	***Title*: Flagging "Narrow" bandwidth RFI in Spectral Line Data**	***Author*:** Minter	**GBT Memo # 318** **RFI memo # 161**

In

Flagging "Narrow" bandwidth RFI in Spectral Line Data

Anthony (Toney) Minter

GBT Memo # 318
RFI Memo # 161

Change Record

VERSION	DATE	REASON
1.0	10 June, 2025	First draft
1.1	24 June, 2025	Revisions from IPG group comments

Title: Flagging of "Narrow" bandwidth RFI	*Author*: Minter	GBT Memo # 318 RFI memo # 161

CONTENTS

1 INTRODUCTION

Over the years many observers have created different routines to flag RFI in their spectral line data. Very few of these routines have been documented and made available to all GBT observers. This memo briefly described RFI flagging routines that I developed which should be of interest to spectral line observers.

The flagging method described in this memo is designed for in-band frequency switching data with narrow bandwidth RFI signals in the presence of strong broad spectral lines such as the 21 cm line of Hydrogen. The method will also work when the spectral lines are weak and generally not detectable in a single scan or integration. The method is not very suitable when the RFI is resolved and covers many spectral channels.

Although the code is written for in-band frequency switching data, it should be extendable to out of band frequency switching and position switching data with minor modifications.

The routines are written for GBTIDL but should be portable to Dysh (the GBT python data reduction package currently under development). The routines could provide a basis for RFI flagging for any pipeline that is created for GBT data.

2 EXAMPLES OF RFI ENCOUNTERED

An example of narrow bandwidth RFI seen in frequency switched data for the 21cm line of Galactic HI is shown in Figure 1. There are two RFI signals that narrow, being only one channel wide. One is at 1419.1 MHz (with its "ghost" appearing at about 1414.1 MHz due to the flip/fold process in reducing the frequency switched data). A second is folded into the spectrum at 1421.7 MHz from its true frequency of about 1426.7 MHz. The Galactic HI 21cm emission is obvious in this single scan. Any routine to remove RFI will need to be able to avoid removing any part of the HI signal.

An example of narrow bandwidth RFI seen in frequency switched data for Galactic OH (1720 MHz satellite line) is shown in Figure 2. The RFI signal is at 1725 MHz (with its folded "ghost" at 1721 MHz). The Galactic OH signal is very weak and not detectible in this single scan. There is also an RFI signal at 1713.5 MHz that is folded to appear at 1717.5 MHz. When removing RFI for this data we do not have to worry about accidently flagging the OH line.

A more complex example of RFI is shown in Figure 3 around the Galactic OH satellite line at 1612 MHz. There is narrow, but wider than a single channel, RFI above 1618 MHz. The folded "ghost" of a broad RFI "band" from 1604.7 to 1605.6 MHz is seen between 1608.7 and 1609.6 MHz.

