



PV-WAVE Function Catalog

Overview**1**

- PV-WAVE 1
- PV-WAVE Optional Toolkits 2
- Database Connection for Oracle, Sybase and ODBC 2
- GTGRID 2
- TS-WAVE for Powerful Time Series Analysis 3

PV-WAVE Foundation**4**

- Arrays 4
- Graphics and Plotting 7
- GUI Development 12
- Image Processing and Color 19
- Input/Output 22
- Internet Enabling 24
- Mathematics 26
- Programming 29
- Rendering Techniques 35

PV-WAVE ADVANTAGE

PV-WAVE: IMSL Mathematics**38**

- Linear Systems 38
- Eigensystem Analysis 39
- Interpolation and Approximation 39
- Quadrature 40
- Differential Equations 41
- Transforms 41
- Nonlinear Equations 41
- Optimization 41
- Special Functions 42
- Basic Statistics and Random Number Generation 45
- Probability Distribution Functions and Inverses 46
- Utilities 46

PV-WAVE: IMSL Statistics**47**

- Basic Statistics 47
- Regression 47
- Correlation and Covariance 48
- Analysis of Variance 49
- Categorical and Discrete Data Analysis 49
- Nonparametric Statistics 50
- Goodness of Fit 50
- Time Series and Forecasting 50
- Multivariate Analysis 51
- Survival Analysis 52
- Probability Distribution Functions and INverses 52
- Random Number Generation 53
- Utilities 54
- Neural Networks 54

PV-WAVE EXTREME ADVANTAGE

PV-WAVE: Signal Processing Toolkit**55**

-
- Signals and Systems 55
 - Filter Approximation 55
 - Filter Realization 56
 - Transforms and Spectrum Analysis 56
 - Statistical Signal Processing 57
 - Polynomial Manipulation 57
 - Optimization 58
 - Plotting and Signal Generation 58

PV-WAVE: Image Processing Toolkit**59**

-
- Point Operations 59
 - Filtering 59
 - Morphological Image Processing 60
 - Mensuration 61
 - Representation and Description 61
 - Image Transforms 62
 - Geometric Transforms 62
 - Color Image Processing 63
 - Color Quantization 63
 - Classification and Segmentation 63
 - Utilities 64

Database Connection for Oracle and Sybase**65****Database Connection for ODBC****66****GTGRID****67**

Visual Data Analysis You Can Trust

The PV-WAVE® family of products, which includes PV-WAVE, JWAVE™ and TS-WAVE™, is a suite of development solutions that allow users to rapidly import, manipulate, analyze and visualize data of any size and complexity. Unlike other products, PV-WAVE also includes a sophisticated set of analysis routines based upon the IMSL™ Numerical Libraries. These routines have been the industry standard for over 30 years and provide users with unmatched numerical analysis capability for scientific, engineering and business analytics.

In addition, Web-based JWAVE provides access to visual analysis applications from anywhere at anytime, and allows users to rapidly analyze, visualize and share critical information across the enterprise. For more specific applications, TS-WAVE is a highly extensible application that delivers time-series data analysis.

With this set of comprehensive visual data analysis (VDA) solutions, PV-WAVE has earned its place as the choice among experts.

PV-WAVE

At its core, PV-WAVE Foundation is an array based programming language used to build and deploy VDA applications. These applications let users manipulate and visualize simple to extremely complex datasets to detect and display patterns, trends, anomalies and other vital information that would otherwise be overlooked in their data.

PV-WAVE delivers powerful image and signal processing, data import and export, 3D surface, plot, histogram, contour, animation, color editor and a database table display. This rapid data analysis and visualization combined with flexibility and power make PV-WAVE the leader among the experts.

PV-WAVE Advantage includes JWAVE and a sophisticated set of analysis routines based on the industry-standard IMSL Numerical Libraries for reliable and precise numerical analysis. JWAVE™ is the PV-WAVE® Web Development Environment for network-based visual data analysis applications. With JWAVE's multiple graphics and visualization techniques, users can rapidly analyze, visualize and share critical information across the organization, regardless of the computer platform. Using the JWAVE Server users can create a centralized repository for updating and maintaining applications. The JWAVE Server handles very large data sets and allows storage of all application logic on the server, where it belongs.

To solve more specific problems PV-WAVE Extreme Advantage provides toolkits for image processing and signal processing.

- The PV-WAVE Image Processing Toolkit includes an extensive set of filters, transforms and image processing operators designed to meet the needs of even the most demanding image processing applications.
- The PV-WAVE Signal Processing Toolkit provides a broad spectrum of pre-defined and readily customized Digital Signal Processing (DSP) functions, which are designed to improve data analysis and simulations. Filters are transfer-function based for easy design, analysis and realization of custom filters. The toolkit includes function and procedures for the Fourier and wavelet analysis of a signal, as well as utilities and source code to facilitate custom function development.

PV-WAVE Optional Toolkits

Database Connection for Oracle, Sybase and ODBC

PV-WAVE's powerful database connection functions let you import data from an external database into PV-WAVE. If you can access your data source from the computer on which PV-WAVE is running, you can connect to the data source from within PV-WAVE. This eliminates the need to export the data to a file before importing it into PV-WAVE. You can query or update your data source from PV-WAVE using the same Structured Query Language (SQL) commands supported by your local DBMS. You do not need to learn new syntax or complicated function and procedure calls.

GTGRID

PV-WAVE: GTGRID™ gives PV-WAVE users additional interpolation and extrapolation power by providing access to a library of gridding routines prepared by Geophysical Techniques (GT), Inc.

The GTGRID Library is a structured collection of routines for solving surface estimation (gridding) problems. The GTGRID library provides several unique gridding methods, each designed to treat a certain class of data in an optimum manner. It is especially valuable for PV-WAVE users in such areas as:

- Oil and Gas Industry
- Aerospace
- Radar and Sonic Applications
- Weather and Atmospheric Conditions

TS-WAVE for Powerful Time Series Analysis

For time series data analysis, TS-WAVE is a highly extensible application that allows plot creation, and batch processing of your data. The TS-WAVE framework consists of four general functional areas within a single application: time history, tabular data, x-y plotting and batch processing.

ARRAYS

ARRAY CREATION ROUTINES

ASARR – Creates an associative array containing specified variables and expressions.

BINDGEN – Returns a byte array with the specified dimensions, setting the contents of the result to increasing numbers starting at 0.

BYTARR – Returns a byte vector or array.

CINDGEN – Returns a complex single-precision floating-point array.

COMPLEXARR – Returns a complex single-precision floating-point vector or array.

DBLARR – Returns a double-precision floating-point vector or array.

DCINDGEN – Returns a double-precision floating-point complex array.

DCOMPLEXARR – Returns a double-precision floating-point complex vector or array.

DIAG – Makes a diagonal array or extracts the diagonal of an array.

DINDGEN – Returns a double-precision floating-point array with the specified dimensions.

FINDGEN – Returns a single-precision floating-point array with the specified dimensions.

FLTARR – Returns a single-precision floating-point vector or array.

I32ARR – Returns a 32-bit integer vector or array.

I32INDGEN – Returns a 32-bit integer array with the specified dimensions.

INDGEN – Returns an integer array with the specified dimensions.

INTARR – Returns an integer vector or array.

LINDGEN – Returns a longword integer array with the specified dimensions.

LIST – Creates a list array.

LISTARR – Returns a list.

LONARR – Returns a longword integer vector or array.

MAKE_ARRAY – Returns an array of specified type, dimensions, and initialization. It provides the ability to create an array dynamically whose characteristics are not known until run time.

REPLICATE – Forms an array with the given dimensions, filled with the specified scalar value.

SINDGEN – Returns a string array with the specified dimensions.

STRARR – Returns a string array.

ARRAY MANIPULATION ROUTINES

AFFINE – Applies an affine transformation to an array.

ASKEYS – Obtains the key names for a given associative array.

AVG – Returns the average value of an array or one dimension of an array.

BILINEAR – Creates an array containing values calculated using a bilinear interpolation to solve for requested points interior to an input grid.

BLOB – Isolates a homogeneous region in an array.

BLOBCOUNT – Counts homogeneous regions in an array.

BOUNDARY – Computes the boundary of a region in an array.

CORRELATE – Calculates a simple correlation coefficient for two arrays.

CPROD – Returns the Cartesian product of some arrays.

CURVATURES – Computes curvatures on a parametrically defined surface.

DICM_TAG_INFO – Extracts Digital Imaging and Communications in Medicine (DICOM) tags information from an image associative array.

DETERM – Calculates the determinant of a square, two-dimensional input variable.

DERIVN – Differentiates a function represented by an array.

EUCLIDEAN – Transforms the Euclidean metric for a Jacobian $j = \text{Jacobian}(f)$

EXPAND – Expands an array into higher dimensions.

EXTREMA – Finds the local extrema in an array.

GEOMEAN – Calculates the geometric mean of an array.

HISTN – Computes an n dimensional histogram.

HISTOGRAM – Returns the density function of an array.

INDEX_AND – Computes the logical AND for two vectors of positive integers.

INDEX_CONV – Converts one-dimensional indices to n-dimensional indices, or n-dimensional indices to 1D indices.

INDEX_OR – Computes the logical OR for two vectors of positive integers.

INTRP – Interpolates an array along one of its dimensions.

ISASKEY – Matches a key name in a given associative array.

JACOBIAN – Computes the Jacobian of a function represented by n m-dimensional arrays.

MAX – Returns the value of the largest element in an input array.

MEDIAN – Finds the median value of an array, or applies a one- or two- dimensional median filter of a specified width to an array.

MIN – Returns the value of the smallest element in array.

MOMENT – Computes moments of an array.

NEIGHBORS – Finds the neighbors of specified array elements.

NORMALS – Computes unit normals on a parametrically defined surface.

Array Manipulation Routines (cont.)

PADIT – Pads an array with variable thickness.

PRODUCT – Returns the product of all elements in an array.

REBIN – Returns a vector or array resized to the given dimensions.

REFORM – Reformats an array without changing its values numerically.

REPLV – Replicates a vector into an array.

RESAMP – Resamples an array to new dimensions.

REVERSE – Reverses a vector or array for a given dimension.

ROTATE – Returns a rotated and/or transposed copy of the input array.

SAME – Tests if two variables are the same.

SHIF – Shifts an array along one of its dimensions.

SHIFT – Shifts the elements of a vector or array along any dimension by any number of elements.

SIGMA – Calculates the standard deviation value of an array.

SLICE – Subsets an array along one of its dimensions.

SMOOTH – Smooths an array with a boxcar average of a specified width.

SORT – Sorts the contents of an array.

SORTN – Sorts an array of n-tuples.

STDEV – Computes the standard deviation and (optionally) the mean of the input array.

TENSOR_* – Compute the generalized tensor product of two arrays.

TOTAL – Sums the elements of an input array.

TRANSPOSE – Transposes the input array.

UNIQN – Finds the unique n-tuples from a set of n-tuples.

WHERE – Returns a longword vector containing the one-dimensional subscripts of the nonzero elements of the input array.

WHEREFIRST – Returns a longword vector containing the one-dimensional subscripts of the nonzero elements of the input array.

WHEREIN – Find the indices into an array where the values occur in a second array; keywords yield intersection, union, and complement.

WHEREINVEC – Finds the vectors in a matrix that occur in a second matrix.

WHEREMONO – Finds monotone sequences in a one-dimensional array.

INTERPOLATION ROUTINES

AFFINE – Applies an affine transformation to an array.

BILINEAR – Creates an array containing values calculated using a bilinear interpolation to solve for requested points interior to an input grid.

GRIDN – Grids n dimensional data.

INTERPOLATE – Interpolates scattered data at scattered locations.

INTERPOL – Performs a linear interpolation of a vector using a regular grid.

Interpolation Routines (cont.)

INTRP – Interpolates an array along one of its dimensions.

REBIN – Returns a vector or array resized to the given dimensions.

RESAMP – Resamples an array to new dimensions.

SPLINE – Performs a cubic spline interpolation.

GRAPHICS AND PLOTTING

COORDINATE CONVERSION ROUTINES

CONVERT_BASE – Converts a base-10 integer to an arbitrary base representation.

CONVERT_COORD – Converts coordinates from one coordinate system to another.

CONV_FROM_RECT – Converts rectangular coordinates (points) to polar, cylindrical, or spherical coordinates.

CONV_TO_RECT – Converts polar, cylindrical, or spherical coordinates to rectangular coordinates (points).

POLY_DEV – Returns a list of 3D points converted from normal coordinates to device coordinates.

POLY_NORM – Returns a list of 3D points converted from data coordinates to normal coordinates.

POLY_TRANS – Returns a list of 3D points transformed by a 4-by-4 transformation matrix.

GENERAL GRAPHICS ROUTINES

CPROD – Returns the Cartesian product of some arrays.

CURSOR – Reads the position of the interactive graphics cursor from the current graphics device.

DEVICE – Provides device-dependent control over the current graphics device

EMPTY – Causes all buffered output for the current graphics device to be written.

ERASE – Erases the display surface of the currently active window.

IMAGE_CONT – Overlays a contour plot onto an image display of the same array.

MOVIE – Shows a cyclic sequence of images stored in a three-dimensional array.

PLOTS – Plots vectors or points on the current graphics device in either two or three dimensions.

PROFILE – Extracts a profile from an image.

PROFILES – Interactively draw row or column profiles of the image displayed in the current window.

RDPIX – Displays the X, Y, and pixel values at the location of the cursor in the image displayed in the currently active window.

