topo

Version 2.05, September 2000

David L. Windt

windt@astro.columbia.edu

The **topo** library is a set of IDL routines for analyzing 1D or 2D surface topography (i.e., surface roughness) data. I originally wrote these routines in order to analyze WYKO and AFM data of optical surfaces, but the routines are general, and can be used to analyze any type of 1D or 2D topography data.

The available routines are listed below. In particular, the routines `xanlz_prof` , `xanlz_psd` , and `xanlz_surf` are point-and-click (widget) applications for 1D profile analysis, 1D PSD analysis, and 2D surface analysis, respectively.

References

My understanding of surface roughness is based largely on the material contained in these two references:

1. J. M. Bennett and L. Mattsson, '*Introduction to Surface Roughness and Scattering*' (Optical Society of America, Washington, D. C. 1989)

2. E. L. Church, '*Fractal surface finish*', Applied Optics, **27** (8) 1518-1526 (1988)

Getting Started

See the README file in the topo directory for installation instructions.

Note: To use the topo library, you must be running IDL V5.1 or higher.

To load the **topo** library, at the IDL prompt type `.run topostart:`

```
IDL> .run topostart
```

Once the library has been loaded, you can use these routines as you would any IDL procedures and functions.

For instance, you might make up some 2D data like this:

```
IDL> x=vector(0.,100.,256)

IDL> y=x

IDL> z=randomn(seed,256,256)
```

and then try:

```
IDL> xanlz_surf,x,y,z
```

I've included in the `extras.dir` directory some sample data to play around with.

Try this:

```
IDL> rd_wyko,x,y,z,file='...path to topo.../extras.dir/wyko_example.dat'
```

NOTE: On Windows platforms, use the `SWAP_ENDIAN`keyword in the call to `RD_WYKO`when reading the '`wyko_example.dat`' file:

```
IDL> rd_wyko,x,y,z,file='...path to topo.../extras.dir/wyko_example.dat',/swap_endian
```

---

and then:

```
IDL> xanlz_surf,x,y,z
```

Or this:

```
IDL> rd_digital_afm,x,y,z,file='...path to topo.../extras.dir/digital_afm_example.dat'
```

NOTE: On NON-Windows platforms, use the SWAP_ENDIAN keyword in the call to `RD_DIGITAL_AFM` when reading the `digital_afm_example.dat` file:

```
IDL> rd_digital_afm,x,y,z,file='...path to topo.../extras.dir/digital_afm_example.dat',/swap_endian
```

---

and then:

```
IDL> xanlz_surf,x,y,z
```

List of Routines

Last Updated: 2-Sep-2000

New routines and those that have changed since the last release (2.04, 1-Dec-1998) are marked with an asterix.

- ANGLE2WAVES
- ANLZ_PROF
- AUTOCOV2PARS
- AUTOCOV2PSD
- AUTOCOV_FIT
- HEIGHT_DIST
- PARS2AUTOCOV
- PARS2PSD
- PLOT_AUTOCOV
- PLOT_H_DIST
- PLOT_PROF
- PLOT_PSD
- PLOT_SLOPE
- PLOT_S_DIST
- PROF2AUTOCOV
- * PROF2PSD
- PROF2SLOPE
- * PSD2RAVEPSD
- PSD2PARS
- PSDPARS2SIGMA_L
- PSDPARS_ANGLE2SIGMA_L
- PSD_FIT
- PSD_SIGMA_XI_H_FIT
- PSD_OMEGA_NU_N_FIT

Routine Descriptions

ANGLE2WAVES

[Next Routine] [List of Routines]

```
 NAME:

        ANGLE2WAVES

 PURPOSE:

        Procedure to compute (and optionally plot) the spatial
        wavelengths corresponding to scattering for incidence angle
        ALPHA, wavelength LAMBDA, into scattering angles
        ALPHA+/-DELTA(0) to ALPHA+/-DELTA(1)

 CATEGORY:

        Topographic analysis

 CALLING SEQUENCE:

        ANGLE2WAVES,ALPHA,LAMBDA,DELTA,WAVES=WAVES
```

INPUTS:

ALPHA - A scalar or 1D array of incidence angles.

LAMBDA - Scalar specifying the incidence wavelength.

DELTA - 2-element array specifying the min and max scattering angles to consider.

KEYWORD PARAMETERS:

WAVES - Output array of spatial wavelengths, having dimensions [2,N_ELEMENTS(ALPHA)]. Units are the same as for LAMBDA.

The following keywords only have an effect when ALPHA is an array:

NOPLOT - Set to inhibit plotting the results.

All IDL PLOT graphics keywords.

PROCEDURE:

```
WAVES=FLTARR(2,N_ELEMENTS(ALPHA))

WAVES(0,*)=LAMBDA/2./SIN(DELTA(0)/2.*!DTOR)/
           COS((ALPHA-DELTA(0)/2.)*!DTOR)

WAVES(1,*)=LAMBDA/2./SIN(DELTA(1)/2.*!DTOR)/
           COS((ALPHA-DELTA(1)/2.)*!DTOR)
```

EXAMPLE:

Compute the range of spatial wavelengths (in angstroms) corresponding to scattering for 30 degree incidence, with scattering angles from 1 to 60 degrees (i.e., 30+/-1 to 30+/-60), for a photon wavelength of 100 angstroms:

```
ANGLE2WAVES,30.,100.,[1.,60.]
```

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

ANLZ_PROF

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

       ANLZ_PROF

 PURPOSE:

       Perform 2D analysis of X,Y profile data.  This procedure
       combines the functionality of several TOPO routines, and
       displays the results in a single composite plot, using the
       PLOT_PROF, PLOT_H_DIST, PLOT_S_DIST, PLOT_AUTOCOV, and
       PLOT_PSD routines.

       This routines is called by the XANLZ_PROF widget application.

 CATEGORY:

       Topographic analysis

 CALLING SEQUENCE:

       ANLZ_PROF,X,Y

 INPUTS:

       X - 1D array of (equally spaced) lengths in angstroms.

       Y - 1D array of heights in angstroms.

 KEYWORD PARAMETERS:

       PLOT - A 4-element vector indicating which plots are to be
              made:

              PLOT(0) - Set to 1 to plot profile.

              PLOT(1) - Set to 1 to plot height and slope
                        distributions.
              PLOT(2) - Set to 1 to plot autocovariance.

              PLOT(3) - Set to 1 to plot PSD.

       XREGION - 1D array of subscripts for X, specifying a
                 region-of-interest.

       SUB_TILT - Set to subtract tilt, using SUB_TILT

       HEIGHTS - Output array of heights (see HEIGHT_DIST), in units
                 determined by the value of TOPO.YUNITS_PTR (see
                 PROCEDURE below.)

       HEIGHT_HIST - Output array of height histogram values (See
                     HEIGHT_DIST)

       N_HEIGHTS - Number of heights to be used for HEIGHT_DIST

       NOFIT_HEIGHT - Set to inhibit fitting the height histogram
                      with a gaussian.

       SLOPES - Output array of slopes in degrees (See HEIGHT_DIST.)

       SLOPE_HIST - Output array of slope histogram values (See
                    HEIGHT_DIST)

       N_SLOPES - Number of slopes to be used for HEIGHT_DIST

       NOFIT_SLOPE - Set to inhibit fitting the height histogram
                     with a gaussian.

       TAU - Output array of lag lengths, in units determined by the
             value of TOPO.XUNITS_PTR (see PROCEDURE below.)

       AUTOCOV - Output array of autocovariance values, in units
                 determined by the value of TOPO.YUNITS_PTR (see
                 PROCEDURE below.)  For example, if TOPO.YUNITS_PTR=1,
                 then the AUTOCOV units will be nm^2.

       FIT_AUTOCOV - Same as FIT_TYPE keyword to PLOT_AUTOCOV.

       FREQ - Output array of spatial frequencies, in units
              determined by the value of TOPO.XUNITS_PTR (see
              PROCEDURE below.)  For example, if TOPO.XUNITS_PTR=1,
              then the FREQ units will be 1/nm.
```

```
PSD - Output array of power-spectral-density values, in units
      determined by the value of TOPO.YUNITS_PTR (see
      PROCEDURE below.)  For example, if TOPO.YUNITS_PTR=1,
      then the PSD units will be nm^3.

ZERO_PAD - See PROF2PSD

HANNING - See PROF2PSD

KAISER - See PROF2PSD

RANGE - See PROF2PSD

FIT_PSD - Same as FIT_TYPE keyword to PLOT_PSD

Z_VALUE - Z parameter needed for Omega/nu/n PSD fits.

PRSTYLE - 5-element array specifying
          [COLOR,LINESTYLE,THICK,PSYM,SYMSIZE] keywords for
          profile plot. (except that PSYM and SYMSIZE are
          ignored!)

HSTYLE - 5-element array specifying
         [COLOR,LINESTYLE,THICK,PSYM,SYMSIZE] keywords for
         height-histogram plot. (except that PSYM and SYMSIZE
         are ignored!)

SSTYLE - 5-element array specifying
         [COLOR,LINESTYLE,THICK,PSYM,SYMSIZE] keywords for
         slope-histogram plot. (except that PSYM and SYMSIZE
         are ignored!)

ASTYLE - 5-element array specifying
         [COLOR,LINESTYLE,THICK,PSYM,SYMSIZE] keywords for
         autocovariance plot.

PSTYLE - 5-element array specifying
         [COLOR,LINESTYLE,THICK,PSYM,SYMSIZE] keywords for
         PSD plot.

PRXAXIS - 4-element array specifyin
          [XTYPE,XRANGE(0),XRANGE(1),XSTYLE] keywords for
          profile plot.

AXAXIS - 4-element array specifyin
         [XTYPE,XRANGE(0),XRANGE(1),XSTYLE] keywords for
```

autocovariance plot.

PXAXIS - 4-element array specifyin [XTYPE,XRANGE(0),XRANGE(1),XSTYLE] keywords for PSD plot.

PRYAXIS - 4-element array specifyin [YTYPE,YRANGE(0),YRANGE(1),YSTYLE] keywords for profile plot.

AYAXIS - 4-element array specifyin [YTYPE,YRANGE(0),YRANGE(1),YSTYLE] keywords for autocovariance plot.

PYAXIS - 4-element array specifyin [YTYPE,YRANGE(0),YRANGE(1),YSTYLE] keywords for PSD plot.

RESTRICTIONS:

The X values must be equally spaced.

X and Y must be in angstroms.

PROCEDURE:

Although X and Y must be in angstroms, the units for displayed (and returned keyword) variables are determined by the values of the relevant tags of the common block variable TOPO.

That is, set TOPO.XUNITS_PTR to 0 for angstroms, 1 for nm, 2 for microns, and 3 for mm. Same goes for TOPO.YUNITS_PTR.

The precision of the fit parameters that are labelled on the plots is determined by the values of the variables TOPO.XPRECISION for length-related parameters (e.g., correlation length), TOPO.YPRECISION for height-related parameters (e.g., rms roughness), and TOPO.SPRECISION for slope-related parameters (e.g., rms slope.) For example, if TOPO.XPRECISION=3, then three places to the right of the decimal will be displayed.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997

```
      January, 1998 - Added call to TEK_COLOR (unless !D.NAME eq
                      'HP'), and when !D.NAME eq 'CGM', set the color
                      index for !p.color to black.