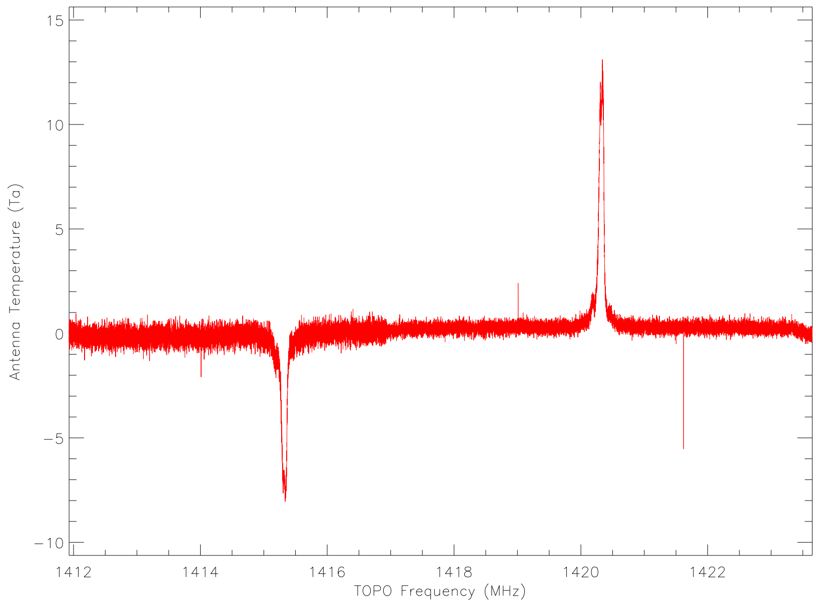

Figure 1

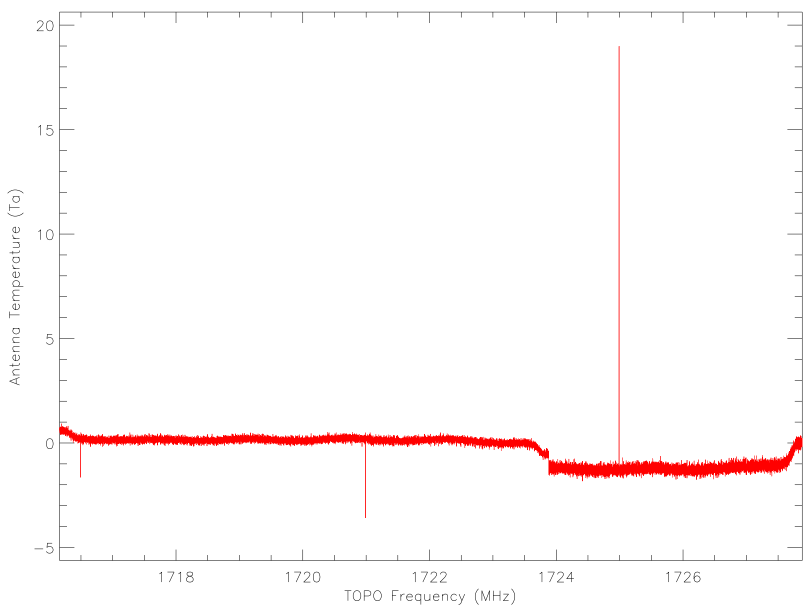

Figure 2

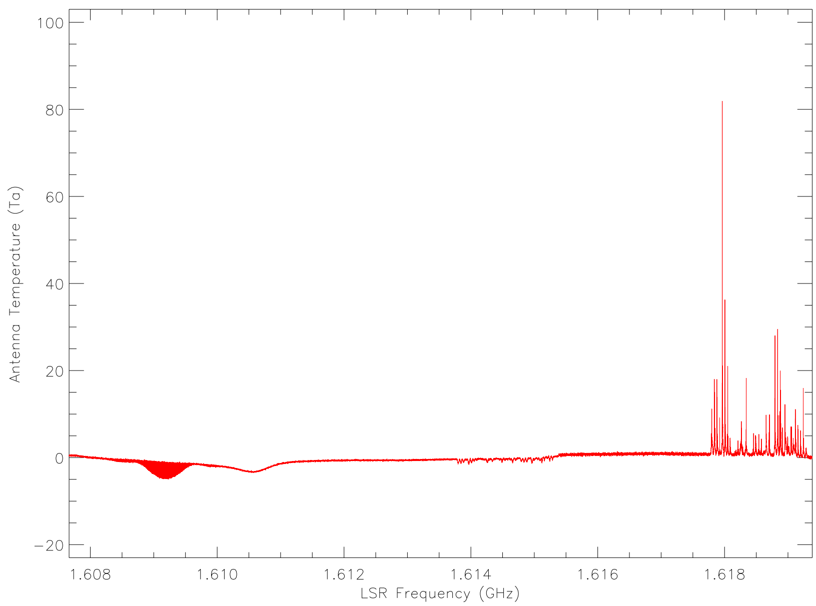

Figure 3

3 FSKURTOSISFLAG

3.1 Description

This routine is written to work on in-band frequency switched data and was written for GBTIDL.