SCALE3D – Scales a three-dimensional unit cube into the viewing area.

SET_PLOT – Specifies the device type used by PV-WAVE graphics procedures.

General Graphics Routines (cont.)

SHOW3 – Displays a two-dimensional array as a combination contour, surface, and image plot. The resulting display shows a surface with an image underneath and a contour overhead.

T3D – Accumulates one or more sequences of translation, scaling, rotation, perspective, or oblique transformations.

THREED – Plots a two-dimensional array as a pseudo three-dimensional plot on the currently selected graphics device.

TVCRS – Manipulates the cursor within a displayed image, allowing it to be enabled and disabled, as well as positioned.

XYOUTS – Draws text on the currently selected graphics device starting at the designated data coordinate.

ZOOM – Expands and displays part of an image (or graphic plot) from the current window in a second window.

GRIDDING ROUTINES

PV-WAVE Foundation provides a general set of gridding functions. For sparse data sets or faults, see the *PV-WAVE: GTGRID Toolkit*

FAST_GRID2 – Returns a gridded, 1D array containing Y values, given random X,Y coordinates (this function works best with dense data points).

FAST_GRID3 – Returns a gridded, 2D array containing Z values, given random X, Y, Z coordinates (this function works best with dense data points).

FAST_GRID4 – Returns a gridded, 3D array containing intensity values, given random 4D coordinates (this function works best with dense data points).

GRID_2D – Returns a gridded, 1D array containing Y values, given random X,Y coordinates (this function works best with sparse data points).

GRID_3D – Returns a gridded, 2D array containing Z values, given random X, Y, Z coordinates (this function works best with sparse data points).

GRID_4D – Returns a gridded, 3D array containing intensity values, given random 4D coordinates (this function works best with sparse data points).

GRIDN – Grids n dimensional data.

GRID_SPHERE – Returns a gridded, 2D array containing radii, given random longitude, latitude, and radius values.

INTERPOLATE – Interpolates scattered data at scattered locations.

MAPPING ROUTINES

MAP – Plots a map.

MAP_CONTOUR – Draws a contour plot from longitude/latitude data stored in a 2D array.

MAP_LAND – For each map pixel, identify longitude, latitude, and cover type.

MAP_PLOTS – Plots vectors or points (specified as longitude/latitude data) on the current map projection.

MAP_POLYFILL – Fills the interior of a region of the display enclosed by an arbitrary 2D polygon.

MAP_REVERSE – Converts output from routines like CURSOR and WtPointer from device, normal, or data coordinates to longitude and latitude coordinates.

Mapping Routines (cont.)

MAP_VELOVECT – Draws a two-dimensional velocity field plot on a map, with each directed arrow indicating the magnitude and direction of the field.

MAP_XYOUTS – Draws text on the currently selected graphics device starting at the designated map coordinate.

USGS_NAMES – Queries a database containing names, FIPS codes, and longitude/latitude values for cities and towns in the United States. Plots a map.

PLOTTING ROUTINES

AXIS – Draws an axis of the specified type and scale at a given position.

BAR – Plots a 2D bar graph that can include stacked and grouped bars, as well as various color and fill pattern options.

BAR2D – Creates a two-dimensional bar plot.

BAR3D – Creates a three-dimensional bar plot.

BOXPLOT – Creates a standard box plot.

CONTOUR – Draws a contour plot from data stored in a rectangular array.

CONTOUR2 – Draws a contour plot from data stored in a rectangular array.

CONTOURFILL – Fills both open and closed contours with specified colors or patterns.

CONTOURFS – Provides a wrapper which calls CONTOUR and CONTOURFILL that accepts all CONTOUR and CONTOURFILL keywords with the exception of Color_Index.

CORRELOGRAM – Plots a correlogram using either autocorrelation function (ACF) values or partial autocorrelation function (PACF) values.

CURSOR – Reads the position of the interactive graphics cursor from the current graphics device.

CUSUM – Creates a two-sided, V-mask Cumulative Sum chart of a process mean.

ERRPLOT – Overplots error bars over a previously-drawn plot.

GRID – Generates a uniform grid from irregularly-spaced data.

IMAGE_CONT – Overlays a contour plot onto an image display of the same array.

MACHART – Creates a standard or Exponentially Weighted Moving Average (EWMA) chart of a process mean.

OPLLOT – Plots vector data over a previously drawn plot.

OPLOTERR – Overplots symmetrical error bars on any plot already output to the display device.

PARETO – Creates a PARETO chart with accompanying legend, cumulative percentages and annotations.

PIE – Displays data as a pie chart.

PIE_CHART – Creates a pie chart with colors, text labels, exploded slices and/or shadows.

PLOT – Produces a simple XY plot.

PLOT_HISTOGRAM – Plots a histogram.

PLOT_IO – Produces an XY plot with logarithmic scaling on the Y axis.

Plotting Routines (cont.)

PLOT_OI – Produces an XY plot with logarithmic scaling on the X axis.

PLOT_OO – Produces an XY plot with logarithmic scaling on both the X and Y axes.

PLOTERR – Plots data points with accompanying symmetrical error bars.

PLOT_FIELD – Plots a two-dimensional velocity field.

PLOTS – Plots vectors or points on the current graphics device in either two or three dimensions.

PLOT_WINDROSE – Creates a Wind Rose plot.

POLYFILL – Fills the interior of a region of the display enclosed by an arbitrary two- or three-dimensional polygon.

POLYSHADE – Constructs a shaded surface representation of one or more solids described by a set of polygons.

PROFILE – Extracts a profile from an image.

PROFILES – Interactively draw row or column profiles of the image displayed in the current window. The profiles are displayed in a new window, which is deleted when you exit the procedure.

SCALE3D – Scales a three-dimensional unit cube into the viewing area.

SET_SHADING – Modifies the light source shading parameters affecting the output of SHADE_SURF and POLYSHADE.

SHADE_SURF – Creates a shaded surface representation of a regular or nearly regular gridded surface, with shading.

SHADE_SURF_IRR – Creates a shaded-surface representation of a semiregularly gridded surface, with shading from either a light source model or from a specified array of intensities.

SHOW3 – Displays a two-dimensional array as a combination contour, surface, and image plot. The resulting display shows a surface with an image underneath and a contour overhead.

SURFACE – Draws the surface of a two-dimensional array projected into two dimensions, with hidden lines removed.

T3D – Accumulates one or more sequences of translation, scaling, rotation, perspective, or oblique transformations.

THREED – Plots a two-dimensional array as a pseudo three-dimensional plot.

USERSYM – Lets you create a custom symbol for marking plotted points.

VEL – Draws a graph of a velocity field with arrows pointing in the direction of the field. The length of an arrow is proportional to the strength of the field at that point.

VELOVECT – Draws a two-dimensional velocity field plot, with each directed arrow indicating the magnitude and direction of the field.

XBAR – Creates an X Bar chart.

XRCHART – Creates an XR chart with an X Bar chart on top and an R chart beneath it.

XSCHART – Creates an XS chart with an X Bar chart on top and an S chart beneath it.

VIEW SETUP ROUTINES

CENTER_VIEW – Sets system viewing parameters to display data in the center of the current window (a convenient way to set up a 3D view).

SET_VIEW3D – Generates a 3D view, given a view position and a view direction.

T3D – Accumulates one or more sequences of translation, scaling, rotation, perspective, or oblique transformation.

VIEWER – Defines a 3D view, a slicing plane, and multiple cut-away volumes for volume rendering.

WINDOWS ROUTINES

WCOPY – (Microsoft Windows Only) Copies the contents of a graphics window onto the Clipboard.

WDELETE – Deletes the specified window.

WINDOW – Creates a window for the display of graphics or text.

WMENU – Displays a menu inside the current window whose choices are given by the elements of a string array and which returns the index of the user's response.

WPASTE – (Microsoft Windows Only) Pastes the contents of the Clipboard into a graphics window.

WPRINT – (Microsoft Windows Only) Prints the contents of a specified window.

WREAD_DIB – (Microsoft Windows Only) Loads a Device Independent Bitmap (DIB) from a file into a graphics window.

WREAD_META – (Microsoft Windows Only) Loads a Windows metafile (WMF) into a graphics window.

WSET – Used to select the current window to be used by the graphics and imaging routines.

WSHOW – Exposes or hides the designated window. It does not automatically make the designated window the active window.

WWRITE_DIB – (Microsoft Windows Only) Saves the contents of a graphics window to a file as a Device Independent Bitmap (DIB).

WWRITE_META – (Microsoft Windows Only) Saves the contents of a graphics window to a file as a Windows metafile (WMF).

VDA TOOLS ROUTINES

Navigator – Starts the Navigator.

WzAnimate – Starts a VDA Tool used for animating a sequence of images.

WzClose – Closes all VDA tools registered with the Tools Manager.

WzColorEdit – Starts a VDA Tool used for editing the image and plot color tables used in other VDA Tools.

WzContour – Starts a VDA Tool used for plotting contours.

WzExport – Starts a VDA Tool used for exporting a PV-WAVE variable to an external file in a specified format.

VDA Tools Routines (cont.)

WzHistogram – Starts a VDA Tool used for plotting a histogram.

WzImage – Starts a VDA Tool used for displaying image data.

WzImport – Starts a VDA Tool used for importing data into PV-WAVE.

WzInit – Initializes the VDA Tools Manager layer.

WzMultiView – Starts a VDA Tool used to display multiple plots.

WzPlot – Starts a VDA Tool used for 2D plotting.

WzPreview – Starts a VDA Tool used to view an ASCII file's contents and select which parts of the file are to be read in as PV-WAVE variables.

WzSurface – Starts a VDA Tool used for surface plots.

WzTable – Starts a VDA Tool used for creating an editable2D array of cells containing string data.

WzVariable – Starts a VDA Tool used for viewing and exporting variables.

GUI DEVELOPMENT

CONCURRENT PROCESSING ROUTINES

ADD_EXEC_ON_SELECT – Adds a single new item to the EXEC_ON_SELECT list.

DROP_EXEC_ON_SELECT – Drops a single item from the EXEC_ON_SELECT list.

EXEC_ON_SELECT – Registers callback procedures on input for a vector of logical unit numbers (LUNs).

SELECT_READ_LUN – Waits for input on any list of logical unit numbers.

RESOURCE FILES

BUILDRESOURCEFILENAME – Returns the full pathname for a specified resource file.

LOADRESOURCES – Loads resources from a resource file.

LOADSTRINGS – Loads strings from a resource file.

VDA TOOLS MANAGER ROUTINES (TM)

TmAddSelectedVars – Adds selected variables from a VDA Tool to the list of selected variables in the Tools Manager.

TmAddVar – Adds a variable to a VDA Tool.

TmAxis – Adds axes to a VDA Tool.

TmCodeGen – Writes a specified string to the code generation file.

TmCopy – Copies the selected graphical elements from the specified VDA Tool to the clipboard.

TmCut – Cuts the selected graphical elements from the specified VDA Tool and moves them to the clipboard.

TmDelVar – Removes variables from a VDA Tool.

VDA Tools Manager Routines (TM) [cont.]

TmDelete – Permanently deletes the selected graphical elements from the specified VDA Tool.

TmDeselectVars – Clears the current list of selected variables.

TmDynamicDisplay – Displays selected data in all active VDA Tools, provided that the VDA Tools can display the related variable(s).

TmEndCodeGen – Closes the file in which generated code is written.

TmEnumerateAttributes – Obtains the attributes for a specified graphical element, variable, or other item in a VDA Tool.

TmEnumerateItems – Obtains the items defined for a specified VDA Tool.

TmEnumerateMethods – Obtains the methods that were set for a VDA Tool.

TmEnumerateSelectedVars – Returns the names of variables on the selected variables list.

TmEnumerateToolNames – Returns all the registered VDA Tool names.

TmEnumerateVars – Returns all the variables associated with an instance of a VDA Tool.

TmExecuteMethod – Executes a method that was set by TmSetMethod.

TmExport – Exports \$MAIN\$-level variables to specified VDA Tools or to all currently active VDA Tools.

TmExportSelection – Exports the contents of the variable selection list to specified VDA Tools.

TmGetAttribute – Returns the value that was set for an attribute in a VDA Tool instance.

TmGetMessage – Loads a string resource file into the resource database and extracts a message string from the database.

TmGetMethod – Returns the data structure of the specified method.

TmGetTop – Gets the top-level widget ID for a VDA Tool.

TmGetUniqueToolName – Returns a unique name for a particular instance of a specified VDA Tool.

TmGetVarMainName – Returns the \$MAIN\$ level name of a variable.

TmInit – Initializes the VDA Tools Manager layer.

TmList – Creates a list item.

TmListAppend – Adds a new item at the end of the specified list.

TmListClear – Resets a specified list to its initial state, clearing all previously defined items.

TmListDelete – Deletes an item in the specified list.

TmListDestroy – Clears all items and destroys the list.

TmListExtend – Extends the specified list by adding empty items.

TmListGetMethod – Returns the procedure name associated with the specified list method name.

TmListInsert – Inserts a new item into the specified list.

TmListReplace – Replaces an item in a list with a new item.

TmListRetrieve – Gets the items currently set in the specified list.

TmListSetMethod – Sets the method procedure name for a specific list method.

VDA Tools Manager Routines (TM) [cont.]

TmPaste – Pastes the graphical elements from the clipboard to the specified VDA Tool.

TmRegister – Registers a VDA Tool with the Tools Manager.

TmRestoreTemplate – Restores a saved VDA Tool template.