      May, 1998 - Added PLOT keyword.

      windt@astro.columbia.edu
```

AUTOCOV2PARS

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

        AUTOCOV2PARS

 PURPOSE:

        Procedure to compute the rms roughness SIGMA and correlation
        length L from the the autocovariance function, C(TAU).

 CATEGORY:

        Topographic analysis

 CALLING SEQUENCE:

        AUTOCOV2PARS,TAU,C,SIGMA,L

 INPUTS:

        TAU - 1D array of lag lengths.

        C - 1D array of autocovariance values.

 OUTPUTS:

        SIGMA - Rms roughness, in units determined by the units of C.
                For example, if [C]=nm^2, then [SIGMA]=nm.

        L - Correlation length, in same units as TAU.
 PROCEDURE:

        SIGMA=SQRT(C)

        L = 2/SIGMA^4 * Integral(C^2)

 MODIFICATION HISTORY:

      David L. Windt, Bell Laboratories, May 1997
      windt@astro.columbia.edu
```

AUTOCOV2PSD

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

        AUTOCOV2PSD

 PURPOSE:

        Function to compute the power-spectral-density function S(F)
        from the the autocovariance function, C(TAU).

 CATEGORY:

        Topographic analysis

 CALLING SEQUENCE:

        S=AUTOCOV2PSD(TAUX[,TAUY],C)

 INPUTS:

        TAUX - 1D array of lag lengths along X direction.

        TAUY - Optional 1D array of lag lengths along Y direction.

        C - 1D or 2D array of autocovariance values.

 OUTPUTS:
```

S - 1D or 2D array of power-spectral-density values.

KEYWORD PARAMETERS:

FX - 1D output array of spatial frequencies along X direction.

FY - 1D output array of spatial frequencies along Y direction.

POSITIVE_ONLY - Set to only compute S(F) for positive spatial frequencies.

HANNING - Set to use a Hanning window function.

RANGE - 2-element array specifying the min and max spatial frequency values to be considered.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

AUTOCOV_FIT

[Previous Routine] [Next Routine] [List of Routines]

NAME:

AUTOCOV_FIT

PURPOSE:

Function to fit the autocovariance function with either a gaussian, and exponential, or a gaussian plus an exponential.

CATEGORY:

Topographic analysis

CALLING SEQUENCE:

```
        Result=AUTOCOV_FIT(TAU,C,PARS,FIT=FIT)

 INPUTS:

        TAU - 1D array of lag lengths.

        C - 1D array of autocovariance values.

        FIT - Set to 0 to fit to a gaussian.  Set to 1 to fit to an
              exponential. Set to 2 to fit to a gaussian+exponential.

 OUTPUTS:

       Result - Fit function.

       PARS - 1-D array of fit parameters.

 PROCEDURE:

       The autocovariance function is fit using either the GAUSS_FIT,
       the EXPO_FIT, or the GAUSSEXPO_FIT functions.  See the
       documentation for these functions (in the windt library) for a
       desciption of the elements of PARS.

 MODIFICATION HISTORY:

      David L. Windt, Bell Laboratories, May 1997
      windt@astro.columbia.edu
```

HEIGHT_DIST

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

        HEIGHT_DIST

 PURPOSE:

        Function to compute the height histogram from X,Y profile
        data.  The function can also be used to compute the slope
```

```
       histogram, in conjunction with the PROF2SLOPE function.

CATEGORY:

       Topographic analysis

CALLING SEQUENCE:

       Result=HEIGHT_DIST(X,Y[,HEIGHTS])

INPUTS:

       X - 1D array of (equally spaced) lengths.

       Y - 1D array of heights.

KEYWORD PARAMETERS:

       N_HTS - Integer specifying the number of bins to be used in
               conjunction with the HISTOGRAM function.

       RANGE - 2-element array specifying the min and max height
               values to consider.

OUTPUTS:

       Result - 1D array of density values. (See the HISTOGRAM
                function.)

OPTIONAL OUTPUTS:

       HEIGHTS - 1D array of height values; the i'th element of
                 Result is the density of heights in the i'th bin of
                 HEIGHTS.

PROCEDURE:

       The HISTOGRAM function is used.

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu
```

PARS2AUTOCOV

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

        PARS2AUTOCOV

 PURPOSE:

        Function to generate an autocovariance function from an array
        of parameters describing the function.  This function is a
        sort of inverse of the AUTOCOV_FIT routine.

 CATEGORY:

        Topographic analysis

 CALLING SEQUENCE:

        C=PARS2AUTOCOV(TAUX[,TAUY],P,TYPE=TYPE)

 INPUTS:

        TAUX - 1D array of lag lengths along X direction.

        TAUY - Optional 1D array of lag lengths along Y direction.

        P - Array of parameters describing the function.  The number
            of elements of P required depends on the value of the
            TYPE keyword:

        TYPE - Keyword specifying the functional form for C: 0 for
               gaussian, 1 for exponential, 2 for
               gaussian+exponential.

 OUTPUTS:

        C - 1D or 2D array of autocovariance values.

 PROCEDURE:

        TYPE=0 - Gaussian:

                  C=P(0)^2*EXP(-(TAU/P(1))^2)+P(2)
```

```
       TYPE=1 - Exponential:

                 C=P(0)^2*EXP(-ABS(TAU)/P(1))+P(2)

       TYPE=2 - Gaussian+Exponential:

                 C=P(0)^2*EXP(-(TAU/P(1))^2)+P(2)^2*EXP(-ABS(TAU)/P(3))+P(4)

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu
```

PARS2PSD

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       PARS2PSD

PURPOSE:

       Function to generate a power-law power-spectral-density
       function from an array of parameters describing the function.
       This function is a sort of inverse of the PSD_FIT routine.

CATEGORY:

       Topographic analysis

CALLING SEQUENCE:

       S=PARS2PSD(FX[,FY],P)

INPUTS:

       FX - 1D output array of spatial frequencies along X
            direction.

       FY - 1D output array of spatial frequencies along Y
```

direction.

P - 2-element array of parameters describing the power-law function. P(0)= K_n, P(1)=n.

OUTPUTS:

S - 1D or 2D array of power-spectral-density values.

PROCEDURE:

For a 1D PSD function,

```
S=P(0)/(ABS(F)^P(1))
```

For a 2D PSD function,

```
S=GAMMA((P(1)+1)/2)/2./GAMMA(.5)/GAMMA(P(1)/2.)* $
  P(0)/(ABS(F)^(P(1)+1))
```

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

PLOT_AUTOCOV

[Previous Routine] [Next Routine] [List of Routines]

NAME:

PLOT_AUTOCOV

PURPOSE:

Procedure to plot and optionally fit the autocovariance function.

CATEGORY:

Topographic analysis

```
CALLING SEQUENCE:

       PLOT_AUTOCOV,TAUX[,TAUY],C

INPUTS:

       TAUX - 1D array of lag lengths along X direction, in units
              corresponding to the value of TOPO.XUNITS_PTR.

       TAUY - 1D array of lag lengths along Y direction, in units
              corresponding to the value of TOPO.YUNITS_PTR.

       C - 1D array of autocovariance values, in units corresponding
           to the value of TOPO.YUNITS_PTR.

KEYWORD PARAMETERS:

       SIGMA - Output rms roughness, computed from C using
               AUTOCOV2PARS, in units corresponding to the value of
               TOPO.YUNITS_PTR.

       CORR_LENGTH - Correlation length, computed from C using
                     AUTOCOV2PARS, in units corresponding to the
                     value of TOPO.XUNITS_PTR.

       FIT_TYPE - Set to -1 for no fit, 0 for gaussian fit, 1 for
                  exponential fit, 2 for gaussian+exponential fit.

       REGION - Set to interactively select a region-of-interest for
                fitting, using GET_ROI.

       CFIT - Output fit dependent variable.

       TAUFIT - Output fit independent variable.

       FIT_PARS - Fit parameters.  (See AUTOCOV_FIT.)

       FIT_COLOR - IDL graphics keyword for fit function.

       FIT_THICK - IDL graphics keyword for fit function.

       FIT_LINESTYLE - IDL graphics keyword for fit function.

       NOLABEL - Set to inhibit labelling the SIGMA, L and fit
                 parameters values.

       LABEL_POSITION - Integer specifying label position, as per
```

PLOT_TEXT.

Plus most IDL PLOT graphics keywords.

RESTRICTIONS:

The units for TAU and C must correspond to the values of the relevant tags of the common block variable TOPO. That is, set TOPO.XUNITS_PTR to 0 for TAU in angstroms, 1 for nm, 2 for microns, and 3 for mm. Similarly, set TOPO.YUNITS_PTR to 0 for C in angstroms^2, 1 for nm^2, 2 for microns^2, and 3 for mm^2.

The precision of the fit parameters that are labelled on the plot is determined by the values of the variables TOPO.XPRECISION for length-related parameters (e.g., correlation length), and TOPO.YPRECISION for height-related parameters (e.g., rms roughness). For example, if TOPO.XPRECISION=3, then three places to the right of the decimal will be displayed.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

PLOT_H_DIST

[Previous Routine] [Next Routine] [List of Routines]

NAME:

PLOT_H_DIST

PURPOSE:

Procedure to compute, plot and optionally fit a height histogram function.

CATEGORY:

Topographic analysis

```
CALLING SEQUENCE:

       PLOT_H_DIST,X,Y

INPUTS:

       X - 1D array of lengths, in units corresponding to the value
           of TOPO.XUNITS_PTR.

       Y - 1D array of heights, in units corresponding to the value
           of TOPO.YUNITS_PTR.

KEYWORD PARAMETERS:

       N_HEIGHTS - Integer specifying the number of bins to be used
                   in conjunction with the HISTOGRAM function.

       HIST - 1D ouput array of density values. (See the HISTOGRAM
              function.)

       HEIGHTS - 1D output array of height values; the i'th element
                 of HIST is the density of heights in the i'th bin
                 of HEIGHTS.

       NOLABEL - Set to inhibit labelling the SIGMA, and L values.

       LABEL_POSITION - Integer specifying label position, as per
                        PLOT_TEXT.

       Plus most IDL PLOT graphics keywords.

RESTRICTIONS:

     The units for X and Y must correspond to the values of the
     relevant tags of the common block variable TOPO.  That is, set
     TOPO.XUNITS_PTR to 0 for X in angstroms, 1 for nm, 2 for
     microns, and 3 for mm.  Similarly, set TOPO.YUNITS_PTR to 0 for
     Y in angstroms, 1 for nm, 2 for microns, and 3 for mm.

     The precision of the fit parameters that are labelled on the
     plot is determined by the values of the variables
     TOPO.XPRECISION for length-related parameters (e.g.,
     correlation length), and TOPO.YPRECISION for height-related
     parameters (e.g., rms roughness).  For example, if
     TOPO.XPRECISION=3, then three places to the right of the
     decimal will be displayed.

PROCEDURE:

       HEIGHT_DIST is used to compute the height histogram function.

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu
```

PLOT_PROF

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       PLOT_PROF

PURPOSE:

       Procedure to plot a profile function.