The routine steps through every other integration in the sdfits file. This allows the GBTILD gettp command to be used to get a spectrum for each signal and reference spectrum in the frequency switched data. For each spectrum the kurtosis for channel n is determined from the data in channels n-kwdith through n+kwidth with kwidth being specified by the user. The first kwidth and last kwidth channels do not have kurtosis values determined and are set to zero. If the kurtosis for a channel in each spectrum is above the user defined klimit value and the frequency is outside of the user defined myokfreqs ranges, then the data in that channel is replaced using the GBTIDL replace command. If the kurtosis of any channel is greater than the user defined kmax value, then that channel is replaced regardless of the frequency. The routine checks for consecutive channels that need to be replaced to make sure that only good channels are used for the GBTIDL replace command.

The routine works for data from both the ACS spectrometer and VEGAS. The input file name for the data set is used to determine if the data are for VEGAS (i.e. have raw.vegas in the name) and are otherwise assumed to be data are from the ACS spectrometer.

The routine sets the channel axis to use the Topocentric reference frame with the frequency being in MHZ. The input values for myokfreqs (which specifies areas where spectral lines are expected, and the data should not be replaced) must also be in MHz.

The Galactic 21cm HI spectra can have sharp peaks, either from narrow line or from self-absorption. This can result in higher kurtosis values. The routine allows the user to input frequencies to avoid replacing around known spectral lines using the myokfreqs parameter – a 2*n dimensional array of frequencies bounding the n spectral lines.

Based on experience with 21 cm HI data and OH 1612, 1665, 1667, and 1720 MHz data the following default values are used:

Kwidth	50
Klimit	2.25
Kmax	15
myokfreqs	[1420.,1421.]

Note that weak RFI may not be strong enough to be "seen" in individual integrations but could be seen in the data averaged from the whole scan. This routine will not find and replace all these RFI signals.

The code for FSKURTOSISFLAG is included in Appendix A.

3.2 Examples

Figures 4 and 5 show the result of running FSKURTOSISFLAG on the data presented in Figure 1. The default values for FSKURTOSISFLAG were used. The RFI at 1426.7 MHz (folded to 1421.7 MHz) has been fully flagged by the routine. The RFI at 1414.1 (frequency shifted to 1419.1 MHz) is only partially flagged. This is due to the weakness of the signal in individual integrations. Further flagging will be needed to remove the 1414.1 MHz RFI from the data.

Figures 6 and 7 show the result of running FSKURTOSISFLAG on the data presented in Figure 2. The default values for FSKURTOSISFLAG were used. The full RFI signal at 1725 MHz has been flagged. The 1713.5 MHz RFI signal (folded to 1717.5 MHz) to near the edge of the band and has not been flagged as it appears in the first kwidth channels of the spectrum. Further flagging will be needed to remove the 1713.5 MHz RFI from the data.

Figure 8 shows how the FSKURTOSIS routine fails to flag all the RFI shown in Figure 3 that is resolved into many spectral channels. FSKURTOSIS does flag some, but not all, of the strong RFI that is only a few channels wide.

