in processing of GIRT data

Processing includes aperture extraction, wavelength calibration and relative flux calibration of the data (star).

B.1 Extract one dimensional sums across the apertures

This task is used for extracting the one dimensional sums across the selected image's aperture. Below mentioned parameter file tells that the input file given for the aperture extraction is s2m6 and the region around the image is from -60:-50,20:30 rows above and below the aperture on which *chebyshev* function will be fitted. The output image will be s2m6sp.

B.2 Wavelength calibration

This task is used for identifying the features in one dimensional image. The image used is the sky image. Input file is d2add, central line of the image is 68 and what is to identify is the absorption features of OH molecule.

B.3 Assigning reference spectra

After identifying the features in the sky frame this task assign the value to each pixel in extracted image and we say that the spectra is wavelength calibrated. Here, d2add reference image used for s2m6sp.

B.4 Dispersion correction to the spectra

This task is used after wavelength calibration to the extracted spectra. This task does the dispersion correction to the image. Here, s2m6sp is the input image for which the dispersion is corrected and it is saved as es2m6sp.fits.

B.5 Spectrum arithmetic

After completion of wavelength calibration of standard and programme star this task is used for the applying the arithmetic operations. Here, es2m6sp is the input file of the programme star (HR 4433) which then divided by standard star 3314j (j for j-band region) giving rise to reduced spectra with 4433j.fits.

Table B.1: *PACKAGE = apextract TASK = apall*

input =	s2m6	List of input images
(output =	s2m6sp)	List of output spectra
(apertur=	)	Apertures
(format =	onedspec)	Extracted spectra format
(referen=	)	List of aperture reference images
(profile=	)	List of aperture profile images
(interac=	yes)	Run task interactively?
(find =	yes)	Find apertures?
(recente=	yes)	Recenter apertures?

```
(resize =             yes)  Resize apertures?
(edit =               yes)  Edit apertures?
(trace =              yes)  Trace apertures?
(fittrac=             yes)  Fit the traced points interactively?
(extract=             yes)  Extract spectra?
(extras =              no)  Extract sky, sigma, etc.?
(review =             yes)  Review extractions?

(line =             INDEF)  Dispersion line
(nsum =                10)  Number of dispersion lines to sum or median

                            # DEFAULT APERTURE PARAMETERS

(lower =              -5.)  Lower aperture limit relative to center
(upper =               5.)  Upper aperture limit relative to center
(apidtab=                )  Aperture ID table (optional)

                            # DEFAULT BACKGROUND PARAMETERS

(b_funct=       chebyshev)  Background function
(b_order=               1)  Background function order
(b_sampl=  -60:-50,20:30)  Background sample regions
(b_naver=              -3)  Background average or median
(b_niter=               0)  Background rejection iterations
(b_low_r=              3.)  Background lower rejection sigma
(b_high_=              3.)  Background upper rejection sigma
(b_grow =              0.)  Background rejection growing radius
                            # APERTURE CENTERING PARAMETERS
nfind =                  1  Number of apertures to be found automatically
(width =               5.)  Profile centering width
```

```
(radius =          10.)  Profile centering radius
(thresho=           0.)  Detection threshold for profile centering

                         # AUTOMATIC FINDING AND ORDERING PARAMETERS
(minsep =           5.)  Minimum separation between spectra
(maxsep =        1000.)  Maximum separation between spectra
(order =    increasing)  Order of apertures

                         # RECENTERING PARAMETERS

(aprecen=             )  Apertures for recentering calculation
(npeaks =       INDEF)   Select brightest peaks
(shift =          yes)   Use average shift instead of recentering?

                         # RESIZING PARAMETERS

(llimit =       INDEF)   Lower aperture limit relative to center
(ulimit =       INDEF)   Upper aperture limit relative to center
(ylevel =         0.1)   Fraction of peak or intensity for automatic width
(peak =           yes)   Is ylevel a fraction of the peak?
(bkg =            yes)   Subtract background in automatic width?
(r_grow =          0.)   Grow limits by this factor
(avglimi=          no)   Average limits over all apertures?
```

```
                    # TRACING PARAMETERS

(t_nsum =        10)  Number of dispersion lines to sum
(t_step =        10)  Tracing step
(t_nlost=         3)  Number of consecutive times profile is lost before quitting
(t_funct=  legendre)  Trace fitting function
(t_order=         2)  Trace fitting function order
(t_sampl=         *)  Trace sample regions
(t_naver=         1)  Trace average or median
(t_niter=         0)  Trace rejection iterations
(t_low_r=        3.)  Trace lower rejection sigma
(t_high_=        3.)  Trace upper rejection sigma
(t_grow =        0.)  Trace rejection growing radius

                    # EXTRACTION PARAMETERS
(backgro=   average)  Background to subtract
(skybox =         1)  Box car smoothing length for sky
(weights=      none)  Extraction weights (none | variance)
(pfit =       fit1d)  Profile fitting type (fit1d | fit2d)
(clean =         no)  Detect and replace bad pixels?
(saturat=     INDEF)  Saturation level
(readnoi=        0.)  Read out noise sigma (photons)
(gain =          1.)  Photon gain (photons/data number)
(lsigma =        4.)  Lower rejection threshold
(usigma =        4.)  Upper rejection threshold
(nsubaps=         1)  Number of subapertures per aperture
(mode =          ql)
```