CATEGORY:

       Topographic analysis

CALLING SEQUENCE:

       PLOT_PROF,X,Y

INPUTS:

       X - 1D array of lengths, in units corresponding to the value
           of TOPO.XUNITS_PTR.

       Y - 1D array of heights, in units corresponding to the value
           of TOPO.YUNITS_PTR.

KEYWORD PARAMETERS:

       OVERPLOT - Set for overplot.
```

NOLABEL - Set to inhibit labelling the SIGMA, and L values.

LABEL_POSITION - Integer specifying label position, as per PLOT_TEXT.

Plus most IDL PLOT graphics keywords.

RESTRICTIONS:

The units for X and Y must correspond to the values of the relevant tags of the common block variable TOPO. That is, set TOPO.XUNITS_PTR to 0 for X in angstroms, 1 for nm, 2 for microns, and 3 for mm. Similarly, set TOPO.YUNITS_PTR to 0 for Y in angstroms, 1 for nm, 2 for microns, and 3 for mm.

The precision of the parameters that are labelled on the plot is determined by the values of the variables TOPO.XPRECISION for length-related parameters (e.g., correlation length), and TOPO.YPRECISION for height-related parameters (e.g., rms roughness). For example, if TOPO.XPRECISION=3, then three places to the right of the decimal will be displayed.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

PLOT_PSD

[Previous Routine] [Next Routine] [List of Routines]

NAME:

PLOT_PSD

PURPOSE:

Procedure to plot and optionally fit the power-spectral-density function.

```
CATEGORY:

       Topographic analysis

CALLING SEQUENCE:

       PLOT_PSD,FX[,FY],S

INPUTS:

       FX - 1D array of spatial frequencies along X direction, in
            units corresponding to the value of TOPO.XUNITS_PTR.
            For example, if TOPO.XUNITS_PTR=1, then [FX]=1/nm.

       FY - 1D array of spatial frequencies along Y direction, in
            units corresponding to the value of TOPO.XUNITS_PTR.
            For example, if TOPO.XUNITS_PTR=1, then [FY]=1/nm.

       S - 1D array of power-spectral-density values, in units
           corresponding to the value of TOPO.YUNITS_PTR.  For
           example, if TOPO.YUNITS_PTR=1, then [S]=nm^3.

KEYWORD PARAMETERS:

       OVERPLOT - Set for overplot.

       SIGMA - Rms roughness, computed from S using PSD2PARS.

       CORR_LENGTH - Correlation length, computed from S using
                     PSD2PARS.

       FIT_TYPE - Set to -1 for no fit, 0 for power-law fit, 1 for
                  sigma/xi/h fit, and 2 for omega/nu/n fit.  If
                  FIT_TYPE=2, then Z_VALUE must be provided.  Note that
                  if FIT_TYPE is set to 0, then TWOD will be set to
                  zero as well; if FIT_TYPE is set to 1 or 2, then
                  TWOD will be set to 1.

       Z_VALUE - Film thickness parameter needed for omega/nu/n PSD
                 model fit.  Units for Z_VALUE are given by the
                 value of TOPO.XUNITS_PTR.

       RANGE - 2-element array of frequency values defining the
               region-of-interest for fitting.
       REGION - Set to interactively select a region-of-interest for
                fitting, using GET_ROI.


       TWOD - Set to indicate that the 1D PSD being plotted is
              actually a slice from a 2D PSD function, so that the
              units are labelled properly.  This keyword may be overwritten
              if FIT_TYPE is specified.

       SFIT - Output fit dependent variable.

       FFIT - Output fit independent variable.

       FIT_PARS - Fit parameters.  (See PSD_FIT.)

       FIT_COLOR - IDL graphics keyword for fit function.

       FIT_THICK - IDL graphics keyword for fit function.

       FIT_LINESTYLE - IDL graphics keyword for fit function.

       NOLABEL - Set to inhibit labelling the SIGMA, L and fit
                 parameters values.

       LABEL_POSITION - Integer specifying label position, as per
                        PLOT_TEXT.

       Plus most IDL PLOT graphics keywords.

RESTRICTIONS:

     The units for F and S must correspond to the values of the
     relevant tags of the common block variable TOPO.  That is, set
     TOPO.XUNITS_PTR to 0 for F in angstroms, 1 for nm, 2 for
     microns, and 3 for mm.  Similarly, set TOPO.YUNITS_PTR to 0 for
     S in angstroms^3, 1 for nm^3, 2 for microns^3, and 3 for mm^3.

     The precision of the fit parameters that are labelled on the
     plot is determined by the values of the variables
     TOPO.XPRECISION for length-related parameters (e.g.,
     correlation length), and TOPO.YPRECISION for height-related
     parameters (e.g., rms roughness).  For example, if
     TOPO.XPRECISION=3, then three places to the right of the
     decimal will be displayed.

MODIFICATION HISTORY:
     David L. Windt, Bell Laboratories, May 1997

     May 1998 - Added FIT_TYPE=1,2, and REGION and Z_VALUE keywords.

               - F and S are now correctly converted according the
                 values of TOPO.XUNITS_PTR and TOPO.YUNITS_PTR

     windt@astro.columbia.edu
```

PLOT_SLOPE

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

       PLOT_SLOPE

 PURPOSE:

       Procedure to plot a slope function.

 CATEGORY:

       Topographic analysis

 CALLING SEQUENCE:

       PLOT_SLOPE,X,SLOPE

 INPUTS:

       X - 1D array of lengths, in units corresponding to the value
           of TOPO.XUNITS_PTR.

       SLOPE - 1D array of slope values, in degrees.

 KEYWORD PARAMETERS:

       OVERPLOT - Set for overplot.

       NOLABEL - Set to inhibit labelling the SIGMA, and L values.
```

LABEL_POSITION - Integer specifying label position, as per PLOT_TEXT.

Plus most IDL PLOT graphics keywords.

RESTRICTIONS:

The units for X must correspond to the value of the relevant tag of the common block variable TOPO. That is, set TOPO.XUNITS_PTR to 0 for X in angstroms, 1 for nm, 2 for microns, and 3 for mm.

The precision of the rms slope parameter that is labelled on the plot is determined by the value of the variable TOPO.SPRECISION. For example, if TOPO.SPRECISION=3, then three places to the right of the decimal will be displayed.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

PLOT_S_DIST

[Previous Routine] [Next Routine] [List of Routines]

NAME:

PLOT_S_DIST

PURPOSE:

Procedure to compute, plot and optionally fit a slope histogram function.

CATEGORY:

Topographic analysis

CALLING SEQUENCE:

```
       PLOT_S_DIST,X,Y

INPUTS:

       X - 1D array of lengths, in units corresponding to the value
           of TOPO.XUNITS_PTR.

       Y - 1D array of heights, in units corresponding to the value
           of TOPO.YUNITS_PTR.

KEYWORD PARAMETERS:

       N_SLOPES - Integer specifying the number of bins to be used
                  in conjunction with the HISTOGRAM function.

       HIST - 1D ouput array of density values. (See the HISTOGRAM
              function.)

       SLOPES - 1D output array of slope values; the i'th element of
                HIST is the density of slopes in the i'th bin of
                SLOPES.

       NOLABEL - Set to inhibit labelling the SIGMA, and L values.

       LABEL_POSITION - Integer specifying label position, as per
                        PLOT_TEXT.

       Plus most IDL PLOT graphics keywords.

RESTRICTIONS:

     The units for X and Y must correspond to the values of the
     relevant tags of the common block variable TOPO.  That is, set
     TOPO.XUNITS_PTR to 0 for X in angstroms, 1 for nm, 2 for
     microns, and 3 for mm.  Similarly, set TOPO.YUNITS_PTR to 0 for
     Y in angstroms, 1 for nm, 2 for microns, and 3 for mm.

     The precision of the fit parameter (i.e., rms slope) that is
     labelled on the plot is determined by the value of the variable
     TOPO.SPRECISION.  For example, if TOPO.SPRECISION=3, then three
     places to the right of the decimal will be displayed.

PROCEDURE:

     PROF2SLOPE is used to compute the slope values; HEIGHT_DIST
     is used to compute the slope histogram function.
```

```
 MODIFICATION HISTORY:

      David L. Windt, Bell Laboratories, May 1997
      windt@astro.columbia.edu
```

PROF2AUTOCOV

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

       PROF2AUTOCOV

 PURPOSE:

       Function to compute the autocovariance function from the
       profile data.

 CATEGORY:

       Topographic analysis

 CALLING SEQUENCE:

       C=PROF2AUTOCOV(X,Y,TAU=TAU)

 INPUTS:

       X - 1D array of (equally-spaced) lengths.

       Y - 1D array of heights.

 OUTPUTS:

       TAU - 1D array of lag lengths, in units of [X].

       C - 1D array of autocovariance values, in units of [Y]^2.

 KEYWORD PARAMETERS:

       POSITIVE_ONLY - Set to compute the autocovariance function
```

for positive lag lengths only.

RESTRICTIONS:

The X values must be equally spaced.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

PROF2PSD

[Previous Routine] [Next Routine] [List of Routines]

NAME:

PROF2PSD

PURPOSE:

Function to compute the power-spectral-density function from the profile data.

CATEGORY:

Topographic analysis

CALLING SEQUENCE:

`S=PROF2PSD(X,Y,F=F)`

INPUTS:

X - 1D array of (equally-spaced) lengths.

Y - 1D array of heights.

OUTPUTS:

F - 1D array of spatial frequencies, in units of 1/[X].

```
       S - 1D array of PSD values, in units of [Y]^3.

 KEYWORD PARAMETERS:

       POSITIVE_ONLY - Set to compute the psd function for positive
                       frequencies only.

       RANGE - 2-element array specifying the min and max spatial
               frequencies to be considered. Default is from
               1/(length) to 1/(2*interval) (i.e., the Nyquist
               frequency), where length is the length of the scan,
               and interval is the spacing between points.

       ZERO_PAD - Set this to an integer specifying the number of
                  zero-height points to add on either side of the
                  profile data.

       HANNING - Set this to use a Hanning window function.

       KAISER - Set this to use a Kaiser-Bessel window function

 RESTRICTIONS:

       The X values must be equally spaced.

 PROCEDURE

      S=Length*ABS(FFT(Y*Window),-1)^2

      Where Length is as described above, and Window is the value of
      the optional window function (Hanning or Kaiser-Bessel).

 MODIFICATION HISTORY:

      David L. Windt, Bell Laboratories, May 1997
      windt@astro.columbia.edu

      Nov 1998: When using the HANNING or KAISER keywords, the window
      function is now normalized so that the integral of the PSD is
      ~constant, i.e., independent of your choice of window.