TmRestoreTools – Restores the VDA Tools that were saved with the TmSaveTools procedure.

TmSaveTools – Saves the specified VDA Tools in a file.

TmSetAttribute – Set an attribute for an item in the given VDA Tool.

TmSetMethod – Sets a method for a given VDA Tool.

TmStartCodegen – Opens a file into which PV-WAVE code is written.

TmUnregister – Removes the specified VDA Tool from the Tools Manager registry.

VDA TOOLS MANAGER GRAPHICAL ELEMENT ROUTINES (TM)

TmAddGrael – Adds a graphical element to the graphical element list for the specified instance of a VDA Tool.

TmAddSelectedGrael – Adds a graphical element to the graphical element selection list.

TmAxis – Adds axes to a VDA Tool.

TmBitmap – Adds a bitmap (2D array) to a VDA Tool.

TmBottomGrael – Sets the specified graphical element to be on the bottom of the display list (displayed behind the other graphical elements).

TmDelGrael – Removes a specified graphical element from the list of graphical elements associated with a VDA Tool instance.

TmDelSelectedGraels – Deletes a graphical element from the list of selected graphical elements.

TmEnumerateGraelMethods – Obtain a list of all the methods set for a graphical element in a specified VDA Tool.

TmEnumerateGraels – Returns all the graphical elements that were set for a given VDA Tool.

TmEnumerateSelectedGraels – Obtains a list of graphical elements or other items currently on the graphical items selection list.

TmExecuteGraelMethod – Executes a method for a graphical method based on the method name.

TmGetGraelMethod – Obtains the data structure for the specified method.

TmGetGraelRectangle – Returns the rectangular boundary of a graphical element.

TmGetUniqueGraelName – Obtains a unique name based on the name of the specified graphical element.

TmGroupGraels – Groups a number of selected graphical elements as one graphical element with a unique name.

TmLegend – Adds a legend to a VDA Tool. The user determines the exact size and position of the legend interactively.

TmLine – Adds a line to a VDA Tool. The user determines the exact length and position of the line interactively.

TmRect – Adds a rectangle to a VDA Tool. The user determines the exact size and position of the rectangle interactively.

TmSetGraelMethod – Sets the name of the method procedure for a given method name and graphical element.

VDA Tools Manager Graphical Element Routines (TM) (cont.)

TmSetGraeiRectangle – Sets the selection rectangle for a graphical element, or a set of graphical elements.

TmText – Adds text to a VDA Tool. The user determines the position of the text and the text itself interactively.

TmTopGraei – Sets the specified graphical element to be at the top of the display list (displayed in front of other graphical elements).

TmUngroupGraels – Ungroups a group of graphical elements.

VDA UTILITIES ROUTINES (WO)

WoAddButtons – Adds a bank of buttons to a button bar.

WoAddMessage – Adds a message to a message area created by WoMessage.

WoAddStatus – Display a message in the status bar of a VDA Tool.

WoBuildResourceFilename – Returns the full path name for a specified resource file.

WoButtonBar – Creates a predefined, two-row button bar that can be included in a VDA Tool.

WoButtonBarSet – Changes the setting of a button in a button bar.

WoButtonBarSetSensitivity – Sets the sensitivity of one or more buttons on a button bar.

WoCheckFile – Confirms if a file is readable or writable.

WoColorButton – Creates a button that brings up a color table dialog box used to set colors in a VDA Tool. The button has an associated color pixmap that reflects the currently selected color.

WoColorButtonGetValue – Gets the currently selected color index from a color button created by WoColorButton.

WoColorButtonSetValue – Sets the current color index for a color button created by WoColorButton, and updates the color button's color pixmap.

WoColorConvert – Convert from a long RGB value to an index into the current color table, or from an index in the current color table to an RGB value.

WoColorGrid – Creates a grid of color squares from the current color table.

WoColorGridGetValue – Gets the color indices for a range of colors in a color grid.

WoColorGridSetValue – Sets the color indices for a range of colors in the color grid.

WoColorWheel – Creates a color wheel that can be used to modify a single color in the current color table.

WoConfirmClose – Displays a dialog box requiring the user to confirm a window close action.

WoDialogStatus – Saves or restores the status of a dialog box by saving or restoring the state of its widgets as stored in the Tools Manager.

WoFontOptionMenu – Creates an option menu with the standard list of software (vector-drawn) fonts found in PV-WAVE.

WoFontOptionMenuGetValue – Gets the software font command for the currently selected font.

WoFontOptionMenuSetValue – Sets the current font and updates the font option menu.

VDA Utilities Routines (WO) (cont.)

WoGenericDialog – Creates a generic dialog box for use in VDA Tools.

WoGetToolNameFromTitle – Gets the unique name of a VDA tool given the unique window title of the VDA Tool.

WoGetUniqueWindowTitle – Given a window title, adds a numeric suffix to make the title unique.

WoLabeledText – Creates a group of aligned text widgets (widgets with a label and a text field).

WoLinestyleOptionsMenu – Creates an option menu for selecting linestyles.

WoLinestyleOptionsMenuGetValue – Gets the currently selected linestyle.

WoLineStyleOptionsMenuSetValue – Sets the option menu to a specified linestyle.

WoLoadResources – Loads resources and strings from a file for VDA tools.

WoLoadStrings – Loads strings from a resource file for use by the VDA tools.

WoMenuBar – Create a menu bar for a VDA Tool.

WoMenuBarSetSensitivity – Sets the sensitivity of one or more items in a menu.

WoMenuBarSetToggle – Sets the status of a menu toggle button.

WoMessage – Creates a message area for a VDA Tool

WoSaveAsPixmap – Saves graphics from a specified VDA Tool as a pixmap.

WoSetCursor – Changes the cursor for a VDA Tool.

WoSetToolIcon – Assigns a pixmap to be the icon for a VDA Tool.

WoSetWindowTitle – Specifies a unique title for a VDA Tool window.

WoStatus – Create a status bar for a VDA Tool.

WoVariableOptionsMenu – Creates an option menu containing the names of all of the variables associated with the current tool.

WoVariableOptionsMenuGetValue – Gets the currently selected variable name from an option menu that was created with the **WoVariableOptionsMenu** function.

WoVariableOptionsMenuSetValue – Sets the current selection in the variable option menu.

WAVE WIDGETS ROUTINES (WW)

WwAlert – Creates a modal (blocking) or modeless (non-blocking) popup alert box containing a message and optional control buttons.

WwAlertPopdown – Destroys an alert box.

WwButtonBox – Creates a horizontally or vertically oriented box containing push buttons.

WwCallback – Adds or removes a WAVE Widgets callback.

WwCommand – Creates a command window.

WwControlsBox – Creates a box containing sliders.

WwDialog – Creates a blocking or nonblocking dialog box.

Wave Widgets Routines (WW) (cont.)

WwDrawing – Creates a drawing area, which allows users to display graphics generated by PV-WAVE.

WwFileSelection – Creates a file selection widget, which lets the user display the contents of directories and select files.

WwGenericDialog – Creates a generic dialog box that can be filled with custom widgets.

WwGetButton – Obtains the index of a pressed or released button passed as an event structure by a WAVE Widgets event handler.

WwGetKey – Obtains the ASCII value of a pressed or released key passed as an event structure by a WAVE Widgets event handler.

WwGetPosition – Obtains coordinates of a selected point inside a widget. The selected point coordinates are passed in an event structure by a WAVE Widgets event handler.

WwGetValue – Returns a specific value for a given widget.

WwHandler – Adds or removes a WAVE Widgets event handler from a widget.

WwInit – Initializes the WAVE Widgets environment, opens the display, creates the first top-level shell, and creates a layout widget.

WwLayout – Creates a layout widget that is used to control the arrangement of other widgets.

WwList – Creates a scrolling list widget.

WwListUtils – Manages the contents of a list widget.

WwLoop – Handles the dispatching of events and calling of PV-WAVE callbacks.

WwMainWindow – Creates top-level window and layout widget.

WwMenuBar – Creates a menu bar.

WwMenuItem – Adds, modifies, or deletes specified menu items.

WwMessage – Creates a blocking or nonblocking message box.

WwMultiClickHandler – Adds or removes a multi-click event handler.

WwOptionsMenu – Creates an option menu.

WwPickFile – Creates a modal file selection dialog that blocks until a file name has been selected.

WwPopupMenu – Creates a popup menu.

WwPreview – Creates an ASCII data preview widget.

WwPreviewUtils – Manages the contents of a preview widget.

WwRadioBox – Creates a box containing radio buttons.

WwResource – Queries, creates, saves, or modifies the widget resource database.

WwSeparator – Creates a horizontal or vertical line that separates components in a graphical user interface.

WwSetCursor – Sets the cursor for a widget.

WwSetValue – Sets the specified value for a given widget.

WwTable – Creates an editable 2D array of cells containing string data, similar to a spreadsheet.

WwTableUtils – Manages the contents of a table widget.

WwText – Creates a text widget that can be used for both single-line text entry or as a full text editor. In addition, this function can create a static text label.

WwTimer – Registers a WAVE Widgets timer procedure.

WwToolBox – Creates an array of graphic buttons (icons).

WAVE WIDGETS UTILITIES (WG)

WgAnimateTool – Creates a window for animating a sequence of images.

WgCbarTool – Creates a simple color bar that can be used to view and interactively shift a PV-WAVE color table.

WgCeditTool – Creates a full-featured set of menus and widgets enclosed in a window; this window allows you to edit the values in PV-WAVE color tables in many different ways.

WgControlsBox – Displays slider and returns user-selected slider value.

WgCtTool – Creates a simple widget that can be used interactively to modify a PV-WAVE color table.

WgFileSelection – Creates a file selection widget which lets the user display the contents of directories and select files.

WgIsoSurfTool – Creates a window with a built-in set of controls; these controls allow you to easily view and modify an iso-surface taken from a three-dimensional block of data.

WgList – Displays a list of items and returns an array of indices into the list corresponding to user-selected items.

WgOrbit – Creates an interactive window for viewing objects.

WgMovieTool – Creates a window that cycles through a sequence of images.

WgImageTool – Creates a GUI with sliders allowing the user to modify pixel values of the image and get basic statistics of the image.

WgSliceTool – Creates a window with a built-in set of controls; these controls allow you to easily select and view “slices” from a three-dimensional block of data.

WgSlideShow – Views multiple images in a slide show widget.

WgStripTool – Creates a window that displays data in a style that simulates a real-time, moving strip chart.

WgSurfaceTool – Creates a surface window with a built-in set of controls: these controls allow you to interactively modify surface parameters and view the result of those modifications.

WgText – Displays a single-line text field for user input.

WgTextTool – Creates a scrolling window for viewing text from a file or character string.

WIDGET TOOLBOX ROUTINES (WT)

WtAddCallback – Registers a PV-WAVE callback routine for a given widget.

WtAddHandler – Registers the X event handler function for a given widget.

WtClose – Closes current Xt (Motif) session, and destroys all children of the top-level widget created in WtInit. This routine can also be used to destroy additional widget trees.

WtCreate – Creates a widget or shell instance specified by widget class.

WtCursor – Sets or changes the cursor.

WtGet – Retrieves widget resources.

WtInit – Initializes the Widget Toolbox and the Xt toolkit, opens the display, and creates the first top-level shell.

Widget Toolbox Routines (WT) (cont.)

WtInput – Registers a PV-WAVE input source handler procedure.

WtList – Controls the characteristics of scrolling list widgets.

WtLookupString – Maps a KeyPress or KeyRelease event to its KeyEvent structure (and optionally, to its Keysym) when a user presses a key.

WtLoop – Handles the dispatching of events and calling of PV-WAVE callback routines.

WtMainLoop – Handles the dispatching of events.

WtPointer – The pointer utility function.

WtPreview – Handles utility functions for the preview widget (XvnPreview).

WtProcessEvent – Handles the dispatching of a Widget Toolbox event.

WtResource – Queries, creates, saves, or modifies the widget resource database.

WtSet – Sets widget resources.

WtTable – Modifies an xbaeMatrix class widget.

WtTimer – Registers a callback function for a given timer.

WtWorkProc – Registers a PV-WAVE work procedure for background processing.

IMAGE PROCESSING AND COLOR

PV-WAVE Foundation provides a basic set of image processing functions. For more advanced functions, see the PV-WAVE: Image Processing Toolkit

COLOR TABLE MANIPULATION ROUTINES

C_EDIT – Interactively create a new color table based on the HLS or HSV color system.

COLOR_PALETTE – Displays the current color table colors and their associated color table indices.

HLS – Generates and loads color tables into an image display device based on the HLS color system.

HSV – Generates and loads color tables into an image display device based on the HSV color system.

LOADCT – Loads a predefined color table.

LOADCT_CUSTOM – Loads a predefined custom color table.

MODIFYCT – Replace one of the PV-WAVE color tables with a new color table.

PALETTE – Interactively create a new color table based on the RGB color system.

PSEUDO – Creates a pseudo color table based on the Hue, Lightness, Saturation (HLS) color system.

Color Table Manipulation Routines (cont.)

STRETCH – Linearly expands the range of the color table currently loaded to cover an arbitrary range of pixel values.

TVLCT – Loads the display color translation tables from the specified variables.

WgCbarTool – Creates a simple color bar that can be used to view and interactively shift a color table.

WgCeditTool – Creates a full-featured set of menus and widgets enclosed in a window; this window allows you to edit the values in PV-WAVE color tables in many different ways.

WgCtTool – Creates a simple widget that can be used interactively to modify a PV-WAVE color table.