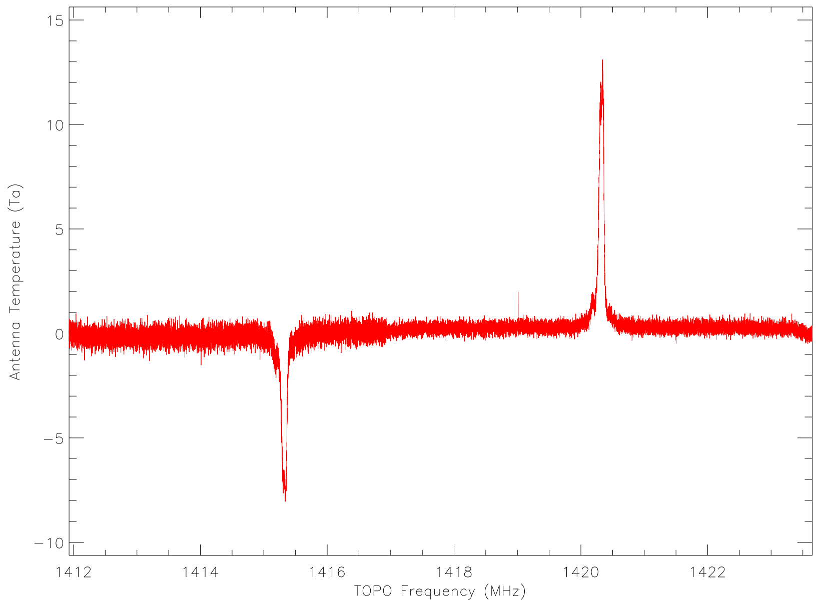

Figure 4

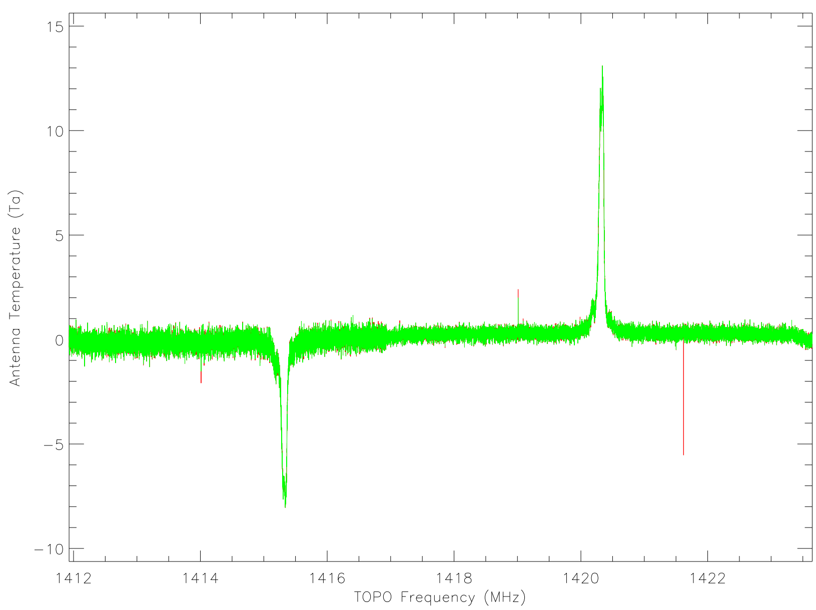

Figure 5: Raw data is shown in red. Data after running FSKURTOSISFLAG is shown in green.