      September 2000:  Corrected a problem wherein the spatial
      frequencies were computed incorrectly.
```

PROF2SLOPE

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

       PROF2SLOPE

 PURPOSE:

       Function to compute the slope from the profile data.

 CATEGORY:

       Topographic analysis

 CALLING SEQUENCE:

       Result=PROF2SLOPE(X,Y)

 INPUTS:

       X - 1D array of (equally-spaced) lengths.

       Y - 1D array of heights.

 OUTPUTS:

       Result - 1D array of slope values, in degrees.

 RESTRICTIONS:

      The X values must be equally spaced.

      X and Y must have the same units.

 MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu
```

PSD2RAVEPSD

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

        PSD2RAVEPSD

 PURPOSE:

        Function to compute the radially-averaged PSD from a 2D PSD array.

 CATEGORY:

        Topographic analysis

 CALLING SEQUENCE:

        SR=PSD2RAVEPSD(FX,FY,S,FR=FR)

 INPUTS:

        FX - 1D array of spatial frequencies along X
             direction.

        FY - Optional 1D array of spatial frequencies along Y
             direction.

        S - 2D array of power-spectral-density values.

 OUTPUTS:

        SR - 1D array of radially-averaged PSD values.

        FR - 1D array of spatial frequencies.

 PROCEDURE:

        The radially-averaged PSD is computed for a vector of FR values,
        with FR = sqrt(FX^2+FY^2), and for FR values ranging from the
        minimum spatial frequency to the maximum spatial frequency
        along X or Y (whichever is smallest.)

        The i'th value of SR, at a specific FR(i), is equal to the
        average of all S values contained in the anulus defined by
```

```
       FR(i)-.5*DFR < sqrt(FX^2+FY^2) <= FR(i)+.5*DFR, where DFR
       is the spatial frequency increment.

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1998
     windt@astro.columbia.edu

     September 2000: Corrected some problems with the way SR and FR
     were computed that arose when FX and FY are unequal.
     Eric Gullikson, LBL
     EMGullikson@lbl.gov
```

PSD2PARS

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       PSD2PARS

PURPOSE:

       Procedure to compute the rms roughness SIGMA and correlation
       length L from the the power-spectral-density function S(F).

CATEGORY:

       Topographic analysis

CALLING SEQUENCE:

       PSD2PARS,F,S,SIGMA,L

INPUTS:

       F - 1D array of spatial frequencies.

       S - 1D array of power-spectral-density values.

OUTPUTS:
```

SIGMA - Rms roughness, in units of [S]^(1/3)

L - Correlation length, in units of 1/[F]

PROCEDURE:

SIGMA=SQRT(2*Integral(S))

L = 1/2/SIGMA^4 * Integral(S^2)

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997

windt@astro.columbia.edu

PSDPARS2SIGMA_L

[Previous Routine] [Next Routine] [List of Routines]

NAME:

PSDPARS2SIGMA_L

PURPOSE:

Procedure to compute the rms roughness SIGMA and correlation length L from the scan LENGTH and the power-law parameters K_N and N.

CATEGORY:

Topographic analysis

CALLING SEQUENCE:

PSDPARS2SIGMA_L,K_N,N,LENGTH,SIGMA,L

INPUTS:

K_N - Power-law scaling factor.

N - Power-law exponent.

LENGTH - Scan length.

OUTPUTS:

SIGMA - Rms roughness.

L - Correlation length.

PROCEDURE:

`SIGMA = SQRT(K_n*LENGTH^(N-1)/(N-1))`

`L = (N-1)^2*LENGTH/2/(2*N-1)`

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

PSDPARS_ANGLE2SIGMA_L

[Previous Routine] [Next Routine] [List of Routines]

NAME:

PSDPARS_ANGLE2SIGMA_L

PURPOSE:

Function to compute the rms roughness SIGMA and correlation length L from (a) the maximum spatial wavelength corresponding to scattering for incidence angle ALPHA, wavelength LAMBDA, into scattering angles ALPHA+/-DELTA(0) to ALPHA+/-DELTA(1), and (b) the power-law parameters K_N and N.

CATEGORY:

Topographic analysis

CALLING SEQUENCE:

```
       PSDPARS_ANGLE2SIGMA_L,K_N,N,ALPHA,LAMBDA,DELTA,SIGMA,L

INPUTS:

       K_N - Power-law scaling factor.

       N - Power-law exponent.

       ALPHA - A scalar or 1D array of incidence angles.

       LAMBDA - Scalar specifying the incidence wavelength.

       DELTA - 2-element array specifying the min and max scattering
               angles to consider.

OUTPUTS:

       SIGMA - Rms roughness.

       L - Correlation length.

PROCEDURE:

       The ANGLE2WAVES procedure is used to compute the range of
       spatial wavelengths corresponding to scattering at incidence
       angle ALPHA, wavelength LAMBDA, into scattering angles
       ALPHA+/-DELTA(i).  The maximum wavelength is then used along
       with the specified K_N and N to compute SIGMA and L, using
       the PSDPARS2SIGMA_L routine.

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu
```

PSD_FIT

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:
```

PSD_FIT

PURPOSE:

Function to fit a 1D array of power-spectral-density function values with a power-law.

CATEGORY:

Topographic analysis

CALLING SEQUENCE:

```
Result=PSD_FIT(F,S,PARS)
```

INPUTS:

F - 1D array of spatial frequencies.

S - 1D array of power-spectral-density values.

OUTPUTS:

Result - Fit function.

PARS - 1-D array of fit parameters: PARS(0)=K_n, PARS(1)=N

KEYWORD PARAMETERS:

RANGE - 2-element array of min and max spatial frequencies to be used for fitting. Default is to use the entire range of data.

FFIT - Array of F values corresponding to the range used for fitting.

TWOD - If TWOD is set, then the 1D array of PSD values is consider to represent a slice (or radial average) of a 2D PSD function.

PROCEDURE:

A 1D PSD function is fit to the function

```
S=K_n/(ABS(F)^N)
```

A 2D PSD function is fit to the function

```
          S=Gamma((N+1)/2)/(2*Gamma(1/2)*Gamma(N/2))*K_n/F^(N+1)

 MODIFICATION HISTORY:

      David L. Windt, Bell Laboratories, May 1997

      May 1998 - Added RANGE, FFIT and TWOD keywords.

               - If a 2-sided PSD is provided, then fit function
                 returned is now only 1-sided (i.e., defined only for
                 positive frequencies).

      windt@astro.columbia.edu
```

PSD_SIGMA_XI_H_FIT

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

      PSD_SIGMA_XI_H_FIT

 PURPOSE:

        Function to fit a power-spectral-density function with a
        function of the form:

        PSD(2*pi*F)=4 pi H sigma^2 xi^2 /(1 + abs(2*pi*F)^2 xi^2)^(1+H)

        Note that this form refers to a 2D PSD function; thus it only
        makes sense to use this program to fit a 1D array of
        radially-averaged PSD values.

 CATEGORY:

        Topographic analysis

 CALLING SEQUENCE:

        Result=PSD_SIGMA_XI_H_FIT(F,S,PARS)

 INPUTS:
        F - 1D array of spatial frequencies.

        S - 1D array of power-spectral-density values.

 OUTPUTS:

       Result - Fit function.

       PARS - 1-D array of fit parameters: [sigma,xi,H]

 KEYWORD PARAMETERS:

       RANGE - 2-element array of min and max spatial frequencies to
               be used for fitting.  Default is to use the entire
               range of data.

       FFIT - Array of F values corresponding to the range used for
              fitting.

 RESTRICTIONS:

      F and S must use the same length units.

 MODIFICATION HISTORY:

      David L. Windt, Bell Laboratories, May 1998
      windt@astro.columbia.edu
```

PSD_OMEGA_NU_N_FIT

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

     PSD_OMEGA_NU_N_FIT

PURPOSE:

       Function to fit a power-spectral-density function with a
       function of the form:
```

```
       PSD(2*pi*F)=Omega * (1-exp(-2*nu*(2*pi*F)^n*z)) / (2*nu*(2*pi*F)^n)

       Note that this form refers to a 2D PSD function; thus it only
       makes sense to use this program to fit a 1D array of
       radially-averaged PSD values.

       This form of the PSD comes from the stochastic model of thin
       film growth and erosion, developed by D. Stearns,
       Appl. Phys. Lett. 62, 1745-1747 (1993)

CATEGORY:

       Topographic analysis

CALLING SEQUENCE:

       Result=PSD_OMEGA_NU_N_FIT(F,S,Z,PARS)

INPUTS:

       F - 1D array of spatial frequencies.

       S - 1D array of power-spectral-density values.

       Z - z parameter (film thickness)

OUTPUTS:

      Result - Fit function.

      PARS - 1-D array of fit parameters: [Omega, nu, n]

KEYWORD PARAMETERS:

      RANGE - 2-element array of min and max spatial frequencies to
              be used for fitting.  Default is to use the entire
              range of data.

      FFIT - Array of F values corresponding to the range used for
             fitting.

COMMON BLOCKS

     COMMON PSD_OMEGA_NU_N,Z

RESTRICTIONS:
     F and S must use the same length units.

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1998
     windt@astro.columbia.edu
```

RD_DIGITAL_AFM

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

      RD_DIGITAL_AFM

PURPOSE:

      Read a Digital Instruments Nanoscope III AFM image.

CALLING SEQUENCE:

       RD_DIGITAL_AFM,X,Y,Z

KEYWORD PARAMETERS:

      FILE -  name of Nanoscope III data file.

      TILT_SUB - set to subtract tilt.

      SPHERE_SUB - set to subtract a 2nd order polynomial
                   background.

      HIST_EQUAL - set to perform histogram equalization of image.

      SWAP_ENDIAN - set to use the SWAP_ENDIAN function to convert
                    the data file from 'big endian' to 'little
                    endian', or visa-versa. Useful for reading on a
                    Windows platform data files originally
                    stored on Unix or Mac platforms, or visa-versa.

OUTPUTS:
      X - Vector of x-position values, in angstroms.

      Y - Vector of y-position values, in angstroms.

      Z - Array of height values, in angstroms.

MODIFICATION HISTORY:

   David L. Windt, Bell Labs, 1992.
   windt@astro.columbia.edu

   Actually working: February 1998

   November 1998: The procedure can now handle Nanoscope III
                  file version 0x0440003.
```

RD_GRIFFITH_AFM

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

      RD_GRIFFITH_AFM

 PURPOSE:

      Read a Joe Griffith AFM file.

 CALLING SEQUENCE:

      RD_GRIFFITH_AFM,X,Y,Z

 KEYWORD PARAMETERS:

      FILE - name of AFM data file, without the .arr ending.  It is
             assumed that both the .arr and .hed files exist.

      TILT_SUB - set to subtract tilt.

      SPHERE_SUB - set to subtract a 2nd order polynomial background.
```

```
     HIST_EQUAL - set to perform histogram equalization.

OUTPUTS:

     X - Vector of x-position values, in angstroms.