IMAGE DISPLAY ROUTINES

COLOR_BAR – Creates a horizontal or vertical color bar with associated legends.

TV – Displays images without scaling the intensity.

TVCRS – Manipulates the cursor within a displayed image, allowing it to be enabled and disabled, as well as positioned.

TVLCT – Loads the display color translation tables from the specified variables.

TVRD – Returns the contents of the specified rectangular portion of a displayed image.

TVSCL – Scales the intensity values of an input image into the range of the image display, usually from 0 to 255, and outputs the data to the image display at the specified location.

TVSIZE – Displays images at the current or specified size and device resolution.

IMAGE I/O ROUTINES

DICM_TAG_INFO – Extracts Digital Imaging and Communications in Medicine (DICOM) tags information from an image associative array.

IMAGE_CHECK – Determines that the input variable is an associative array in image format. The function also checks to make sure that all fields in the image associative array are present.

IMAGE_COLOR_QUANT – Quantizes a 24-bit image to 8-bit pseudo color.

IMAGE_CREATE – Creates an associative array in image format.

IMAGE_DISPLAY – Displays an image.

IMAGE_QUERY_FILE – Return the type of a specified image file.

IMAGE_READ – Reads image files and returns an associative array in image format.

IMAGE_WRITE – Writes PV-WAVE graphics to a specified file type.

READ_XBM – Reads the contents of an X-bitmap (XBM) file into a PV-WAVE variable.

WRITE_XBM – Writes an image to an X-bitmap (XBM) file.

IMAGE PROCESSING ROUTINES

AFFINE – Applies an affine transformation to an array.

BLOB – Isolates a homogeneous region in an array.

BLOBCOUNT – Counts homogeneous regions in an array.

BOUNDARY – Computes the boundary of a region in an array.

CONGRID – Shrinks or expands an image or array.

CONVOL – Convolves an array with a kernel (or another array).

DEFROI – Defines an irregular region of interest within an image by using the image display system and the mouse.

DERIVN – Differentiates a function represented by an array.

DIGITAL_FILTER – Constructs finite impulse response digital filters for signal processing.

DILATE – Implements the morphologic dilation operator for shape processing.

DIST – Generates a square array in which each element equals the euclidean distance from the nearest corner.

ERODE – Implements the morphologic erosion operator for shape processing.

FFT – Returns the Fast Fourier Transform for the input variable.

HANNING – Implements a window function for Fast Fourier Transform signal or image filtering.

HIST_EQUAL – Returns a histogram-equalized image or vector.

HISTOGRAM – Returns the density function of an array.

IMAGE_CONT – Overlays a contour plot onto an image display of the same array.

LEEFILT – Performs image smoothing by applying the Lee Filter algorithm.

MEDIAN – Finds the median value of an array, or applies a one- or two- dimensional median filter of a specified width to an array.

MOMENT – Computes moments of an array.

MOVIE – Shows a cyclic sequence of images stored in a three-dimensional array.

NEIGHBORS – Finds the neighbors of specified array elements.

POLY_2D – Performs polynomial warping of images.

POLYFILLV – Returns a vector containing the subscripts of the array elements contained inside a specified polygon.

POLYWARP – Calculates the co's needed for a polynomial image warping transformation.

PROFILE – Extracts a profile from an image.

PROFILES – Interactively draw row or column profiles of the image displayed in the current window.

RDPIX – Displays the X, Y, and pixel values at the location of the cursor in the image.

REBIN – Returns a vector or array resized to the given dimensions.

REFORM – Reformats an array without changing its values numerically.

RESAMP – Resamples an array to new dimensions.

Image Processing Routines (cont.)

ROBERTS – Performs a Roberts edge enhancement of an image.

ROT – Rotates and magnifies (or demagnifies) a two-dimensional array.

ROT_INT – Rotates and magnifies (or demagnifies) an image on the display screen.

ROTATE – Returns a rotated and/or transposed copy of the input array.

SHIFT – Shifts the elements of a vector or array along any dimension by any number of elements.

SHOW3 – Displays a two-dimensional array as a combination contour, surface, and image plot. The resulting display shows a surface with an image underneath and a contour overhead.

SMOOTH – Smooths an array with a boxcar average of a specified width

.SOBEL – Performs a Sobel edge enhancement of an image.

TRANSPOSE – Transposes the input array.

ZOOM – Expands and displays part of an image (or graphic plot) from the current window in a second window.

INPUT/OUTPUT

DATA CONNECTION ROUTINES

DC_ERROR_MSG – Returns the text string associated with the negative status code generated by a “DC” data import/export function that does not complete successfully.

DC_OPTIONS – Sets the error message reporting level for all “DC” import/export functions.

DC_READ_8_BIT – Reads an 8-bit image file.

DC_READ_24_BIT – Reads a 24-bit image file.

DC_READ_CONTAINER – Reads a single variable from an HP VEE Container file.

DC_READ_DIB – Reads data from a Device Independent Bitmap (DIB) format file into a variable.

DC_READ_FIXED – Reads fixed-formatted ASCII data using a PV-WAVE format that you specify.

DC_READ_FREE – Reads freely-formatted ASCII files.

DC_READ_TIFF – Reads a Tag Image File Format (TIFF) file.

DC_SCAN_CONTAINER – Scans an HP VEE Container file to determine the number and location of defined variables.

DC_WRITE_8_BIT – Writes 8-bit image data to a file.

DC_WRITE_24_BIT – Writes 24-bit image data to a file.

DC_WRITE_DIB – Writes image data from a variable to a Device Independent Bitmap (DIB) format file.

DC_WRITE_FIXED – Writes the contents of one or more PV-WAVE variables (in ASCII fixed format) to a file using a format that you specify.

DC_WRITE_FREE – Writes the contents of one or more PV-WAVE variables to a file in ASCII free format.

DC_WRITE_TIFF – Writes image data to a file using the Tag Image File Format (TIFF) format.

HDF4 ROUTINES

GETNCERR – Retrieves the current value of the “ncerr” variable as discussed in the error section of the NetCDF User's Guide.

GETNCOPTS – Retrieves the current value of the ncopts variable as discussed in the error section of the NetCDF User's Guide.

HDFGET24 – Obtains an HDF Raster 24 image.

HDFGETANN – Obtains HDF object (e.g., an SDS, Raster 8image, etc.) annotations, either a label or a description.

HDFGETFILEANN – Obtains an HDF file annotation, either label or description.

HDFGETNT – Obtains the HDF number type (i.e., data type) and descriptive number type string for the current HDF Scientific Data Set.

HDFGETR8 – Obtains an HDF Raster 8 image and associated palette.

HDFGETRANGE – Gets the maximum and minimum range for the current HDF Scientific Data Set.

HDFGETSDS – Gets an HDF Scientific Data Set.

HDFLCT – Loads an HDF palette as a PV-WAVE color table.

HDFPUT24 – Puts an HDF Raster 24 image into an HDF file.

HDFPUTFILEANN – Inserts HDF file labels and file descriptions (annotations) into a file.

HDFPUTR8 – Writes an 8 bit image to an HDF file.

HDFPUTSDS – Writes a Scientific Data Set to an HDF file.

HDFSCAN – Scans an HDF file and prints a simple list of file contents by HDF object type.

HDFSETNT – Computes and sets the HDF number type(i.e., data type) and descriptive number type string for the specified data array.

HDF_STARTUP – A batch file used to initialize the HDF interface. See PV-WAVE Reference Appendix A, “The PV-WAVE HDF Interface” for more information.

HDF_TEST – Runs the PV-WAVE HDF test suite.

SETNCOPTS – Sets the value of the ncopts variable and defines the level of error reporting for the NetCDF functions as discussed in the error section of the NetCDF User's Guide.

INPUT AND OUTPUT ROUTINES

ASSOC – Associates an array structure with a file, allowing random access input and output.

BYTEORDER – Converts integers between host and network byte ordering. Can also be used to swap the order of bytes within both short and long integers.

GET_KBRD – Returns the next character available from standard input

LN03 – Opens or closes an output file for LN03 graphics output.

PRINT – PRINT performs output to the standard output stream.

Input and Output Routines (cont.)

PRINTF – PRINTF requires the output file unit to be specified.

READ – Read input from the standard input stream into PV-WAVE variables.

READ_AIRS – Reads in hourly data from a file in the AIRS (Aerometric Information Retrieval System) format.

READF – Read input from a file into PV-WAVE variables.

READU – Reads binary (unformatted) input from a specified file.

READU_32 – READU_32 reads a long data type as a 4-byte value from the given input stream. It is useful in reading 32-bit Longs on a 64-bit system. In addition, it can be used to read any other PV-WAVE data types.

REWIND – (OpenVMS Only) Rewinds the tape on the designated PV-WAVE tape unit.

SKIPF – (OpenVMS Only) Skips files or records on the designated magnetic tape unit.

SKIPF – (OpenVMS Only) Skips records on the designated magnetic tape unit.

TAPRD – (OpenVMS Only) Reads the next record on the selected tape unit into the specified array.

TAPWRT – (OpenVMS Only) Writes data from the input array to the selected tape unit.

WPRINT – (Microsoft Windows Only) Prints the contents of a specified window.

WREAD_DIB – (Microsoft Windows Only) Loads a Device Independent Bitmap (DIB) from a file into a graphics window.

WREAD_META – (Microsoft Windows Only) Loads a Windows metafile (WMF) into a graphics window.

WRITEU – Writes binary (unformatted) data from an expression into a file.

WRITE_DIB – (Microsoft Windows Only) Saves the contents of a graphics window to a file as a Device Independent Bitmap (DIB).

WRITE_META – (Microsoft Windows Only) Saves the contents of a graphics window to a file as a Windows metafile (WMF).

INTERNET ENABLING

HYPertext MARKUP LANGUAGE (HTML) ROUTINES

HTML_BLOCK – Writes out a specifically formatted “block” of HTML text.

HTML_CLOSE – Closes an HTML file, after end-tagging major elements.

HTML_HEADING – Creates a heading, with a level specification.

HTML_HIGHLIGHT – Allows for all the basic textual highlighting elements in HTML.

HTML_IMAGE – Returns a string or an array of strings containing a reference or references to image URL(s).

HTML_LINK – Sets up links to Uniform Resource Locations (URLs).

HTML_LIST – Generates HTML code for all types of lists.

HTML_OPEN – Opens the output HTML file, writes out the basic HTML information and sets an HTML output file information variable, hinfo.

Hypertext Markup Language (HTML) Routines (cont.)

HTML_PARAGRAPH – Defines an HTML paragraph.

HTML_RULE – Inserts a horizontal-line separator.

HTML_SAFE – Escapes special characters so that the HTML displays them as intended, rather than using them for format tagging. The escapes codes are for: <, >, &, and " .

HTML_TABLE – Writes out an HTML table.

OPENURL – Opens a file on the Internet to be accessed (through Java) using PV-WAVE.

XML ROUTINES

SaxInit – Parses an XML document and registers the event handler function for execution.

XmlAddNode – Adds a node created with XmlNewNode to a node tree.

XmlDocDump – Creates an XML document either in memory or in an external file from an XML Document Object Model (DOM).

XmlEvaluate – Evaluates an XPATH expression and returns the selected nodes or by default returns all nodes in the XML document.

XmlFreeDoc – Frees memory for XML documents in memory (DOM representation).

XmlFreeNode – Frees memory for XML documents in memory (DOM representation).

XmlNewDoc – Creates a new XML document in memory as an XML Document Object Model (DOM).

XmlNewNode – Creates a new XML Node.

XmlParse – Parses an XML file or an in memory XML document and builds an internal Document Object Model tree which can be used in other XML functions.

XmlSetAttr – Sets an attribute for a node.

XmlSetContent – Sets content for a node.

XmlSetContext – Sets the star location or the context of the parsed document.

VIRTUAL REALITY MODELING LANGUAGE (VRML) ROUTINES

VRML_AXIS – Adds an axis to a VRML world.

VRML_CAMERA – Positions a VRML camera for rendering the VRML view.

VRML_CLOSE – Closes the VRML file.

VRML_CONE – Creates a VRML cone.

VRML_CUBE – Positions a VRML cube in the world.

VRML_CYLINDER – Positions a VRML cylinder in the world.

VRML_LIGHT – Sets up the light source for a VRML world.

Virtual Reality Modeling Language (VRML) Routines (cont.)

VRML_LINE – Creates a VRML line object.

VRML_OPEN – Opens a VRML file and writes out header information consistent with VRML formatting.

VRML_POLY – Creates a VRML polyline node, based on PV-WAVE's variables for vertex list and polygon list.

VRML_SPHERE – Creates a sphere in a VRML world.

VRML_SPOTLIGHT – Creates a VRML spotlight.

VRML_SURFACE – Creates a VRML surface node based on PV-WAVE-type variables.

VRML_TEXT – Creates a VRML text object in an open VRML file.

MATHEMATICS

PV-WAVE Foundation provides a basic set of mathematical and statistical functions. For more advanced functions, see the PV-WAVE: IMSL Mathematics Toolkit and the PV-WAVE: IMSL Statistics Toolkit.

GENERAL MATHEMATICS FUNCTIONS

ABS – Returns the absolute value of x.

AVG – Returns the average value of an array or of one dimension of an array.

BILINEAR – Creates an array containing values calculated using a bilinear interpolation to solve for requested points interior to an input grid.

CHECK_MATH – Returns and clears the accumulated math error status.

CONJ – Returns the complex conjugate of the input variable.

CONVOL – Convolve an array with a kernel (or another array).