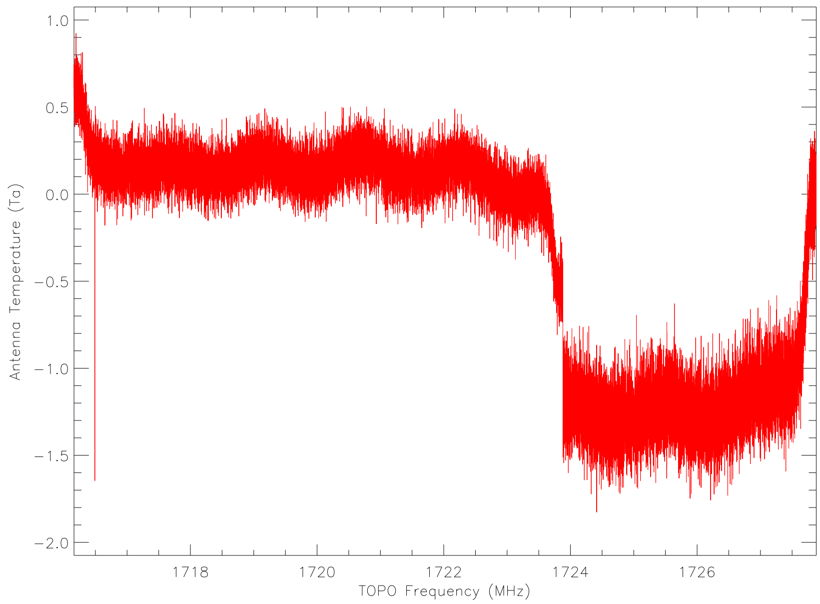

Figure 6

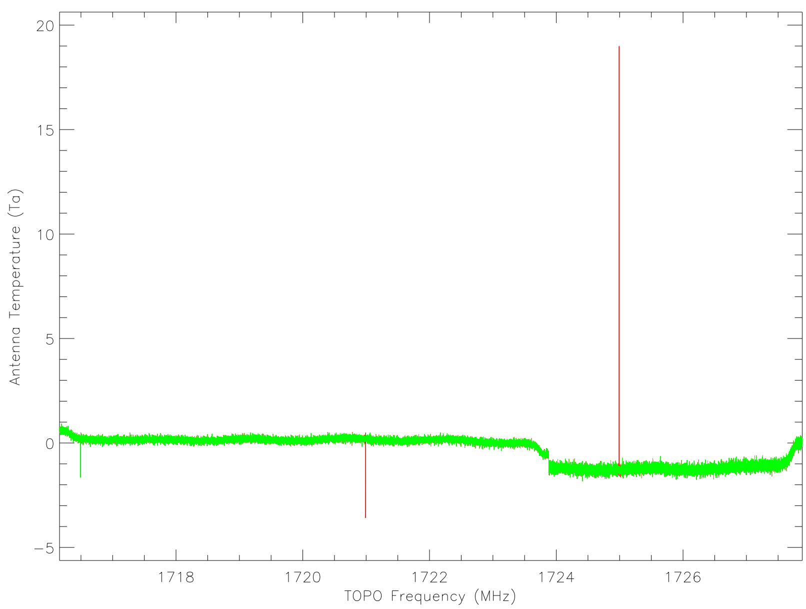

Figure 7: Raw data is shown in red. Data after running FSKURTOSISFLAG is shown in green.

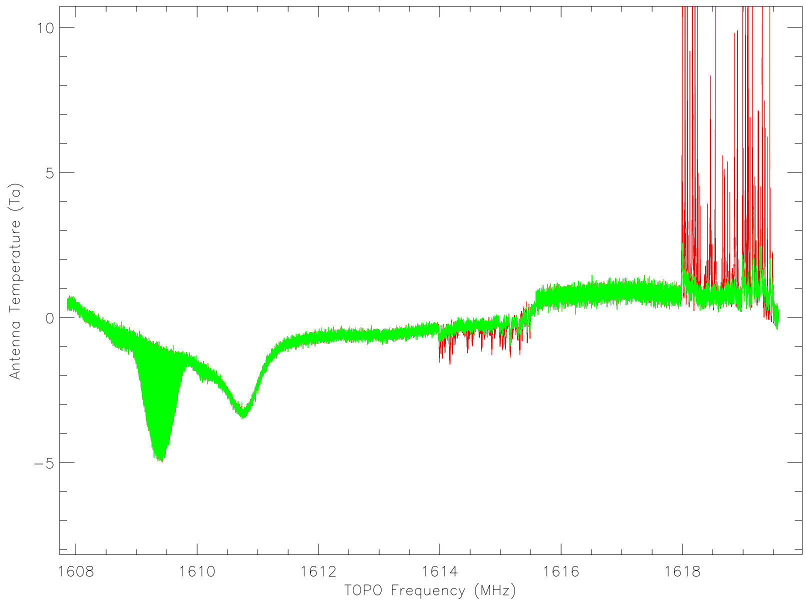

Figure 8: Raw data is shown in red. Data after running FSKURTOSISFLAG is shown in green.

4 REPLACERFIFREQ

4.1 Description

Since FSKURTOSISFLAG cannot flag all signals, another simple routine, RPLACERFIFREQ, was created to flag any remaining RFI signals. This routine just uses a given frequency and frequency half-width to determine the channels to be replaced.

REPLACERFIFREQ assumes that the data are from the output of FSKURTOSISFLAG. If REPLACERFIFREQ is to be run on raw data, then a simple modification would be needed to ensure that the INPUTFILE could be read correctly.