     Y - Vector of y-position values, in angstroms.

     Z - Array of height values, in angstroms.

MODIFICATION HISTORY:

  David L. Windt, Bell Labs, 1991.
  windt@astro.columbia.edu
```

RD_MICROMAP

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       RD_MICROMAP

PURPOSE:

      Read a Micromap data file.

CALLING SEQUENCE:

      RD_MICROMAP,FILE=FILE,X,Y,Z

KEYWORD PARAMETERS:

      FILE -  name of Micromap data file.

      TILT_SUB - set to subtract tilt.

      SPHERE_SUB - set to subtract a 2nd order polynomial background.

      HIST_EQUAL - set to perform histogram equalization of image.

OUTPUTS:
      X - Vector of x-position values, in angstroms.

      Y - Vector of y-position values, in angstroms.

      Z - Array of height values, in angstroms.

MODIFICATION HISTORY:

  David L. Windt, Bell Laboratories, May 1998

  windt@astro.columbia.edu
```

RD_WYKO

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       RD_WYKO

PURPOSE:

      Read a WYKO TOPO-3D data file.

CALLING SEQUENCE:

      RD_WYKO,FILE=FILE,X,Y,Z,HEADER=HEADER

KEYWORD PARAMETERS:

      FILE -  name of WYKO data file.

      OFFSET - set OFFSET to skip an additional 512 bytes at the
               beginning of the data file.  This additional 512-byte
               header is introduced by BASIC/UX when the file is
               transferred from an LIF to an HFS disk.

      TILT_SUB - set to subtract tilt.

      SPHERE_SUB - set to subtract a 2nd order polynomial background.

      HIST_EQUAL - set to perform histogram equalization of image.
      SWAP_ENDIAN - set to use the SWAP_ENDIAN function to convert
                    the data file from 'big endian' to 'little
                    endian', or visa-versa. Useful for reading on a
                    Windows platform WYKO data files originally
                    stored on Unix or Mac platforms, or visa-versa.

OUTPUTS:

      X - Vector of x-position values, in angstroms.

      Y - Vector of y-position values, in angstroms.

      Z - Array of height values, in angstroms.

OPTIONAL OUTPUT PARAMETERS:

      HEADER - structure value containing the raw header information.

MODIFICATION HISTORY:

  David L. Windt, Bell Laboratories, March 1991.

  August, 1997 - Added SWAP_ENDIAN keyword.

  May, 1998 - Added SUB_TILT keyword

  windt@astro.columbia.edu
```

RD_ZYGO

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       RD_ZYGO

PURPOSE:

      Read a ZYGO NewView data file.

CALLING SEQUENCE:
      RD_ZYGO,FILE=FILE,X,Y,Z

KEYWORD PARAMETERS:

      FILE -  name of ZYGO data file.

      TILT_SUB - set to subtract tilt.

      SPHERE_SUB - set to subtract a 2nd order polynomial background.

      HIST_EQUAL - set to perform histogram equalization of image.

OUTPUTS:

      X - Vector of x-position values, in angstroms.

      Y - Vector of y-position values, in angstroms.

      Z - Array of height values, in angstroms.

MODIFICATION HISTORY:

  David L. Windt, Bell Laboratories, June 1998

  windt@astro.columbia.edu
```

SUB_TILT

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       SUB_TILT

PURPOSE:

       Function to subtract tilt (i.e., a straight line) from
       profile data.

CATEGORY:

       Topographic analysis
```

```
CALLING SEQUENCE:

       Result=SUB_TILT(X,Y,CURVATURE=CURVATURE)

INPUTS:

       X - 1D array of lengths.

       Y - 1D array of heights.

KEYWORD PARAMETERS:

       CURVATURE - Set to subtract a 2nd order polynomial instead of
                   a straight line.

OUTPUTS:

       Result - The new profile data.

EXAMPLE:

      NewY=SUB_TILT(X,Y)

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu
```

SURF2AVE_PSD

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       SURF2AVE_PSD

PURPOSE:

       Procedure to compute the average power-spectral-density
       functions along X and Y from surface data.
```

CATEGORY:

Topographic analysis

CALLING SEQUENCE:

```
SURF2AVE_PSD,X,Y,Z,XAVEPSD=XAVEPSD,YAVEPSD=YAVEPSD,
                   FX=FX,FY=FY,XPSD=XPSD,YPSD=YPSD
```

INPUTS:

X - 1D array of lengths along X direction.

Y - 1D array of lengths along Y direction.

Z - 2D array of heights.

OUTPUTS:

FX - 1D array of spatial frequencies along X direction, in units of 1/[X].

FY - 1D array of spatial frequencies along Y direction, in units of 1/[Y].

XAVEPSD - 1D array of PSD values = average of XPSD along Y direction, in units of [Z]^3.

YAVEPSD - 1D array of PSD values = average of YPSD along X direction, in units of [Z]^3.

XPSD - 2D array of PSD values computed along X direction, in units of [Z]^3.

YPSD - 2D array of PSD values computed along Y direction, in units of [Z]^3.

KEYWORD PARAMETERS:

POSITIVE_ONLY - Set to compute the autocovariance function for positive lag lengths only.

RANGE - 2-element array specifying the min and max spatial frequencies to be considered. Default is from 1/(length) to 1/(2*interval) (i.e., the Nyquist frequency), where length is the length of the scan, and interval is the spacing between points.

```
       ZERO_PAD - Set this to an integer specifying the number of
                  zero-height points to add on either side of the
                  profile data.

       HANNING - Set this to use a Hanning window function.

       KAISER - Set this to use a Kaiser-Bessel window function

 PROCEDURE:

    This function computes the 1D PSD functions along every line in
    the X and Y directions of the image array, and then computes the
    averages in X and Y of these 1D PSD functions.  Note that SUB_TILT
    is used on each line prior to computing the PSD.

 MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu

     September 2000:  Corrected a problem that caused SURF2AVE_PSD to fail
     if X and/or Y contained an odd number of points.
```

SURF2PSD

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

       SURF2PSD

 PURPOSE:

       Function to compute the 2D power-spectral-density function
       from surface data.

 CATEGORY:

       Topographic analysis

 CALLING SEQUENCE:
       S=SURF2PSD(X,Y,Z,FX=FX,FY=FY)

INPUTS:

       X - 1D array of lengths along X direction.

       Y - 1D array of lengths along Y direction.

       Z - 2D array of heights.

OUTPUTS:

       Result - 2D PSD function, in units of [Z]^4.

       FX - 1D array of spatial frequencies along X direction, in
            units of 1/[X].

       FY - 1D array of spatial frequencies along Y direction, in
            units of 1/[Y].

KEYWORD PARAMETERS:

       POSITIVE_ONLY - Set to compute the autocovariance function
                       for positive lag lengths only.

       XRANGE - 2-element array specifying the min and max spatial
                frequencies along X direction to be
                considered. Default is from 1/(Xlength) to
                1/(2*interval) (i.e., the Nyquist frequency), where
                Xlength is the length of the scan along X, and
                interval is the spacing between points.

       YRANGE - 2-element array specifying the min and max spatial
                frequencies along Y direction to be
                considered. Default is from 1/(Ylength) to
                1/(2*interval) (i.e., the Nyquist frequency), where
                Ylength is the length of the scan alogn Y, and
                interval is the spacing between points.

       ZERO_PAD - Set this to an integer specifying the number of
                  zero-height points to add on either side of the
                  profile data.

       HANNING - Set this to use a Hanning window function.
```

KAISER - Set this to use a Kaiser-Bessel window function

PROCEDURE:

`S=XLength*YLength*ABS(FFT(Z*Window),-1)^2`

Where XLength and YLength are as described above, and Window is the value of the optional window function (Hanning or Kaiser-Bessel).

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997
windt@astro.columbia.edu

Nov 1998: When using the HANNING or KAISER keywords, the window function is now normalized so that the integral of the PSD is ~constant, i.e., independent of your choice of window.

September 2000: Corrected a problem wherein the spatial frequencies were computed incorrectly. Also, non-isotropic surfaces are now handled correctly. Much thanks to Eric Gullikson, LBL, EMGullikson@lbl.gov

---

TOPOSTART

[Previous Routine] [Next Routine] [List of Routines]

NAME:

TOPOSTART

PURPOSE:

The start procedure for the TOPO surface topography library.

CATEGORY:

Topographic analysis

CALLING SEQUENCE:

```
       At the IDL command line, type ".run TOPOSTART".

COMMON BLOCKS:

       COMMON TOPO,TOPOHOME,TOPO
       COMMON PLOT_PRINT,PRINTPARS (See PLOT_PRINT)

PROCEDURE:

       The site-configuration file (topositeconfig.pro) is read,
       defining widget fonts, etc.  The topo save file is loaded
       into IDL, using the RESTORE command.

RESTRICTIONS:

       UNITS and PRECISION:

       When using any of the topo plot procedures: PLOT_PROF,
       PLOT_SLOPE, PLOT_H_DIST, PLOT_S_DIST, PLOT_AUTOCOV, and
       PLOT_PSD; or the procedures ANLZ_PROF, XANLZ_PROF, XANLZ_PSD, and
       XANLZ_SURF, the data length (X for 1D, or X and Y for 2D) and
       height (Y for 1D, or Z for 2D) units MUST ALL BE IN
       ANGSTROMS.  However, the common block variable TOPO is used
       to control the units that are actually displayed.

       In particular, the variables TOPO.XUNITS_PTR and
       TOPO.YUNITS_PTR determine the units for lengths and heights,
       respectively; set TOPO.XUNITS_PTR to 0 for angstroms, 1 for
       nm, 2 for microns, or 3 for mm.  Same goes for
       TOPO.YUNITS_PTR.

       The precision of the fit parameters that are labelled on the
       plots is determined by the values of the variables
       TOPO.XPRECISION for length-related parameters (e.g.,
       correlation length), TOPO.YPRECISION for height-related
       parameters (e.g., rms roughness), and TOPO.SPRECISION for
       slope-related parameters (e.g., rms slope.)  For example, if
       TOPO.XPRECISION=3, then three places to the right of the
       decimal will be displayed.

       The TOPO.*UNITS_PTR and TOPO.*PRECISION variables can either
       be set explicitly before executing any of the aforementioned
       procedures, or they can be set transparently to the user be
       selecting the appropriate menu items when using the
       XANLZ_PROF, XANLZ_PSD, or XANLZ_SURF procedure.
```

```
 MODIFICATION HISTORY:

      David L. Windt, Bell Laboratories, May 1997
      windt@astro.columbia.edu
```

TOPO_X_CONVERT

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

        TOPO_X_CONVERT

 PURPOSE:

        Function to convert length variables according to the value
        of the common block variable TOPO.XUNITS_PTR.


 CATEGORY:

        Topographic analysis

 CALLING SEQUENCE:

        NewX=TOPO_X_CONVERT(X)

 INPUTS:

        X - 1D array of length values, in angstroms.

 OUTPUTS:

        Result - 1D array of length values, in units specified by
                 TOPO.XUNITS_PTR, i.e., 1 => angstroms, 2 => nm, 3 =>
                 microns, and 4 => mm.

 KEYWORD PARAMETERS:

        TO_ANGSTROMS - Set to convert an input X from the units
                       specified by TOPO.XUNITS_PTR to angstroms.
```

```
COMMON BLOCKS:

       COMMON TOPO

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu
```

TOPO_Y_CONVERT

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       TOPO_Y_CONVERT

PURPOSE:

       Function to convert height variables according to the value
       of the common block variable TOPO.YUNITS_PTR.