CORRELATE – Calculates a simple correlation coefficient for two arrays.

CROSSP – Returns the cross product of two three-element vectors.

CURVATURES – Computes curvatures on a parametrically defined surface.

CURVEFIT – Performs a nonlinear least-squares fit to a function of an arbitrary number of parameters.

DERIV – Calculates the first derivative of a function in x and y.

DERIVN – Differentiates a function represented by an array.

DETERM – Calculates the determinant of a square, two-dimensional input variable.

EUCLIDEAN – Transforms the Euclidean metric for a Jacobian $j = \text{Jacobian}(f)$

EXPON – Performs general exponentiation.

FACTOR – Returns the prime factorization of an integer greater than 1.

FFT – Returns the Fast Fourier Transform for the input variable.

FINITE – Returns a value indicating if the input variable is finite or not.

GAUSSFIT – Fits a Gaussian curve through a data set.

GCD – Returns the greatest common divisor of some integers greater than 0.

GREAT_INT – Greatest Integer Function. Standard Library function that returns the greatest integer less than or equal to the passed value. Also known as the Floor Function.

General Mathematics Functions (cont.)

HILBERT – Constructs a Hilbert transformation matrix.

IMAGINARY – Returns the imaginary part of a complex number.

INVERT – Returns an inverted copy of a square array.

IQR – Computes elements of the inter-quartile range for the Tukey Box Plot as defined in “The Elements of Graphing Data,” Cleveland, W.S., Hobart Press, 1994, pp. 139-142.

ISHFT – Performs the bit shift operation on bytes, integers, and longwords.

JACOBIAN – Computes the Jacobian of a function represented by n m -dimensional arrays.

LCM – Returns the least common multiple of some integers greater than 1.

LUBKSB – Solves the set of n linear equations $Ax = b$.

LUDCMP – Replaces an n -by- n matrix, a , with the LU decomposition of a row-wise permutation of itself.

MPROVE – Iteratively improves the solution vector, x , of a linear set of equations, $Ax = b$. (You must call the LUDCMP procedure before calling MPROVE.)

NCHSK – Computes binomial coefficients, often stated in words such as “ N choose K ”, which is the number of ways k elements can be drawn from a set of n elements, $k < n$, where the ordering of the k elements does not matter.

NORMALS – Computes unit normals on a parametrically defined surface.

POLY – Evaluates a polynomial function of a variable.

POLY_AREA – Returns the area of an n -sided polygon, given the vertices of the polygon.

POLY_FIT – Fits an n -degree polynomial curve through a set of data points using the least-squares method.

POLYFITW – Fits an n -degree polynomial curve through a set of data points using the least-squares method.

PRIME – Returns all positive primes less than or equal to a scalar input.

PRODUCT – Returns the product of all elements in an array.

QUANTILE – Computes quantiles in a distribution.

RANDOMN – Returns one or more normally distributed floating-point pseudo-random numbers with a mean of zero and a standard deviation of 1.

RANDOMU – Returns one or more uniformly distributed floating-point pseudo-random numbers over the range $0 < Y < 1.0$.

REGRESS – Fits a curve to data using the multiple linear regression method.

SIGMA – Calculates the standard deviation value of an array. (Optionally, it can also calculate the standard deviation over one dimension of an array as a function of the other dimensions.)

SMALL_INT – Smallest Integer Function. Standard Library function that returns the smallest integer greater than or equal to the passed value. Also known as Ceiling Function.

SOBEL – Performs a Sobel edge enhancement of an image.

SPLINE – Performs a cubic spline interpolation.

STDEV – Computes the standard deviation and mean of the input array.

SUM – Sums an array of n dimensions over one of its dimensions.

SURFACE_FIT – Determines the polynomial fit to a surface.

SVBKS – Uses “back substitution” to solve the set of simultaneous linear equations $Ax = b$, given the u , w , and v arrays created by the SVD procedure from the matrix a .

SVD – Performs singular value decomposition on a matrix.

SVDFIT – Fits polynomial function to data using the singular value decomposition method of least-squares curve fitting.

General Mathematics Functions (cont.)

TOTAL – Sums the elements of an input array.

TQLI – Uses the QL algorithm with implicit shifts to determine the eigenvalues and eigenvectors of a real, symmetric, tridiagonal matrix.

TRED2 – Reduces a real, symmetric matrix to tridiagonal form, using Householder's method.

TRIDAG – Solves tridiagonal systems of linear equations.

ZROOTS – Finds the roots of the m-degree complex polynomial, using Laguerre's method.

OPTIMIZATION AND REGRESSION ROUTINES

CURVEFIT – Performs a nonlinear least-squares fit to a function of an arbitrary number of parameters.

EFFICIENT – Calculates the efficient frontier (a set of optimal portfolios).

GAUSSFIT – Fits a Gaussian curve through a data set.

MINIMIZE – Minimizes a real valued function of n real variables.

POLY_FIT – Fits an n-degree polynomial curve through a set of data points using the least-squares method.

POLYFITW – Fits an n-degree polynomial curve through a set of data points using the least-squares method.

REGRESS – Fits a curve to data using the multiple linear regression method.

SVDFIT – Fits a polynomial function to data using the singular value decomposition method of least-squares curve fitting.

SPECIAL MATHEMATICAL FUNCTIONS

BESELI – Calculates the Bessel I function for the input parameter.

BESELJ – Calculates the Bessel J function for the input parameter.

BESELY – Calculates the Bessel Y function for the input parameter.

ERRORF – Calculates the standard error function of the input variable.

GAMMA – Calculates the gamma function of the input variable.

GAUSSINT – Evaluates the integral of the Gaussian probability function.

TRANSCENDENTAL MATHEMATICAL FUNCTIONS

ACOS – Returns the arc-cosine of x.

ALOG – Returns the natural logarithm of x.

ALOG10 – Returns the logarithm to the base 10 of x.

ASIN – Returns the arcsine of x.

ATAN – Calculates the arctangent of the input variable (s).

COS – Calculates the cosine of the input variable.

COSH – Calculates the hyperbolic cosine of the input variable.

Transcendental Mathematical Functions (cont.)

EXP – Raises e to the power of the value of the input variable.

SIN – Returns the sine of the input variable.

SINH – Returns the hyperbolic sine of the input variable.

SQRT – Calculates the square root of the input variable.

TAN – Returns the tangent of the input variable.

TANH – Returns the hyperbolic tangent of the input variable.

PROGRAMMING

DATA CONVERSION ROUTINES

BYTE – Converts an expression to byte data type.

BYTSCL – Scales and converts an array to byte data type.

COMPLEX – Converts an expression to complex data type.

DCOMPLEX – Converts an expression to double-precision complex data type.

DOUBLE – Converts an expression to double-precision floating-point data type.

FIX – Converts an expression to integer data type.

FLOAT – Converts an expression to single-precision floating-point data type.

INT32 – Converts an expression to longword integer data type.

LONG – Converts an expression to longword integer data type.

NINT – Converts input to the nearest integer.

SCL – Emulates BYTSCL function but for all data types.

SCLNONLIN – Performs general scaling of array values (generalizes SCL).

STRING – Converts the input parameters to characters and returns a string expression.

DATA EXTRACTION ROUTINES

BYTE – Extracts data from an expression and places it in a byte scalar or array.

COMPLEX – Extracts data from an expression and places it in a complex scalar or array.

DCOMPLEX – Extracts data from an expression and places it in a complex scalar or array.

DOUBLE – Extracts data from an expression and places it in a double-precision floating-point scalar or array.

FIX – Extracts data from an expression and places it in an integer scalar or array.

FLOAT – Extracts data from an expression and places it in a single-precision floating-point scalar or array.

INT32 – Converts an expression to longword integer data type.

LONG – Extracts data from an expression and places it in a longword integer scalar or array.

DATE/TIME FUNCTIONS

CREATE_HOLIDAYS – Creates the system variable !Holiday_List.

CREATE_WEEKENDS – Creates the system variable !Weekend_List.

DAY_NAME – Returns a string array containing the name of the day of the week for each day in a Date/Time variable.

DAY_OF_WEEK – Returns an array of integers containing the day of the week for each date in a Date/Time variable.

DAY_OF_YEAR – Returns an array of integers containing the day of the year for each date in a Date/Time variable.

DT_ADD – Increment the values in a Date/Time variable by a specified amount.

DT_COMPRESS – Removes holidays and weekends from the Julian day portion of Date/Time variables.

DT_DURATION – Determines the elapsed time between two Date/Time variables.

DT_PRINT – Prints the values of PV-WAVE Date/Time variables in a readable manner.

DT_SUBTRACT – Decrements the values in a Date/Time variable by a specified amount.

DT_TO_SEC – Converts PV-WAVE Date/Time variables to double-precision variables containing the number of seconds elapsed from a base date.

DT_TO_STR – Converts PV-WAVE Date/Time variables into string data.

DT_TO_VAR – Converts a PV-WAVE Date/Time variable to regular numerical data.

DTGEN – Returns an array of PV-WAVE Date/Time variables beginning from a specified date and incremented by a specified amount.

JUL_TO_DT – Converts a Julian day number to a PV-WAVE Date/Time variable.

LOAD_HOLIDAYS – Passes the value of the !Holiday_List system variable to the Date/Time routines.

LOAD_WEEKENDS – Passes the value of the !Weekend_List system variable to the Date/Time routines.

MONTH_NAME – Returns a string or array of strings containing the names of the months contained in a Date/Time variable.

SEC_TO_DT – Converts any number of seconds into PV-WAVE Date/Time variables.

STR_TO_DT – Converts date and time string data to PV-WAVE Date/Time variables.

TODAY – Returns a Date/Time variable containing the current system date and time.

VAR_TO_DT – Converts scalars or arrays of scalars representing dates and times into PV-WAVE Date/Time variables.

FILE MANIPULATION ROUTINES

CLOSE – Closes the specified file units.

EOF – Tests the specified file unit for the end-of-file condition.

FINDFILE – Returns a string array containing the names of all files matching a specified file description.

FLUSH – Causes all buffered output on the specified file units to be written.

FREE_LUN – Deallocates file units previously allocated with GET_LUN.

FSTAT – Returns an expression containing status information about a specified file unit.

GET_LUN – Allocates a file unit from a pool of free units.

LFPOINT_LUN – Allows the current position of the specified file to be set to any arbitrary point in the file. This routine is for large files (>2.5 gigabytes on supported platforms only).

OPENR – Opens an existing file for input only.

OPENU – Opens an existing file for input and output.

OPENURL – Opens a file on the internet to be accessed using PV-WAVE.

OPENW – Opens a new file for input and output.

PARSEFILENAME – Extracts specified parts of a full file pathname.

POINT_LUN – Allows the current position of the specified file to be set to any arbitrary point in the file.

HELP AND INFORMATION ROUTINES

HELP – Starts the online help system.

INFO – Displays information on many aspects of the current PV-WAVE session.

PWD – Prints the current default directory.

OPERATING SYSTEM ACCESS ROUTINES

CALL_UNIX – (UNIX Only) Lets a PV-WAVE procedure communicate with an external routine written in C.

CD – Changes the current working directory.

DELETE_SYMBOL – (OpenVMS Only) Deletes a DCL (Digital Command Language) interpreter symbol from the current process.

DEL_FILE – Deletes a specified file on your system.

DELLOG – (OpenVMS Only) Deletes a logical name.

ENVIRONMENT – (UNIX Only) Returns a string array containing all the UNIX environment strings for the PV-WAVE process.

GETENV – Returns the specified equivalence string from the environment of the PV-WAVE process.

GET_SYMBOL – (OpenVMS Only) Returns the value of an OpenVMS DCL interpreter symbol as a scalar string.

LINKNLOAD – Provides simplified access to external routines in sharable images.

Operating System Access Routines (cont.)

POPD – Pops a directory from the top of a last-in, first-out directory stack.

PRINTD – Lists the directories located in the directory stack, and the current working directory.

PUSHD – Pushes a directory onto the top of a last-in, first-out directory stack.

SETENV – (UNIX Only) Adds or changes an environment string in the process environment.

SETLOG – (OpenVMS Only) Defines a logical name.

SET_SYMBOL – (OpenVMS Only) Defines a DCL interpreter symbol for the current process.

SPAWN – Spawns a child process to execute a given command.

SYSTIME – Returns the current system time as either a string or as the number of seconds elapsed since January 1, 1970.

TRNLOG – (OpenVMS Only) Searches the OpenVMS name tables for a specified logical name and returns the equivalence string (s) in a PV-WAVE variable.

WEOF – (OpenVMS Only) Writes an end-of-file mark on the designated unit at the current position.

WIN32_locale – Creates a window for the display of graphics or text.

WIN32_PICK_PRINTER – Displays a Windows printer dialog.

WIN32_PICK_FONT – Displays a Windows common font dialog.

PROGRAMMING ROUTINES

ADDVAR – Creates a variable on the \$MAIN\$ program level and binds a local variable to it.

BREAKPOINT – Lets you insert and remove breakpoints in programs for debugging.

CHECKFILE – Determines if a file can be read from or written to.

CHECK_MATH – Returns and clears the accumulated math error status.

DEFINE_KEY – Programs a keyboard function key with a string value, or with a specified action.

DEFSYSV – Creates a new system variable initialized to the specified value.

DELFUNC – Deletes one or more compiled functions from memory.

DELPROC – Deletes one or more compiled procedures from memory.

DELSTRUCT – Deletes one or more named structure definitions from memory.

DELVAR – Deletes variables from the main program level.