The code for REPLACERFIFREQ is included in Appendix B.

APPENDIX A: THE FSKURTOSISFLAG CODE

```
pro fskurtosisflag, inputfile, outputfile, klimit=klimit, kwidth=kwidth, kmax=kmax, myokfreqs=myokfreqs
;+
; NAME:
;     FSKURTOSISFLAG
;
; PURPOSE:
;     Replace by interpolation any narrow RFI signals in frequency
;     swithed data.  Experience suggests the majority of kurtosis
;     values are below 2 even when strong spectral lines such as the
;     Galactice 21 cm emission are present
;
; CALLING SEQUENCE:
;     fskurtosisflag, inputfile, outputfile [, klimit=klimit] [, kwidth=kwidth]
;     [, kmax=kmax] [,myokfreqs=myokfreqs]
;
; INPUTS:
;     INPUTFILE - the raw data file with the frequency swicthed data
;     OUTPUTFILE - the data file to which the RFI corrected data will be written
;     KLIMIT - kurtosis values equal to or above klimit are flagged
;          if they are not withing the frequenies defined by
;          myokfreqs. Default value is 2.25 if not specified.
;     KWIDTH - the kurtosis for channel n is determined from channel
;          n-kwidth through n+kwidth. Note that data will not be
;          flagged in the first kwidth and last kwidth channels.
;          Default value is 50 channels if not specified.
;     KMAX - if the kurtosis value is above kmax then those channels
;         will be flagged regardless of other parameters.  Default
;         value is 15.
;     MYOKFREQS - is an array of frequency ranges that are not to be
;            flagged.  The array should be 2*n elements.  These
;            should be a mininum and maximum frequency in the
;            topocentric frame around the spectral lines
;            expected in the data.  The units should be MHz.  If
;            no values are entered then all frequencies will be
```

```
;              considered for flagging.
;
; Examples:
;
; Typical use of fskurtosisflag for Galactic HI:
;     gskurtosisflag,'baddata.raw.vegas','gooddata.fits'
;
; Typical use of fskurtosisflag for an OH spectral lines
;     gskurtosisflag,'baddata.raw.vegas','gooddata.fits',myokfreqs=[1719.,1721.]
;
; Use of fskurtosisflag specifying all inputs
;     gskurtosisflag,'baddata.raw.vegas','gooddata.fits',klimit=1.75, kwidth=40,
;         kmax=100.,myokfreqs=[1664.,1668.]
;
;  Modification History
;     Written by Anthony Minter, GBO/NRAO 9 Jun 2025
;

 ; is this spectrometer data or vegas data
  splitfile = strsplit(inputfile,".",/EXTRACT)
  if splitfile[2] eq 'acs' then begin
    filein,inputfile
  endif else begin
    dirin,inputfile
  endelse

  ; set the output file
  fileout,outputfile

 ; if kurtosis is above this value then we should flag/replace the data
 if n_elements(klimit) eq 0 then klimit=2.25

 ; half width of channels over which to determine kurtosis
 if n_elements(kwidth) eq 0 then kwidth=50

 ; if kurtosis is above this value then definitely flag the data regardless
 if n_elements(kmax) eq 0 then kmax=15.
```

```
; assume observing HI if no frequencies providedflag
if n_elements(myokfreqs) eq 0 then myokfreqs = [1420.,1421.]

numfreq=n_elements(myokfreqs)

; get first record and set x axis
getrec,0
setframe,'TOPO'
setxunit,'MHz'

; how many records in data set
nrec=nrecords()

; loop through records, step by 2 records - one for each cal state
; in each sig_state
for nr=0L,nrec-1, 2 do begin
  ; cal off data (usually)
  getrec,nr
  copy,0,1
  a=getdata()
  ; cal on data (usually)
  getrec,nr+1
  copy,0,2
  b=getdata()
  ; frequency axis for data
  f=getxarray()

  ; now use gettp for the same data
  ; first need to determine scan number, ifnum, intnum, plnum,, sig_state
  myscan = !g.s[0].scan_number
  myif = !g.s[0].if_number
  myint = !g.s[0].integration
  mypol = !g.s[0].polarization_num
  mysigst = !g.s[0].sig_state
  ; now get the gettp data
  gettp,myscan,intnum=myint,ifnum=myif,plnum=mypol,sig_state=mysigst
  c=getdata()
  copy,0,3
```

```
;running kurtosis over kwidth*2 channels
k=c-c
m=c
nchan=n_elements(m)
for i=kwidth,nchan-1-kwidth do begin
  k[i]=kurtosis(c[i-kwidth:i+kwidth])
endfor

;badd is an array of the elements of the data that have kurtosis values > 2
badd = where(k gt  klimit,count)
bcount=count
; badf is an array of elements of the array of badd elements
; badf are flagged elements that should b e good - across the HI line
; set these values back to a kurtosis of zero
if bcount gt 0 then begin
  ; loop through the frequency list only remove if k less than kmax
  if numfreq gt 1 then begin
    for nf=0,numfreq-1,2 do begin
      badf = where(f[badd] ge myokfreqs[nf] and f[badd] le myokfreqs[nf+1] and k[badd] lt kmax,count)
      if count gt 0 then k[badd[badf]]=0.
    endfor
  endif
endif

;replace bad channels
; need to loop through twice - once for each record
for j=0,1 do begin
  copy,j+1,0  ; move the record to the container to flag
  bads=where(k gt klimit,count)
  if count gt 0 then begin
    good=-1
    notgood=-1
    nchan=count
    for n=0,nchan-1 do begin
      if good lt 0 then begin
        good=bads[n]-1
        notgood=bads[n]
```

```
          endif else begin
            if bads[n] ne notgood+1 or n eq nchan-1 then begin
              print,'replace',good,notgood+1
              replace,good,notgood+1
              good=-1
              notgood=-1
            endif else begin
              notgood=bads[n]
            endelse
          endelse
        endfor
      endif
      ; move spectrum back to data container
      copy,0,j+1
      ; save the data to the new file
      keep
    endfor

  endfor

  ; change files
  fileout,'foo.fits'
  filein,outputfile
end
```