CATEGORY:

       Topographic analysis

CALLING SEQUENCE:

       NewY=TOPO_Y_CONVERT(Y)

INPUTS:

       Y - 1D array of height values, in angstroms.

OUTPUTS:

       Result - 1D array of height values, in units specified by
                TOPO.YUNITS_PTR, i.e., 1 => angstroms, 2 => nm, 3 =>
                microns, and 4 => mm.
```

```
KEYWORD PARAMETERS:

       TO_ANGSTROMS - Set to convert an input Y from the units
                      specified by TOPO.YUNITS_PTR to angstroms.

COMMON BLOCKS:

       COMMON TOPO

MODIFICATION HISTORY:

     David L. Windt, Bell Laboratories, May 1997
     windt@astro.columbia.edu
```

XANLZ_PROF

[Previous Routine] [Next Routine] [List of Routines]

```
NAME:

       XANLZ_PROF

PURPOSE:

       A widget application for 1D profile analysis.  This program
       is really a widget interface to the ANLZ_PROF routine.

CATEGORY:

       Topographic analysis

CALLING SEQUENCE:

       XANLZ_PROF[,X,Y,GROUP=GROUP]

OPTIONAL INPUTS:

       X - 1D array of (equally-spaced) lengths, in angstroms.

       Y - 1D array of heights, in angstroms.

KEYWORD PARAMETERS:
```

GROUP - Widget GROUP_LEADER keyword to XMANAGER.

COMMON BLOCKS:

COMMON TOPO
COMMON PLOT_PRINT

RESTRICTIONS:

The X values must be equally spaced.

X and Y must be in angstroms.

PROCEDURE:

Although X and Y must be in angstroms, the units for displayed variables are determined by the values of the relevant tags of the common block variable TOPO.

That is, set TOPO.XUNITS_PTR to 0 for angstroms, 1 for nm, 2 for microns, and 3 for mm. Same goes for TOPO.YUNITS_PTR.

The precision of the fit parameters that are labelled on the plots is determined by the values of the variables TOPO.XPRECISION for length-related parameters (e.g., correlation length), TOPO.YPRECISION for height-related parameters (e.g., rms roughness), and TOPO.SPRECISION for slope-related parameters (e.g., rms slope.) For example, if TOPO.XPRECISION=3, then three places to the right of the decimal will be displayed.

The TOPO.*UNITS_PTR and TOPO.*PRECISION variables can either be set explicitly before executing this procedure, or they can be set transparently to the user be selecting the appropriate menu items once the procedure is running.

If no data are passed, the user is prompted to enter the IDL command string used to read in new data. This command string *must explicitly define X and Y in angstroms*. For example, if your data is in the form of a plain text (ASCII) file, consisting of two columns of data, X and Y, then you can use the EROM routine, as in

```
EROM,X,Y,FILE='MyProfile.dat'
```

If your data is not in this form then you will probably need to write your own IDL procedure to read in the data, with X and Y as explicit parameters, i.e.,

```
MY_PROCEDURE,X,Y
```

or

```
Result=MY_FUNCTION(X,Y)
```

Once the profile data are defined, this routine uses the ANLZ_PROF routine to compute the height distribution, the slope distribution, the autocovariance, and the power-spectral-density, with optional fitting, and displays the results in a composite plot that can be printed using PLOT_PRINT. Various popup-widgets are used to adjust the computations, fits and plots to the users preference.

No common blocks are used that prevent multiple instances of this routine from being used.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997

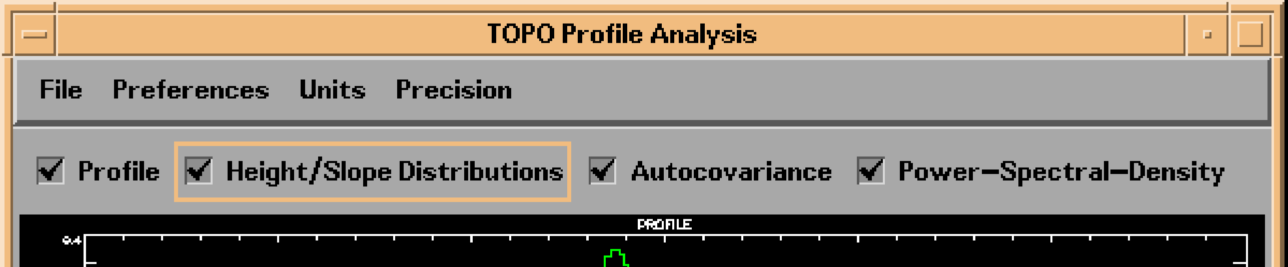
May 1998 - Added ability to specify range of F values used for PSD fitting.

- Added ability to specify which plots are made, as in the new PLOT keyword to ANLZ_PROF.

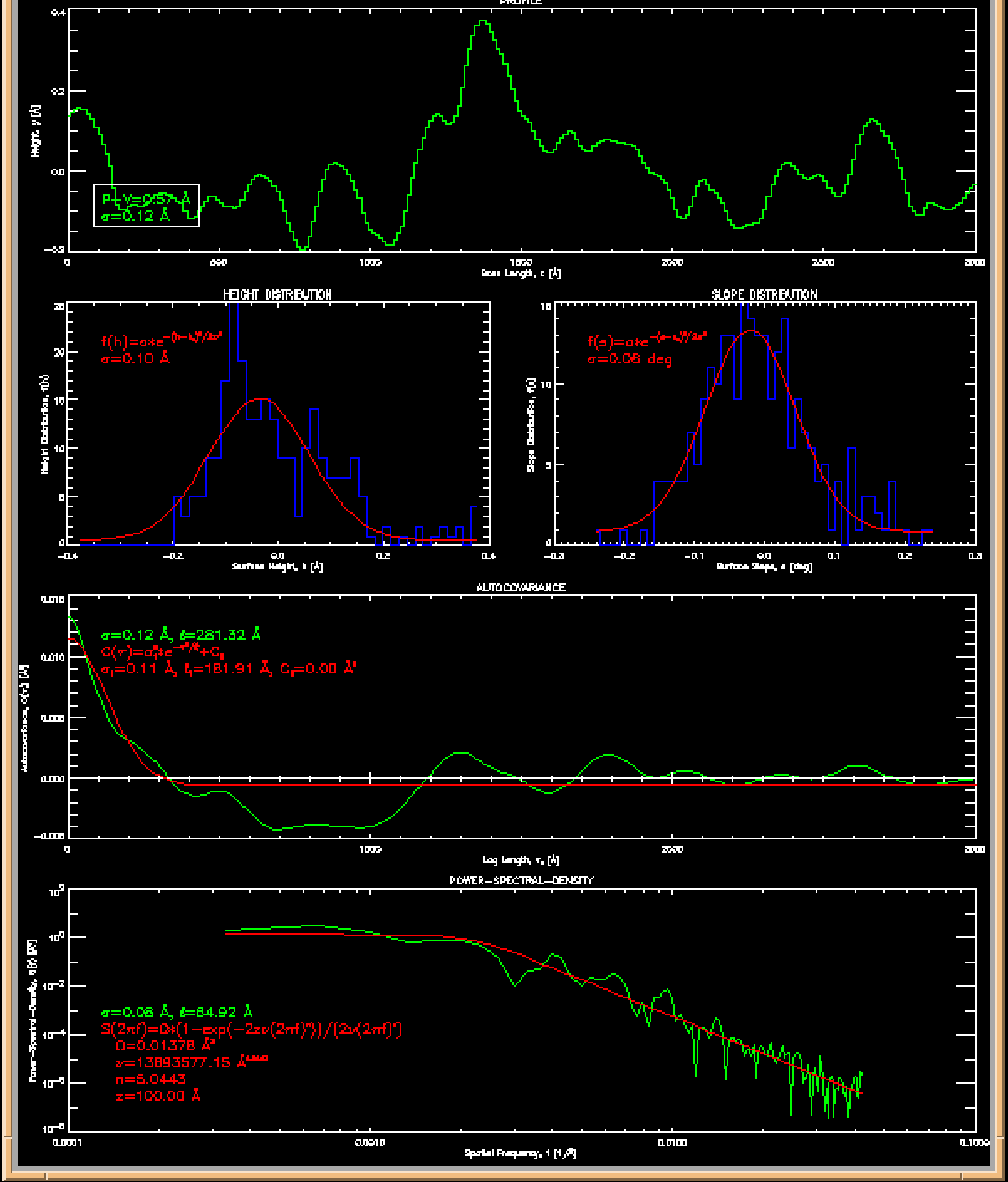
- Draw widget now gets resized when the base is resized.

windt@astro.columbia.edu

ILLUSTRATION:



☑ Profile ☑ Height/Slope Distributions ☑ Autocovariance ☑ Power-Spectral-Density


PROFILE
Height, y (Å)
0.4
0.2
0.0
−0.2
P−V=0.57 Å
σ=0.12 Å
0
500
1000
1500
2000
2500
3000
Scan Length, x (Å)
HEIGHT DISTRIBUTION
f(h)=a∗e^−(h−h₀)²/2σ²
σ=0.10 Å
Height Distribution, f(h)
25
20
15
10
5
0
−0.4
−0.2
0.0
0.2
0.4
Surface Height, h (Å)
SLOPE DISTRIBUTION
f(s)=a∗e^−(s−s₀)²/2σ²
σ=0.06 deg
Slope Distribution, f(s)
15
10
5
0
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Surface Slope, s (deg)
AUTOCOVARIANCE
σ=0.12 Å, ℓ=281.32 Å
C(τ)=σ₁²∗e^−τ/ℓ₁+C₀
σ₁=0.11 Å, ℓ₁=181.91 Å, C₀=0.00 Å²
Autocovariance, C(τ) (Å²)
0.015
0.010
0.005
0.000
−0.005
0
1000
2000
3000
Lag Length, τ (Å)
POWER−SPECTRAL−DENSITY
Power−Spectral−Density, S(f) (Å³)
10²
10⁰
10⁻²
10⁻⁴
10⁻⁶
10⁻⁸
σ=0.06 Å, ℓ=84.92 Å
S(2πf)=Ω∗(1−exp(−2zν(2πf)ⁿ))/(2ν(2πf)ⁿ)
Ω=0.01378 Å³
ν=13893577.15 Å
n=5.0443
z=100.00 Å
0.0001
0.0010
0.0100
0.1000
Spatial Frequency, f (1/Å)

XANLZ_PSD

[Previous Routine] [Next Routine] [List of Routines]

```
 NAME:

       XANLZ_PSD

 PURPOSE:

       A widget application for 1D PSD analysis.  This program
       is essentially a widget interface to the PLOT_PSD routine.