ERRCHECK – Provides comprehensive error checking that can be used by any PV-WAVE routine.

EXIT – Exits PV-WAVE and returns you to the operating system.

FINITE – Returns a value indicating if the input variable is finite or not.

HAK – Implement a “hit any key to continue” function.

KEYWORD_SET – Tests if an input expression has a nonzero value.

Programming Routines (cont.)

MESSAGE – Issues error and informational messages using the same mechanism employed by built-in PV-WAVE routines.

N_ELEMENTS – Returns the number of elements contained in any expression or variable.

N_PARAMS – Returns the number of non-keyword parameters used in calling a PV-WAVE procedure or function.

N_TAGS – Returns the number of structure tags contained in any expression.

OMPTUNE – Tests and records platform-specific parallel-performance for a representative subset of all possible PV-WAVE array operations.

ON_ERROR – Determines the action taken when an error is detected inside a PV-WAVE user-written procedure or function.

ON_ERROR_GOTO – Specifies a statement to jump to if an error occurs in the current procedure.

ON_IOERROR – Specifies a statement to jump to if an I/O error occurs in the current procedure.

PARAM_PRESENT – Tests if a parameter was actually present in the call to a procedure or function.

PARSEFILENAME – Extracts specified parts of a full file pathname.

RENAME – Renames a PV-WAVE variable.

RETAILL – Issues RETURNS from nested routines. Used primarily to recover from errors in user-written procedures and functions.

RETURN – Returns control to the caller of a user-written procedure or function.

SAME – Tests if two variables are the same.

SET_OMP – Sets the run-time OpenMP parameters for shared-memory, multi-processor machines running supported operating systems.

SIZE – Returns a vector containing size and type information for the given expression.

STOP – Stops the execution of a running program or batch file, and returns control to the interactive mode.

STRMESSAGE – Returns the text of the error message specified by the input error number.

STRUCTINFO – Gathers information about the tags of a PV-WAVE structure variable.

STRUCTNAME – Returns the name of a structure if the input variable name is of type structure.

STRUCTREF – Returns a list of all existing references to a structure.

TAG_NAMES – Returns a string array containing the names of the tags in a structure expression.

UPVAR – Accesses a variable that is not on the current program level.

VAR_MATCH – Scans for PV-WAVE variables that match the given criteria.

WAIT – Suspends execution of a PV-WAVE program for a specified period.

SESSION ROUTINES

INFO – Displays information on many aspects of the current PV-WAVE session.

JOURNAL – Provides a record of an interactive session by saving in a file all text entered from the terminal in response to a prompt.

RESTORE – Restores the PV-WAVE objects saved in a file by the SAVE procedure.

SAVE – Saves variables in a file for later recovery by RESTORE.

STRING PROCESSING ROUTINES

STRCOMPRESS – Compresses the white space in an input string.

STR_IS_NUM – Determines if a string contains a numerical value.

STRJOIN – Concatenates all of the elements of a string array into a single scalar string.

STRLEN – Returns the length of the input parameter.

STRLOOKUP – Queries, creates, saves, or modifies a string server database.

STRLOWCASE – Converts a copy of the input string to lowercase letters.

STRMATCH – Matches a specified string to an existing regular expression.

STRMID – Extracts a substring from a string expression.

STRPOS – Searches for the occurrence of a substring within an object string, and returns its position.

STRPUT – Inserts the contents of one string into another.

STRSPLIT – Splits a string into an array of tokens (substrings).

STRSUBST – Performs string substitution (search and replace).

STRTRIM – Removes extra blank spaces from an input string.

STRUPCASE – Converts a copy of the input string to uppercase letters.

TABLE MANIPULATION FUNCTIONS

BUILD_TABLE – Creates a table from one or more vectors (one-dimensional arrays).

GROUP_BY – Performs summary (aggregate) functions to groups of rows in a PV-WAVE table variable.

ORDER_BY – Sorts the rows in a PV-WAVE table variable to create a new table.

QUERY_TABLE – Subsets a table created with the BUILD_TABLE function.

UNIQUE – Returns a vector (one-dimensional array) containing the unique elements from another vector variable.

RENDERING TECHNIQUES

3D VISUALIZATION TOOLKIT (VTK) ROUTINES

VtkADDATTRIBUTE – Collects point attributes for VTK datasets.

VtkAXES – Creates a set of axes.

VtkBAR – Creates a plot of 3D cylinders.

VtkCAMERA – Changes the camera's parameters.

VtkCOLORBAR – Adds a color bar legend to a VTK scene using the current PV-WAVE color table.

VtkCLOSE – Closes the VTK process.

VtkCOMMAND – Sends Tcl and VTK commands to the Tcl process.

vtkERASE – Erases the contents of the current VTK window.

VtkGETNAMES – Returns the names of all Lights, Cameras and Actors in a renderer.

vtkGRID – Adds 3D grid lines to a VTK scene.

VtkHEDGEHOG – Creates a HedgeHog (vector) plot.

VtkINIT – Initializes the VTK system.

VtkISOSURFACE – Displays the requested iso surfaces of a volume.

VtkLIGHT – Adds a light to a VTK window.

VtkPLOTS – Adds a polyline.

VtkPOLYDATA – Passes vertex/polygon lists, lines, points, and triangles to VTK.

VtkPOLYSHADE – Renders a polygon object.

vtkPPMREAD – Reads a PPM file.

vtkPPMWRITE – Writes the contents of a VTK window to a PPM file.

VtkRECTILINEARGRID – Passes data describing a Rectilinear Grid to VTK.

vtkRENDERWINDOW – Renders a VTK window.

VtkSCATTER – Renders 3D points.

vtkSHOW3 – Shows a 2D array three ways in a display that combines vtkSURFACE, "CONTOUR", and an image map.

VtkSLICEVOL – Creates a sliced 3D volume at specific x, y, z locations.

VtkSTRUCTUREDGRID – Passes data describing a structured grid to VTK.

VtkSTRUCTUREDPOINTS – Passes data describing structured points to VTK.

VtkSURFACE – Renders a surface.

VtkSURFGEN – Generates a 3D surface from sampled points assumed to lie on a surface.

VtkTEXT – Adds a text string.

vtkTVRD – Returns the contents of a VTK window as a bitmapped image.

VtkUNSTRUCTUREDGRID – Passes data describing an unstructured grid to VTK.

vtkWDELETE – Closes a VTK window, however it does not shut down the Tcl process.

vtkWINDOW – Creates a VTK window.

VtkWRITEVRML – Creates a Virtual Reality Modeling Language file (VRML .wrl file) from a scene in a VTK window.

vtkWSET – Sets the active VTK window.

POLYGON GENERATION ROUTINES

POLY_SPHERE – Generates the vertex list and polygon list that represent a sphere.

POLY_SURF – Generates a 3D vertex list and a polygon list, given a 2D array containing Z values.

SHADE_VOLUME – Given a 3D volume and a contour value, produces a list of vertices and polygons describing the contour surface (also known as an iso-surface).

POLYGON MANIPULATION ROUTINES

POLY_C_CONV – Returns a list of colors for each polygon, given a polygon list and a list of colors for each vertex.

POLY_COUNT – Returns the total number of polygons contained in a polygon list.

POLY_MERGE – Merges two vertex lists and two polygon lists together so that they can be rendered in a single pass.

POLYGON RENDERING TECHNIQUES

MOLEC – Creates an image of a ball and stick molecular model.

POLY_PLOT – Renders a given list of polygons.

POLYSHADE – Constructs a shaded surface representation of one or more solids described by a set of polygons.

RENDER – Generates a ray-traced rendered image from one or more predefined objects.

RENDER24 – Generates a ray-traced rendered 24-bit image of m objects.

RAY TRACING ROUTINES

CONE – Defines a conic object that can be used by the RENDER function.

CYLINDER – Defines a cylindrical object that can be used by the RENDER function.

MESH – Defines a polygonal mesh object that can be used by the RENDER function.

RENDER – Generates a ray-traced rendered image from one or more predefined objects.

RENDER24 – Generates a ray-traced rendered 24-bit image of m objects.

SPHERE – Defines a spherical object that can be used by the RENDER function.

VOLUME – Defines the volumetric data that can be used by the RENDER function.

VOLUME MANIPULATION ROUTINES

AFFINE – Applies an affine transformation to an array.

BLOB – Isolates a homogeneous region in an array.

BLOBCOUNT – Counts homogeneous regions in an array.

BOUNDARY – Computes the boundary of a region in an array.

DERIVN – Differentiates a function represented by an array.

MOMENT – Computes moments of an array.

NEIGHBORS – Finds the neighbors of specified array elements.

RESAMP – Resamples an array to new dimensions.

SLICE_VOL – Returns a 2D array containing a slice from a 3D volumetric array.

VOL_PAD – Returns a 3D volume of data padded on all six sides with zeroes.

VOL_TRANS – Returns a 3D volume of data transformed by a 4-by-4 matrix.

VOLUME RENDERING ROUTINES

RENDER – Generates a ray-traced rendered image from one or more predefined objects.

VECTOR_FIELD3 – Plots a 3D vector field from three arrays.

VOL_MARKER – Displays colored markers scattered throughout a volume.

VOL_REND – Renders volumetric data in a translucent manner.

LINEAR SYSTEMS

MATRIX INVERSION

INV – General matrix inversion

LINEAR EQUATIONS WITH FULL MATRICES

LUSOL – Systems involving general matrices

LUFAC – LU factorization of general matrices

CHSOL – Systems involving symmetric positive definite matrices

CHFAC – Factorization of symmetric positive definite matrices

LINEAR LEAST SQUARES WITH FULL MATRICES

QRSOL – Least-squares solution

QRFAC – Least-squares factorization

SVDCOMP – Singular Value Decomposition (SVD) and generalized inverse

CHNDSOL – Solve and generalized inverse for positive semidefinite matrices

CHNDFAC – Factor and generalized inverse for positive semidefinite matrices

LINLSQ – Solves a linear least-squares problem with linear constraints.

SPARSE MATRICES

SP_LUSOL – Solve a sparse system of linear equations $Ax = b$.

SP_LUFAC – Compute an LU factorization of a sparse matrix stored in either coordinate format or CSC format.

SP_BDSOL – Solve a general band system of linear equations $Ax = b$.

SP_BDFAC – Compute the LU factorization of a matrix stored in band storage mode.

Sparse Matrices (cont.)

SP_PDSOL – Solve a sparse symmetric positive definite system of linear equations $Ax = b$.

SP_PDFAC – Solve a sparse symmetric positive definite system of linear equations $Ax = b$.

SP_BDPDSOL – Solve a symmetric positive definite system of linear equations $Ax = b$ in band symmetric storage mode.

SP_BDPDFAC – Compute the RTR Cholesky factorization of symmetric positive definite matrix, A , in band symmetric storage mode.

SP_GMRES – Solve a linear system $Ax = b$ using the restarted generalized minimum residual (GMRES) method.

SP_CG – Solve a real symmetric definite linear system using a conjugate gradient method. Using keywords, a preconditioner can be supplied.

SP_MVMUL – Compute a matrix-vector product involving sparse matrix and a dense vector.

EIGENSYSTEM ANALYSIS

EIG – General and symmetric matrices

EIGSYMGEN – Real symmetric matrices and B positive definite

GENEIG – Computes the generalized eigenexpansion of a system $Ax = Bx$.

INTERPOLATION AND APPROXIMATION

CUBIC SPLINE INTERPOLATION

CSINTERP – Derivative end conditions

CSSHAPE – Shape preserving

B-SPLINE INTERPOLATION

BSINTERP – One-dimensional and two-dimensional interpolation

BSKNOTS – Knot sequence given interpolation data

B-SPLINE AND CUBIC SPLINE EVALUATION AND INTERPOLATION

SPVALUE – Evaluation and differentiation

SPINTEG – Integration

LEAST-SQUARES APPROXIMATION AND SMOOTHING

FCNLSQ – General functions

BSLSQ – Splines with fixed knots

CONLSQ – Constrained spline fit

CSSMOOTH – Cubic-smoothing spline

POLYEVAL – Evaluates a polynomial in n variables.

POLYFITN – Fits a polynomial to some n dimensional data.

WgSplineTool – Widget-based interface

SMOOTHDATA1D – Smooth one-dimensional data by error detection.

WENDCOEF – Computes the coefficients for the Wendland polynomial of type (d,k) .

SCATTERED DATA INTERPOLATION

SCAT2DINTERP – Akima's surface-fitting method

RADBF – Computes a fit using radial-basis functions

RADBE – Evaluates a radial-basis fit

RBFIMSCL – Multiscale radial basis interpolation for n -dimensional data.

QUADRATURE

CUBIC SPLINE INTERPOLATION

INTFCN – Integration of a user-defined univariate or bivariate function

INTFCNHYPER – Iterated integral on a hyper-rectangle

GQUAD – Gauss quadrature formulas

DERIV – Numerical differentiation using three-point Lagrangian

FCN_DERIV – Computes the first, second, or third derivative of a user-supplied function.

INTFCN_QMC – Integrates a function on a hyper-rectangle using a quasi-Monte Carlo method.

DIFFERENTIAL EQUATIONS

DEA_PETZOLD_GEAR – Solves a first order differential-algebraic system of equations, $g(t, y, y') = 0$, using the Petzold-Gear BDF method.

ODE – Adams-Gear or Runge-Kutta method.

ODEBV – Solves a general system of ordinary differential equations.

PDE_1D_MG – Solves a system of partial differential equations.