Table B.2: *PACKAGE = onedspec TASK = identify*

images =	d2add	Images containing features to be identified
(section=	line 68)	Section to apply to two dimensional images
(databas=	database)	Database in which to record feature data
(coordli=	jsky.dat)	User coordinate list
(units =	)	Coordinate units
(nsum =	10)	Number of lines/columns/bands to sum in 2D images
(match =	-3.)	Coordinate list matching limit
(maxfeat=	50)	Maximum number of features for automatic identification
(zwidth =	100.)	Zoom graph width in user units
(ftype =	absorption)	Feature type
(fwidth =	4.)	Feature width in pixels
(cradius=	5.)	Centering radius in pixels
(thresho=	0.)	Feature threshold for centering
(minsep =	2.)	Minimum pixel separation
(functio=	spline3)	Coordinate function
(order =	1)	Order of coordinate function
(sample =	*)	Coordinate sample regions
(niterat=	0)	Rejection iterations
(low_rej=	3.)	Lower rejection sigma
(hig_re=	3.)	Upper rejection sigma
(grow =	0.)	Rejection growing radius
(autowri=	no)	Automatically write to database
(graphic=	xterm)	Graphics output device
(cursor =	)	Graphics cursor input
crval =		Approximate coordinate (at reference pixel)
cdelt =		Approximate dispersion
(aidpars=	)	Automatic identification algorithm parameters
(mode =	ql)	

Table B.3: *PACKAGE = onedspec TASK = refspectra*

input =	s2m6sp	List of input spectra
(referen=	d2add)	List of reference spectra
(apertur=	)	Input aperture selection list
(refaps =	)	Reference aperture selection list
(ignorea=	yes)	Ignore input and reference apertures?
(select =	interp)	Selection method for reference spectra
(sort =	)	Sort key
(group =	)	Group key
(time =	no)	Is sort key a time?
(timewra=	17.)	Time wrap point for time sorting
(overrid=	no)	Override previous assignments?
(confirm=	yes)	Confirm reference spectrum assignments?
(assign =	yes)	Assign the reference spectra to the input spectrum?
(logfile=	STDOUT,logfile)	List of logfiles
(verbose=	no)	Verbose log output?
answer =	yes	Accept assignment?
(mode =	ql)	

Table B.4: *PACKAGE = onedspec TASK = dispcor*

input =	s2m6sp	List of input spectra
output =	es2m6sp.fits	List of output spectra
(lineari=	yes)	Linearize (interpolate) spectra?
(databas=	database)	Dispersion solution database
(table =	)	Wavelength table for apertures
(w1 =	INDEF)	Starting wavelength
(w2 =	INDEF)	Ending wavelength
(dw =	INDEF)	Wavelength interval per pixel
(nw =	INDEF)	Number of output pixels
(log =	no)	Logarithmic wavelength scale?
(flux =	yes)	Conserve flux?
(samedis=	no)	Same dispersion in all apertures?
(global =	no)	Apply global defaults?
(ignorea=	no)	Ignore apertures?
(confirm=	no)	Confirm dispersion coordinates?
(listonl=	no)	List the dispersion coordinates only?
(verbose=	yes)	Print linear dispersion assignments?
(logfile=	)	Log file
(mode =	ql)	

Table B.5: *PACKAGE = onedspec TASK = sarith*

input1 =	es2m6sp	List of input spectra
op =	/	Operation
input2 =	3314j	List of input spectra or constants
output =	4433j.fits	List of output spectra
(w1 =	INDEF)	Starting wavelength
(w2 =	INDEF)	Ending wavelength
(apertur=	)	List of input apertures or columns/lines
(bands =	)	List of input bands or lines/bands
(beams =	)	List of input beams or echelle orders
(apmodul=	0)	Input aperture modulus (0=none)
(reverse=	no)	Reverse order of operands in binary operation?
(ignorea=	no)	Ignore second operand aperture numbers?
(format =	multispec)	Output spectral format
(renumbe=	no)	Renumber output apertures?
(offset =	0)	Output aperture number offset
(clobber=	no)	Modify existing output images?
(merge =	no)	Merge with existing output images?
(rebin =	yes)	Rebin to exact wavelength region?
(errval =	0.)	Arithmetic error replacement value
(verbose=	no)	Print operations?
(mode =	ql)	