 CATEGORY:

       Topographic analysis

 CALLING SEQUENCE:

       XANLZ_PSD[,F,S,GROUP=GROUP]

 OPTIONAL INPUTS:

       F - 1D array of spatial frequencies, in 1/angstroms.

       S - 1D array of power-spectral-density values, in A^3.

 KEYWORD PARAMETERS:

       TWOD - Set to indicate that the F and S values correspond to
              a 1D slice of a 2D PSD function, or a radially-averaged
              PSD function.


       GROUP - Widget GROUP_LEADER keyword to XMANAGER.

       TITLE - Title for main widget.

 COMMON BLOCKS:

       COMMON TOPO
       COMMON PLOT_PRINT

 PROCEDURE:
```

Although F and S must be in angstroms, the units for displayed variables are determined by the values of the relevant tags of the common block variable TOPO.

That is, set TOPO.XUNITS_PTR to 0 for angstroms, 1 for nm, 2 for microns, and 3 for mm. Same goes for TOPO.YUNITS_PTR.

The precision of the fit parameters that are labelled on the plots is determined by the values of the variables TOPO.XPRECISION for length-related parameters (e.g., correlation length), TOPO.YPRECISION for height-related parameters (e.g., rms roughness), and TOPO.SPRECISION for slope-related parameters (e.g., rms slope.) For example, if TOPO.XPRECISION=3, then three places to the right of the decimal will be displayed.

The TOPO.*UNITS_PTR and TOPO.*PRECISION variables can either be set explicitly before executing this procedure, or they can be set transparently to the user be selecting the appropriate menu items once the procedure is running.

If no data are passed, the user is prompted to enter the IDL command string used to read in new data. This command string *must explicitly define F and S in angstroms*. For example, if your data is in the form of a plain text (ASCII) file, consisting of two columns of data, F and S, then you can use the EROM routine, as in

```
EROM,F,S,FILE='MyProfile.dat'
```

If your data is not in this form then you will probably need to write your own IDL procedure to read in the data, with F and S as explicit parameters, i.e.,

```
MY_PROCEDURE,F,S
```

or

```
Result=MY_FUNCTION(F,S)
```

Once the profile data are defined, this routine uses the PLOT_PSD routine to display the results along with an optional fit.

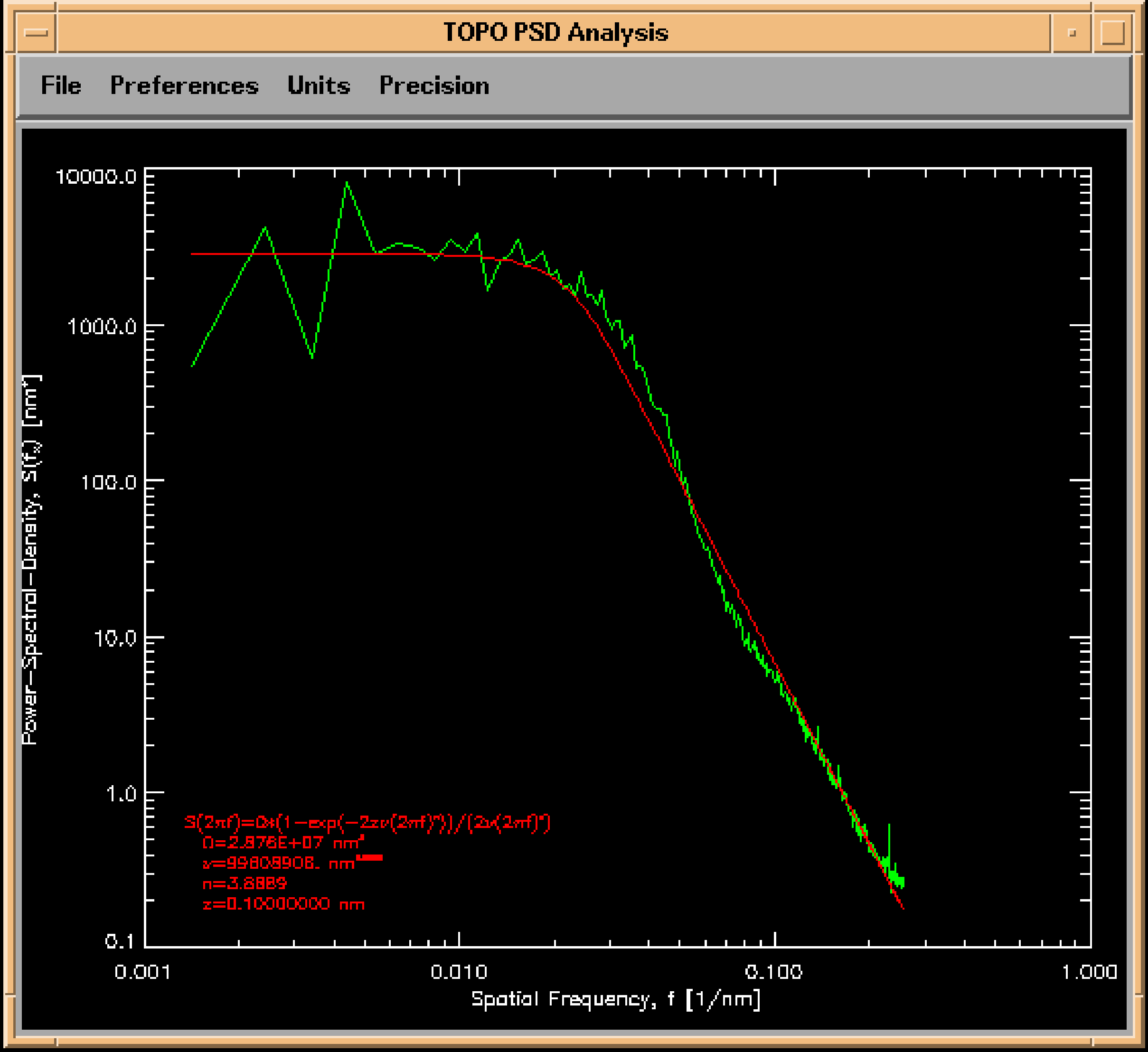
No common blocks are used that prevent multiple instances of this routine from being used.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1998

windt@astro.columbia.edu

ILLUSTRATION:



XANLZ_SURF

[Previous Routine] [List of Routines]

```
 NAME:

        XANLZ_SURF

 PURPOSE:

        A widget application for 2D surface analysis.

 CATEGORY:

        Topographic analysis

 CALLING SEQUENCE:

        XANLZ_SURF[,X,Y,Z]

 OPTIONAL INPUTS:

        X - 1D array of (equally-spaced) lengths along X direction,
            in angstroms.

        Y - 1D array of (equally-spaced) lengths along Y direction,
            in angstroms.

        Z - 2D array of heights, in angstroms.

        WINDOW_XSIZE - The horizontal size of the window for image display, in
pixels.
                       Default is 380.
        WINDOW_YSIZE - The vertical size of the window for image display, in pixels.
                       Default is 400.

 COMMON BLOCKS:

        COMMON TOPO
        COMMON PLOT_PRINT
        COMMON XANLZ_SURF

 RESTRICTIONS:

        The X and Y values must be equally spaced.
```

X, Y and Z must be in angstroms.

PROCEDURE:

Although X, Y and Z must be in angstroms, the units for displayed variables are determined by the values of the relevant tags of the common block variable TOPO.

That is, set TOPO.XUNITS_PTR to 0 for X and Y angstroms, 1 for nm, 2 for microns, and 3 for mm. Similarly, set TOPO.YUNITS_PTR to 0 for Z in angstroms, 1 for nm, 2 for microns, and 3 for mm.

The precision of the fit parameters that are labelled on the plots is determined by the values of the variables TOPO.XPRECISION for length-related parameters (e.g., correlation length), TOPO.YPRECISION for height-related parameters (e.g., rms roughness), and TOPO.SPRECISION for slope-related parameters (e.g., rms slope.) For example, if TOPO.XPRECISION=3, then three places to the right of the decimal will be displayed.

The TOPO.*UNITS_PTR and TOPO.*PRECISION variables can either be set explicitly before executing this procedure, or they can be set transparently to the user be selecting the appropriate menu items once the procedure is running.

If no data are passed, the user is prompted to enter the IDL command string used to read in new data. This command string *must explicitly define X and Y*. For example, if you're reading in WYKO TOPO-3D data, you can use the RD_WYKO procedure, as in

```
RD_WYKO,X,Y,Z,FILE='MyProfile.dat'
```

You might also try the RD_DIGITAL_AFM procedure for reading in data obtained with the Digital Instruments Nanoscope III atomic force microscope.

If you cannot use RD_WYKO or RD_DIGITAL_AFM, then you will need to write your own IDL procedure to read in the data, with X, Y and Z as explicit parameters, i.e.,