PDE_MOL – Solves a system of partial differential equations of the form $u_t = f(x, t, u, u_x, u_{xx})$ using the method of lines. The solution is represented with cubic Hermite polynomials.

POISSON2D – Solves Poisson's or Helmholtz's equation on a two-dimensional rectangle using a fast Poisson solver based on the HODIE finite-difference scheme on a uniform mesh.

TRANSFORMS

FFTCOMP – Real or complex FFT

FFTINIT – Real or complex FFT initialization

CONVOL1D – Compute discrete convolution

CORR1D – Compute discrete correlation

LAPLACE_INV – Computes the inverse Laplace transform of a complex function.

NONLINEAR EQUATIONS

ZEROPOLY – Zeros of a polynomial with real or complex coefficients

ZEROFCN – Real zeros of a function

ZEROSYS – Root of a system of equations using Powell's hybrid method

OPTIMIZATION

FMIN – Univariate function using function and possibly first derivative values

FMINV – Multivariate function using quasi-Newton method

NLINLSQ – Nonlinear least squares using Levenberg-Marquardt algorithm

Optimization (cont.)

LINEAR_PROGRAMMING – Minimizes the Euclidean scalar product $\langle c, x \rangle$ on R^n , for fixed c , linear constraints $bl \leq Ax \leq bu$, and bounds $xl \leq x \leq xu$

LINPROG – Dense linear programming

QUADPROG – Quadratic programming

NONLINPROG – Using successive quadratic programming method

MINCONGEN – Minimizes a general objective function subject to linear equality/inequality constraints.

CONSTRAINED_NLP – Solves a general nonlinear programming problem using a sequential equality constrained quadratic programming method.

SPECIAL FUNCTIONS

ERROR FUNCTIONS

BETA – Beta function

BETAI – Incomplete beta function

ERF – Error function

ERFC – Complementary error function

ERFCE – Evaluates the exponentially scaled complementary error function

ERFE – Evaluates a scaled function related to $ERFC(z)$

LN BETA – Logarithmic beta function

GAMMA FUNCTIONS

GAMMA – Gamma function

LNGAMMA – Logarithmic gamma function

GAMMAI – Incomplete gamma function

BESSEL FUNCTIONS WITH REAL ORDER AND COMPLEX ARGUMENT

BESSI – Modified Bessel function of the first kind

BESSJ – Bessel function of the first kind

BESSK – Modified Bessel function of the second kind

BESSY – Bessel function of the second kind

BESSI_EXP – Evaluates the exponentially scaled modified Bessel function of the first kind of orders zero and one.

BESSK_EXP – Evaluates the exponentially scaled modified Bessel function of the third kind of orders zero and one.

ELLIPTICAL INTEGRALS

ELE – Evaluates the complete elliptic integral of the second kind $E(x)$.

ELK – Evaluates the complete elliptic integral of the kind $K(x)$.

ELRF – Evaluates Carlson's elliptic integral of the first kind $RF(x, y, z)$.

ELRD – Evaluates Carlson's elliptic integral of the second kind $RD(x, y, z)$.

ELRJ – Evaluates Carlson's elliptic integral of the third kind $RJ(x, y, z, r)$.

ELRC – Evaluates an elementary integral from which inverse circular functions, logarithms and inverse hyperbolic functions can be computed.

FRESNEL INTEGRALS

FRESNEL_COSINE – Evaluates the cosine Fresnel integral.

FRESNEL_SINE – Evaluates the sine Fresnel integral.

AIRY FUNCTIONS

AIRY_AI – Evaluates the Airy function.

AIRY_BI – Evaluates the Airy function of the second kind.

HERMITE FUNCTIONS

HERMITE – Evaluates Hermite polynomials.

KELVIN FUNCTIONS

KELVIN_BERO – Evaluates the Kelvin function of the first kind, ber , of order zero.

KELVIN_KERO – Evaluates the Kelvin function of the second kind, ker , of order zero.

KELVIN_BEIO – Evaluates the Kelvin function of the first kind, bei , of order zero.

KELVIN_KEIO – Evaluates the Kelvin function of the second kind, kei , of order zero.

LEGENDRE FUNCTIONS

LEGENDRE – Evaluates the associated Legendre functions.

BASIC FINANCIAL FUNCTIONS

CUM_INTR – Evaluates the cumulative interest paid between two periods.

DEPRECIATION_DDB – Evaluates the depreciation of an asset using the double-declining balance method.

CUM_PRINC – Evaluates the cumulative principal paid between two periods.

DEPRECIATION_SLN – Evaluates the depreciation of an asset using the straight-line method.

DEPRECIATION_DB – Evaluates the depreciation of an asset using the fixed-declining balance method.

DEPRECIATION_SYD – Evaluates the depreciation of an asset using the sum-of-years digits method.

Basic Financial Functions (cont.)

DEPRECIATION_VDB – Evaluates the depreciation of an asset for any given period using the variable-declining balance method.

DOLLAR_DECIMAL – Converts a fractional price to a decimal price.

DOLLAR_FRACTION – Converts a decimal price to a fractional price.

EFFECTIVE_RATE – Evaluates effective annual interest rate.

FUTURE_VALUE – Evaluates the future value of an investment.

FUTURE_VAL_SCHD – Evaluates the future value of an initial principal taking into consideration a schedule of compound interest rates.

INT_PAYMENT – Evaluates the interest payment for an investment for a given period.

INT_RATE_ANNUITY – Evaluates the interest rate per period of an annuity.

INT_RATE_RETURN – Evaluates the internal rate of return for a schedule of cash flows.

INT_RATE_SCHD – Evaluates the internal rate of return for a schedule of cash flows. It is not necessary that the cash flows be periodic.

MOD_INTERN_RATE – Evaluates the modified internal rate of return for a schedule of periodic cash flows.

NET_PRES_VALUE – Evaluates the net present value of a stream of unequal periodic cash flows, which are subject to a given discount rate.

NOMINAL_RATE – Evaluates the nominal annual interest rate.

NUM_PERIODS – Evaluates the number of periods for an investment for which periodic and constant payments are made and the interest rate is constant.

PAYMENT – Evaluates the periodic payment for an investment.

PRESENT_VALUE – Evaluates the net present value of a stream of equal periodic cash flows, which are subject to a given discount rate.

PRES_VAL_SCHD – Evaluates the present value for a schedule of cash flows. It is not necessary that the cash flows be periodic.

PRINC_PAYMENT – Evaluates the payment on the principal for a specified period.

BOND FUNCTIONS

ACCR_INT_MAT – Evaluates the interest which has accrued on a security that pays interest at maturity.

ACCR_INT_PER – Evaluates the interest which has accrued on a security that pays interest periodically.

BOND_EQV_YIELD – Evaluates the bond-equivalent yield of a Treasury bill.

CONVEXITY – Evaluates the convexity for a security.

COUPON_DAYS – Evaluates the number of days in the coupon period containing the settlement date.

COUPON_NUM – Evaluates the number of coupons payable between the settlement date and the maturity date.

SETTLEMENT_DB – Evaluates the number of days starting with the beginning of the coupon period and ending with the settlement date.

Bond Functions (cont.)

COUPON_DNC – Evaluates the number of days starting with the settlement date and ending with the next coupon date.

DEPREC_AMORDEGRC – Evaluates the depreciation for each accounting period. During the evaluation of the function a depreciation coefficient based on the asset life is applied.

DEPREC_AMORLINC – Evaluates the depreciation for each accounting period.

DISCOUNT_PR – Evaluates the price of a security sold for less than its face value.

DISCOUNT_RT – Evaluates the interest rate implied when a security is sold for less than its value at maturity in lieu of interest payments.

DISCOUNT_YLD – Evaluates the annual yield of a discounted security.

DURATION – Evaluates the annual duration of a security, where the security has periodic interest payments.

INT_RATE_SEC – Evaluates the interest rate of a fully invested security.

DURATION_MAC – Evaluates the modified Macauley duration of a security.

COUPON_NCD – Evaluates the first coupon date which follows the settlement date.

COUPON_PCD – Evaluates the coupon date which immediately precedes the settlement date.

PRICE_PERIODIC – Evaluates the price, per \$100 face value, of a security that pays periodic interest.

PRICE_MATURITY – Evaluates the price, per \$100 face value, of a security that pays interest at maturity.

MATURITY_REC – Evaluates the amount one receives when a fully invested security reaches the maturity date.

TBILL_PRICE – Evaluates the price per \$100 face value of a Treasury bill.

TBILL_YIELD – Evaluates the yield of a Treasury bill.

YEAR_FRACTION – Evaluates the fraction of a year represented by the number of whole days between two dates.

YIELD_MATURITY – Evaluates the annual yield of a security that pays interest at maturity.

YIELD_PERIODIC – Evaluates the yield of a security that pays periodic interest.

BASIC STATISTICS AND RANDOM NUMBER GENERATION

CHISQTEST – Chi-squared goodness-of-fit test

FAURE_INIT – Initializes the structure used for computing a shuffled Faure sequence.

FAURE_NEXT_PT – Generates a shuffled Faure sequence.

FREQTABLE – Tallies observations into a one-way frequency table

RANDOMOPT – Control of the random number seed and uniform(0,1) generator

Basic Statistics and Random Number Generation (cont.)

RANDOM – Generates pseudorandom numbers

RANKS – Ranks, normal scores, or exponential scores

PROBABILITY DISTRIBUTION FUNCTIONS AND INVERSES

NORMALCDF – Normal (Gaussian) distribution function

GAMMACDF – Gamma distribution function

BINORMALCDF – Bivariate normal distribution

BETACDF – Beta distribution function

CHISQCDF – Chi-squared distribution function

BINOMIALCDF – Binomial distribution function

FCDF – F distribution function

HYPERGEOCDF – Hypergeometric distribution function

TCDF – Student's t distribution function

POISSONCDF – Poisson distribution function

UTILITIES

DAYSTODATE – Days since epoch to date

MATRIX_NORM – Computes various norms of a rectangular matrix, a matrix stored in band format, and a matrix stored in coordinate format.

DATETODAYS – Date to days since epoch

NANFINDER – Finds all instances of nan, inf, -inf in an array.

CMAST_ERR_PRINT – Sets options for error printing in Math and Stat options.

CONSTANT – Natural and mathematical constants

MACHINE – Machine constants

CMAST_ERR_STOP – Sets options for error recovery in Math and Stat options.

NORM – Vector norms

CMAST_ERR_TRANS – Informational Error codes for routine

BASIC STATISTICS

SIMPLE SUMMARY STATISTICS

SIMPLESTAT – Univariate summary statistics

NORM1SAMP – Mean and variance inference for a single normal population

NORM2SAMP – Inferences for two normal populations

TABULATE, SORT, AND RANK

FREQTABLE – Tallies observations into a one-way frequency table

SORTDATA – Sorts data with options to tally cases into a multiway frequency table

RANKS – Ranks, normal scores, or exponential scores

REGRESSION

MULTIPLE LINEAR REGRESSION

REGRESSORS – Generates regressors for a general linear model

MULTIREGRESS – Fits a multiple linear regression model and optionally produces summary statistics for a regression model

MULTIPREDICT – Computes predicted values, confidence intervals, and diagnostics

VARIABLE SELECTION

ALLBEST – All best regressions

STEPWISE – Stepwise regression

POLYNOMIAL AND NONLINEAR REGRESSION

POLYREGRESS – Fits a polynomial regression model

POLYPREDICT – Computes predicted values, confidence intervals, and diagnostics

NONLINREGRESS – Fits a nonlinear regression model.

INFERENCE AND DIAGNOSTICS

HYPOTH_PARTIAL – Constructs an equivalent completely testable multivariate general linear hypothesis $HbU = G$ from a partially testable hypothesis $HpbU = Gp$.

HYPOTH_SCPH – Computes the matrix of sums of squares and cross products for the multivariate general linear hypothesis $HbU = G$ given the regression fit.

HYPOTH_TEST – Performs tests for a multivariate general linear hypothesis $HbU = G$ given the hypothesis sums of squares and cross products matrix SH .

POLYNOMIAL AND NONLINEAR REGRESSION

NONLINOPT – Fit a nonlinear regression model using Powell's algorithm.

ALTERNATIVES TO LEAST SQUARES REGRESSION

LNORMREGRESS – LAV, Lpnorm, and LMV criteria regression

CORRELATION AND COVARIANCE

COVARIANCES – Variance-covariance or correlation matrix.

PARTIAL_COV – Partial correlations and covariances.

POOLED_COV – Pooled covariance matrix.

ROBUST_COV – Robust estimate of covariance matrix.

ANALYSIS OF VARIANCE

ANOVA1 – Analyzes a one-way classification model

ANOVABALANCED – Balanced fixed, random, or mixed model

ANOVAFACT – Analyzes a balanced factorial design with fixed effects

ANOVANESTED – Nested random mode

CRD_FACTORIAL – Analyzes data from balanced and unbalanced completely randomized experiments.

HOMOGENEITY – Conducts Bartlett's and Levene's tests of the homogeneity of variance assumption in analysis of variance.

LATTICE_DESIGN – Analyzes balanced and partially-balanced lattice experiments.

LATIN_SQUARE – Analyzes data from latin-square experiments.

MULTICOMP – Performs Student-Newman-Keuls multiple comparisons test.

RCBD_FACTORIAL – Analyzes data from balanced and unbalanced randomized complete block experiments.

SPLIT_PLOT – Analyzes a wide variety of split-plot experiments with fixed, mixed or random factors.

STRIP_PLOT – Analyzes data from strip-plot experiments.