APPENDIX B: THE REPLACERFIFREQ CODE

```
pro replacerfifreq,inputfile,outputfile,rfifreq,plusminus
;+
; NAME:
;     REPLACERFIFREQ
;
; PURPOSE:
;     Replace by interpolation any specified frequencies in frequency
;     swithed data.  Data will be replaced for all channels within
;     RFIFREQ-PLUSMINUS and RFIFREQ+PLUSMINUS
;
```

```
; CALLING SEQUENCE:
;     replacerfifreq, inputfile, outputfile, rfifreq, plusminus
;     [, kmax=kmax] [,myokfreqs=myokfreqs]
;
; INPUTS:
;     INPUTFILE - the raw data file with the frequency swicthed data
;     OUTPUTFILE - the data file to which the RFI corrected data will be written
;     RFIFREQ - The Topocentric frequency of the RFI in MHz.
;     PLUSMINUS - The half-width in MHz over which to replace the data.
;
; Examples:
;
; Typical use of fskurtosisflag for Galactic HI:
;      replacerfifreq,'baddata.fits','gooddata.fits',1426.6,0.01
;
;  Modification History
;      Written by Anthony Minter, GBO/NRAO 9 Jun 2025

 filein,inputfile
 fileout,outputfile
 nrec=nrecords()

 getrec,0
 setframe,'TOPO'
 setxunit,'MHz'
 setx,100,200
 sety,-1,1

 for i=0L,nrec-1 do begin
   getrec,i
   f=getxarray()
   bad=where(f ge rfifreq-plusminus and f lt rfifreq+plusminus, count)
   if count gt 0 then begin
     replace,min(bad),max(bad)
     print,FORMAT='("record ",I6," of ",I6,"   channels ",I6," to ",I6)',i,nrec-1,min(bad),max(bad)
   endif
   keep
 endfor
```

```
  fileout,'foo.fits'
  filein,outputfile
end
```