```
MY_PROCEDURE,X,Y,Z
```

or

`Result=MY_FUNCTION(X,Y,Z)`

(The TOPO distribution includes some sample WYKO data, in a file called 'wyko_example.dat'.)

Once the surface data are defined, the user can select interactively 1D profiles along X or Y. These profiles are plotted, as are the corresponding PSD functions. The user can also select X or Y profiles for more detailed analysis using the XANLZ_PROF routine, which can be called directly from XANLZ_SURF.

MODIFICATION HISTORY:

David L. Windt, Bell Laboratories, May 1997

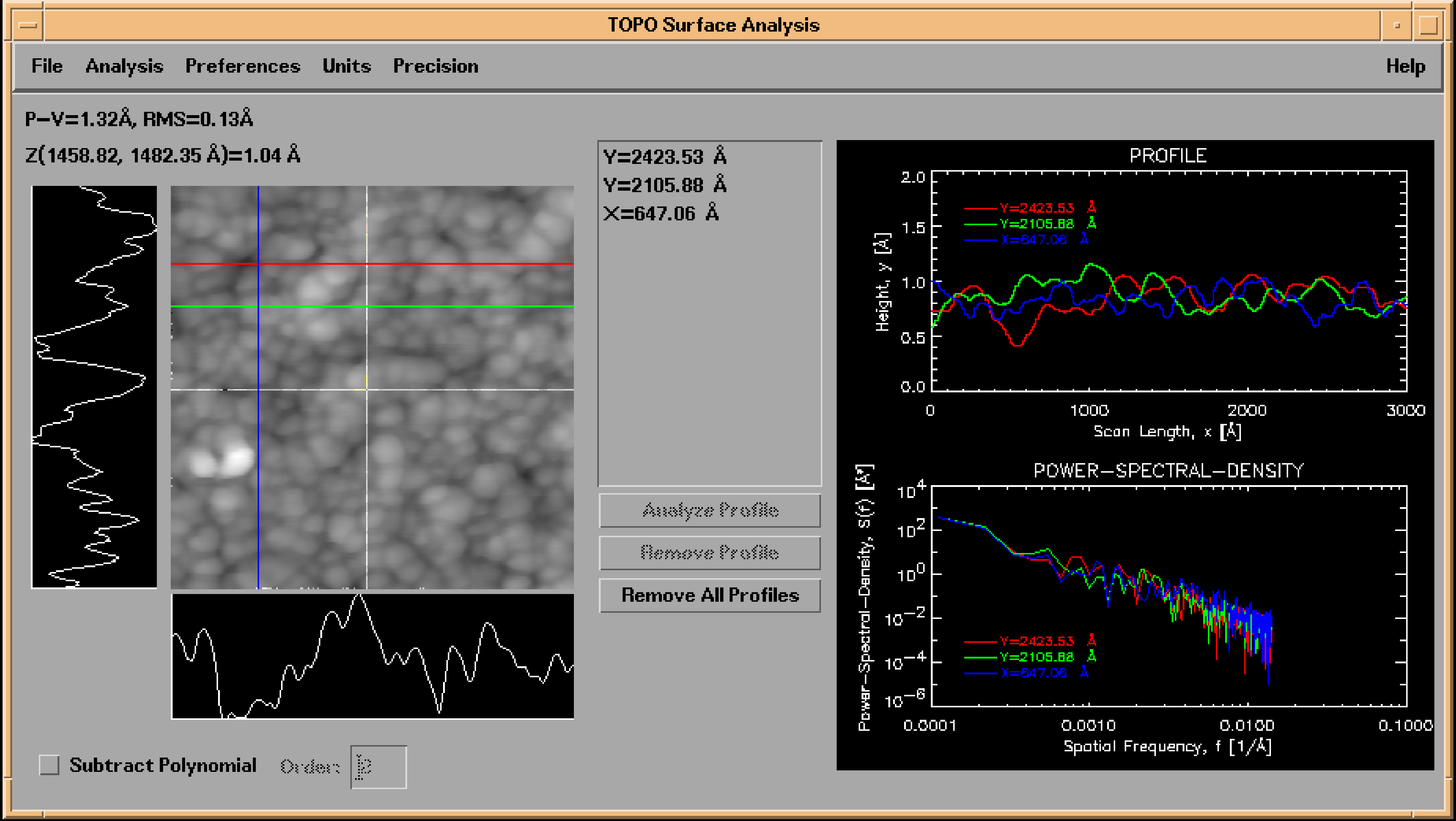
January, 1998 - When !D.NAME eq 'CGM', set the color index for !p.color to black.

May, 1998 - Added analysis menu, with option of using XANLZ_PSD on X-, Y- and radially-averaged PSD function.

Added WINDOW_XSIZE and WINDOW_YSIZE keywords.

windt@astro.columbia.edu

ILLUSTRATION:



P–V=1.32Å, RMS=0.13Å

Z(1458.82, 1482.35 Å)=1.04 Å

Y=2423.53 Å
Y=2105.88 Å
X=647.06 Å

Order: 2

TOPO

Version 2.05, August 2000

David L. Windt

windt@astro.columbia.edu
<http://cletus.phys.columbia.edu/windt/idl>

This file contains instructions for downloading, unpacking and installing TOPO.

Contents of this file:

1. System requirements
2. Downloading TOPO
3. Unpacking TOPO
4. Installing TOPO
5. Starting TOPO
6. Getting help
7. Using TOPO as a XOP extension

---

1. SYSTEM REQUIREMENTS

TOPO is written in the IDL language, and will run on any platform supported by IDL. Little or no IDL expertise is required to use TOPO.

You will need a valid, licensed copy of IDL to run TOPO. IDL version 5.1 or higher is required. IDL is available from Research Systems, Inc, <http://www.rsinc.com>

> NOTE: It is possible to run TOPO as an 'extension' to the XOP program (with limited functionality) which is available for Unix and Windows platforms, free of charge to research institutes and non-profit organizations. information on XOP can be found at <http://www.esrf.fr/computing/scientific/xop/>. Please see section 7 of this document for further instructions.

2. DOWNLOADING TOPO

There is only one TOPO distribution, and it should run on any platform supported by IDL. For your convenience, however, I have created several different download files, making it easier for you to unpack TOPO on your particular platform. Choose whichever one you like.

Download files:

UNIX: topo.tar.Z

Windows: topo.zip

MacOS: topo.hqx

(Note that you may need to explicitly configure your browser to save the download file directly to disk.)

3. UNPACKING TOPO

To unpack the file you downloaded, follow the instructions below for your specific platform. When you're finished, you'll end up with a directory called "topo", which will consume approximately 5 MB of disk space.

Unix

o Unpack the topo.tar.Z file, using the Unix uncompress and tar commands:

```
$ uncompress -c topo.tar.Z | tar -xvf -
```

Windows

o Use the (32-bit) WinZip shareware program to unpack the topo.zip file.

MacOS

o Use the Stuffit Expander freeware program to unpack the topo.hqx file.

4. INSTALLING TOPO

The "topo" directory should nominally be installed in a directory that you must create called "user_contrib", inside the main IDL directory. That is, you should end up with a directory called "~idl/user_contrib/topo" (using Unix "notation".) In any case make sure that the topo directory is included in your IDL path; for help with setting your IDL path please consult the IDL documentation.

The main TOPO program is distributed as an IDL SAVE file, and all the required IDL procedures and functions are already loaded, so you will not need to include any other directories in your IDL path. However, the topo library makes use of many of the general-purpose routines in the windt library. Although you do not need the windt library to use the topo library (as the needed routines are already compiled into the IDL SAVE file), you may wish to download the windt library nonetheless, in order to have access to the documentation for these routines.

If you choose NOT to install the "topo" directory inside of the "user_contrib" directory in the main IDL directory, then you MUST edit the file "topositeconfig.pro", located in the "topo" directory. The file "topositeconfig.pro" contains executable IDL code (although the program itself will not run if you just type ".run topositeconfig.pro" at the IDL prompt.) You must change the value of the 'topohome' variable in this file so that it refers to the actual TOPO installation directory.

You can also specify default widget fonts (in "topositeconfig.pro") to be used by IDL, as per your preference: set the value of the 'font_w' variable to the name of a valid IDL widget font for your site. In addition, you can tell TOPO how to display the angstrom symbol and the greek "mu" symbol for the font you specify.

5. STARTING TOPO

To load the TOPO library, at the IDL prompt type:

```
IDL> .run topostart
```

You can then try one of the three widget programs, XANLZ_PROF for 1D profile analysis, XANLZ_PSD for 1D PSD analysis, or XANLZ_SURF 2D surface analysis.

For instance, you might make up some 2D data like this:

```
IDL> x=vector(0.,100.,256)
IDL> y=x
IDL> z=randomn(seed,256,256)
```

and then try:

```
IDL> xanlz_surf,x,y,z
```

I've included in the extras.dir directory some sample data to play around with:

Try this:

```
IDL> rd_wyko,x,y,z, $
     file='...path to topo.../extras.dir/wyko_example.dat'
```

> NOTE: On Windows platforms, use the SWAP_ENDIAN keyword in the call to RD_WYKO when reading the 'wyko_example.dat' file:
>
> ```
> IDL> rd_wyko,x,y,z, $
>      file='...path to topo.../extras.dir/wyko_example.dat',/swap_endian
> ```

and then:

```
IDL> xanlz_surf,x,y,z
```

Or this:

```
IDL> rd_digital_afm,x,y,z, $
     file='...path to topo.../extras.dir/digital_afm_example.dat'
```

> NOTE: On NON-Windows platforms, use the SWAP_ENDIAN keyword in the call to RD_DIGITAL_AFM when reading the digital_afm_example.dat file:
>
> ```
> IDL> rd_digital_afm,x,y,z, $
>      file='...path to topo.../extras.dir/digital_afm_example.dat',/swap_endian
> ```

and then:

```
IDL> xanlz_surf,x,y,z
```

---

6. GETTING HELP

You can read all about the individual TOPO routines if you need to go beyond what XANLZ_PROF, XANLZ_PSD and XANLZ_SURF provide. These routines are documented in the html file "topo.html", contained in the "topo" directory.

Please let me know about any problems, at <windt@astro.columbia.edu>.

---

7. USING TOPO AS A XOP EXTENSION

(If you will use TOPO with a licensed copy of IDL, you can skip the following instructions.)

It is possible to run TOPO as an extension to the XOP program, but with only very limited functionality. That is, the TOPO routines are intended to be used in your own IDL programs, but it is not possible to use your own IDL programs with XOP (in embedded mode, at least.) Thus, you can run the XANLZ_PROF, XANLZ_PSD, and XANLZ_SURF TOPO applications, but that's pretty much it.

A. To install TOPO as a XOP extension, perform the following three steps:

i. Download and install the XOP program, as described at the XOP website: <http://www.esrf.fr/computing/scientific/xop/>.

ii. Download and unpack the TOPO distribution, following the instructions given above in section 2 and 3. Install the topo directory in the XOP extensions directory. When you complete this step, you should end up with a directory called .../xop/extensions/topo.

iii. Copy and rename the appropriate topo4xop.sav_V file - where V=51 for the version of XOP built from IDL version 5.1, V=52 for the version of XOP built from IDL version 5.2, and V=53 for the version of XOP built from IDL version 5.3 - from the topo directory to the XOP extensions directory. When you complete this step, you should end up with a file called .../xop/extensions/topo4xop.sav. That is:

for XOP / IDL V5.1:

```
copy .../extensions/topo/extras.dir/topo4xop.sav_51
to .../extensions/topo4xop.sav
```

for XOP / IDL V5.2:

```
copy .../extensions/topo/extras.dir/topo4xop.sav_52
to .../extensions/topo4xop.sav
```

for XOP / IDL V5.3:

```
copy .../extensions/topo/extras.dir/topo4xop.sav_53
to .../extensions/topo4xop.sav
```

B. Starting TOPO as a XOP extension:

There are two (platform-specific) ways to run TOPO as a XOP extension:

For Unix platforms:

At the unix system prompt, type:

```
xop topo4xop
```

-or-

From the main XOP window:

a. Select Tools -> IDL Macro

b. Clear the window and type topo4xop

c. Select File -> Quit and accept changes

For Windows platforms:

Assuming your XOP installation directory is c:\xop, double-click on the file:

```
        c:\xop\extensions\topo\extras.dir\topo4xop.bat

        Note: If your XOP installation is different from c:\xop, you will
        need to edit the topo4xop.bat file accordingly.

        -or-

        From the main XOP window:

        a. Select Tools -> IDL Macro

        b. Clear the window and type topo4xop

        c. Select File -> Quit and accept changes
```

NOTE: If you have trouble installing IMD as a XOP extension, *PLEASE CONTACT THE XOP DEVELOPERS DIRECTLY.*

C. Using topo4xop

The topo4xop program is a very simple interface to the TOPO library, consisting of buttons to access the three widget programs mentioned above, and a text entry area where you can type TOPO and IDL commands.

```
---------------------------------------------------------------------------
---------------------------------------------------------------------------
```