SPLIT_SPLIT_PLOT – Analyzes data from split-split-plot experiments.

STRIP_SPLIT_PLOT – Analyzes data from strip-split-plot experiments.

YATES – Estimates missing observations in designed experiments using Yate's method.

CATEGORICAL AND DISCRETE DATA ANALYSIS

CONTINGENCY – Two-way contingency table analysis

EXACT_ENUM – Exact probabilities in a table; total enumeration.

EXACT_NETWORK – Exact probabilities in a table.

CAT_GLM – Generalized linear models.

NONPARAMETRIC STATISTICS

SIGNTEST – Sign test

WILCOXON – Wilcoxon rank sum test

NCTRENDS – Noether's test for cyclical trend.

CSTRENDS – Cox and Stuarts' sign test for trends in location and dispersion.

TIE_STATS – Tie statistics.

KW_TEST – Kruskal-Wallis test.

FRIEDMANS_TEST – Friedman's test.

COCHRANQ – Cochran's Q test.

KTRENDS – K-sample trends test.

GOODNESS OF FIT

CHISQTEST – Chi-squared goodness-of-fit test.

NORMALITY – Shapiro-Wilk W test for normality.

KOLMOGOROV1 – One-sample continuous data Kolmogorov-Smirnov.

KOLMOGOROV2 – Two-sample continuous data Kolmogorov-Smirnov.

MVAR_NORMALITY – Mardia's test for multivariate normality.

RANDOMNESS_TEST – Performs runs up test, pairs test, d2 test or triplets test.

TIME SERIES AND FORECASTING

ARMA – Computes least-squares or method-of-moments estimates of parameters and optionally computes forecasts and their associated probability limits.

AUTOCORRELATION – Sample autocorrelation function.

AUTO_UNI_AR – Automatic selection and fitting of a univariate autoregressive time series model.

AUTO_ARIMA – Automatically identifies time series outliers, determines parameters of a multiplicative seasonal

ARIMA model, and produces forecasts that incorporate the effects of outliers whose effects persist beyond the end of the series.

BOXCOXTRANS – Perform a Box-Cox transformation.

CROSSCORRELATION – Computes the sample cross-correlation function of two stationary time series.

Time Series and Forecasting (cont.)

DIFFERENCE – Performs differencing on a time series.

ESTIMATE_MISSING – Estimates missing values in a time series.

GARCH – Compute estimates of the parameters of a GARCH(p,q) model.

KALMAN – Performs Kalman filtering and evaluates the likelihood function for the state-space model.

LACK_OF_FIT – Lack-of-fit test based on the correlation function.

MAX_ARMA – Exact maximum likelihood estimation of the parameters in a univariate ARMA (autoregressive, moving average) time series model.

MULTI_CROSS – Computes the multichannel cross-correlation function of two mutually stationary multichannel time series.

PARTIAL_AC – Sample partial autocorrelation function.

SEASONAL_FIT – Estimates the optimum seasonality parameters for a time series using an autoregressive model, AR(p), to represent the time series.

TS_OUTLIER_FORECAST – Computes forecasts, their associated probability limits and y weights for an outlier contaminated time series whose underlying outlier free series follows a general seasonal or nonseasonal ARMA model.

TS_OUTLIER_IDENTIFICATION – Detects and determines outliers and simultaneously estimates the model parameters in a time series whose underlying outlier free series follows a general seasonal or nonseasonal ARMA model.

MULTIVARIATE ANALYSIS

CLUSTER_HIERARCHICAL – Performs a hierarchical cluster analysis given a distance matrix.

CLUSTER_NUMBER – Computes cluster membership for a hierarchical cluster tree.

DISCR_ANALYSIS – Perform discriminant function analysis.

DISSIMILARITIES – Computes a matrix of dissimilarities (or similarities) between the columns (or rows) of a matrix.

FACTOR_ANALYSIS – Extracts factor-loading estimates.

K_MEANS – Performs a K-means (centroid) cluster analysis.

PRINC_COMP – Computes principal components.

SURVIVAL ANALYSIS

KAPLAN_MEIER_ESTIMATES – Computes Kaplan-Meier estimates of survival probabilities in stratified samples.

LIFE_TABLES – Produces population and cohort life tables.

NONPARAM_HAZARD_RATE – Performs nonparametric hazard rate estimation using kernel functions and quasi-likelihoods.

PROP_HAZARDS_GEN_LIN – Analyzes survival and reliability data using Cox's proportional hazards model.

SURVIVAL_GLM – Analyzes survival data using a generalized linear model and estimates using various parametric modes.

PROBABILITY DISTRIBUTION FUNCTIONS AND INVERSES

BETACDF – Beta distribution function.

BINOMIALCDF – Binomial distribution function.

BINOMIALPDF – Evaluates the binomial probability function.

BINORMALCDF – Bivariate normal distribution.

CHISQCDF – Chi-squared distribution function.

FCDF – F distribution function.

GAMMACDF – Gamma distribution function.

HYPERGEOCDF – Hypergeometric distribution function.

HYPERGEOPDF – Evaluates the hypergeometric probability function.

NORMALCDF – Normal (Gaussian) distribution function.

POISSONCDF – Poisson distribution function.

POISSONPDF – Evaluates the probability function of a Poisson random variable with parameter theta.

TCDF – Student's t distribution function.

RANDOM NUMBER GENERATION

RANDOM NUMBERS

CONT_TABLE – Sets up a table to generate pseudorandom numbers from a general continuous distribution.

DISCR_TABLE – Sets up a table to generate pseudorandom numbers from a general discrete distribution.

FAURE_INIT – Initializes the structure used for computing a shuffled Faure sequence.

FAURE_NEXT_PT – Generates a shuffled Faure sequence.

RAND_FROM_DATA – Generates pseudorandom numbers from multivariate distribution determined from a given sample.

RAND_GEN_CONT – Generates pseudorandom numbers from a general continuous distribution.

RAND_GEN_DISCR – Generates pseudorandom numbers from a general discrete distribution using an alias method or optionally a table lookup method.

RAND_ORTH_MAT – Generates a pseudorandom orthogonal matrix or a correlation matrix

RANDOM – Generates pseudorandom numbers.

RANDOM_ARMA – Generate pseudorandom ARMA process numbers.

RANDOM_TABLE – Sets or retrieves the current table used in either the shuffled or GFSR random number generator.

RANDOMOPT – Control of the random number seed and uniform(0,1) generator.

RANDOM_NPP – Generates pseudorandom numbers from a nonhomogeneous Poisson process.

RANDOM_ORDER – Generates pseudorandom order statistics from a standard normal distribution.

RANDOM_SAMPLE – Generates a simple pseudorandom sample from a finite population.

RAND_TABLE_2WAY – Generates a pseudorandom two-way table.

MERSENNE TWISTER GENERATOR

RANDOM_MT32_INIT – Initializes the 32-bit Mersenne Twister generator using an array.

RANDOM_MT64_INIT – Initializes the 64-bit Mersenne Twister generator using an array.

NEURAL NETWORKS

MLFF_NETWORK – Links and modifies a multilayered feedforward neural network.

MLFF_NETWORK_FORECAST – Calculates forecasts for trained multilayered feedforward neural networks.

MLFF_NETWORK_INIT – Creates a multilayered feedforward neural network.

MLFF_NETWORK_TRAINER – Trains a multilayered feedforward neural network using quasi-Newton backpropagation.

SCALE_FILTER – Scales or unscales continuous data prior to its use in neural network training, testing, or forecasting.

TIME_SERIES_FILTER – Converts time series data to the format required for processing by a neural network.

TIME_SERIES_CLASS_FILTER – Converts time series data sorted within nominal classes in decreasing chronological order to a useful format for processing by a neural network.

UNSUPERVISED_NOMINAL_FILTER – Converts nominal data into a series of binary encoded columns for input to a neural network.

UNSUPERVISED_ORDINAL_FILTER – Converts ordinal data into proportions.

UTILITIES

CONSTANTS AND DATA SETS

MACHINE – Machine constants

STATDATA – Statistical data sets

MATHEMATICAL SUPPORT

BINOMIALCOEF – Evaluate the binomial coefficient.

BETA – Evaluate the complete beta function.

BETAI – Evaluate the real incomplete beta function.

LNBeta – Evaluate the log of the real beta function.

GAMMA_ADV – Evaluate the real gamma function.

GAMMAI – Evaluate the incomplete gamma function.

LNGAMMA – Evaluate the logarithm of the absolute value of the gamma function.

ERROR HANDLING

CMAST_ERR_TRANS – Informational Error codes for routines

CMAST_ERR_STOP – Sets options for error recovery.

CMAST_ERR_PRINT – Sets options for error printing.

SIGNALS AND SYSTEMS

FILTER DATA STRUCTURE ROUTINES

FILTSTR – Place information into filter structure

PARSEFILT – Extract information from filter structure

PARSEWAVELET – Extracts the wavelet transform coefficients from a wavelet data structure

FILTER ANALYSIS FUNCTIONS

ARG – Phase of a complex array

FREQRESP_S – Complex frequency response of analog transfer function (Bode plot)

FREQRESP_Z – Complex frequency response of digital transfer function

IMPRESP – Impulse response of a filter

SINC – Computes the sine of the input divided by the input

FILTER APPROXIMATION

CLASSICAL FILTER APPROXIMATION

BILINTRANS – Bilinear Transformation

FIRDESIGN – FIR lowpass, highpass, bandpass, bandstop filter design

FIRLS – FIR multiple bandpass FIR filter design

FIRWIN – FIR window functions

FREQTRANS – IIR filter frequency transformation

FREQTRANSDESIGN – IIR filter frequency transformation design for multiple bandpass IIR filter design

IIRDESIGN – IIR Butterworth, Chebyshev I, Chebyshev II, and elliptic filter design

IIRORDER – IIR filter order estimation

ADVANCED FILTER APPROXIMATION

FIRLS – Frequency domain least-squares FIR filter design

IIRLS – Frequency and time domain least-squares filter design

REMEZ – Optimal Chebyshev error FIR filter design (Parks-McClellan algorithm)

SGFDESIGN – Optimal Savitsky-Golay FIR filter design

MULTIRATE FILTER APPROXIMATION

FILTDOWNDESIGN – Decimation filter design

FILTUPDESIGN – Interpolation filter design

QMFDESIGN – Quadrature mirror filter design

FILTER REALIZATION**STANDARD FIR AND IIR FILTER REALIZATION**

FILTER – Basic FIR and IIR filter realization

FIRFILT – Convolution and FFT-based filter realization for FIR filters

IIRFILT – Causal and anti-causal (forward-backward) IIR filter realization

MULTIRATE FILTER REALIZATION

FILTDOWN – Decimation filter

FILTUP – Interpolation filter

QMF – Quadrature mirror filter

TRANSFORMS AND SPECTRUM ANALYSIS

DCMPLXFFT – Double-precision complex fast Fourier transform

FFTCOMP – Fast Fourier transform

FFTINIT – Fast Fourier transform initialization

HILBERT – Constructs a Hilbert transformation

SPECTROGRAM – Spectrogram or short-time Fourier transform analysis

SPECTRUM – Power spectrum analysis (power spectral density)

WAVELET – Wavelet transform

STATISTICAL SIGNAL PROCESSING

FIRLS – FIR Wiener filter design

IIRLS – Prony's and frequency-sampling methods

JURYRC – Jury (reflection coefficient) algorithm

LEVCORR – Auto-correlation sequence computation from factored Toeplitz forms

LEVDURB – Levinson-Durbin algorithm for factoring Toeplitz matrices

LPC – Linear prediction coefficients

RANDOM – Random number generation

RANDOMOPT – Random number generation control

TOEPSOL – Levinson's algorithm for solving Toeplitz linear equations

POLYNOMIAL MANIPULATION

STANDARD POLYNOMIAL MANIPULATION

CONVOL1D – One dimensional convolution

DBLPOLY – Polynomial function evaluation

PAIRCONJ – Pair conjugate complex numbers

PAIRINV – Pair reciprocal complex numbers

P_DEG – Numerical determination of polynomial degree

P_DIV – Polynomial division

P_MULT – Polynomial multiplication

P_SQRT – Polynomial spectral factorization

P_SUM – Polynomial sum

ROOT2POLY – Compute coefficients of polynomial with specified roots

ZEROPOLY – Polynomial root finding

FUNCTIONS FOR POLYNOMIALS IN Z^{-1}

P_STAB – Polynomial stabilization

SCHURCOHN – Schur-Cohn stability test

OPTIMIZATION

FMIN – Univariate unconstrained minimization

FMINV – Multivariate unconstrained minimization

INTFCN – Numerical integration or quadrature

LINPROG – Linear programming constrained minimization

NLINLSQ – Nonlinear least-squares unconstrained minimization

NONLINPROG – Nonlinear programming nonlinearly constrained minimization

NORM – Computes various array norms

QUADPROG – Quadratic programming constrained minimization

PLOTTING AND SIGNAL GENERATION

OPLOTCOMB – Comb plot over a previously drawn plot

PLOTCOMB – Comb plot

PLOTZP – Zero-pole plot

REFLINES – Reference lines to indicate specific axis values

SIGNAL – Generates commonly used signals

POINT OPERATIONS

ALGEBRAIC AND LOGICAL OPERATIONS

BLEND – Blends two images together

IPALOG – Computes the natural logarithm of an image, excluding zero values

IPMATH – Performs mathematical and logical operations on a single image or between two images

THRESHOLDING

DENSITY_SLICE – Performs color density slicing on individual 2D images

THRESH_ADAP – Performs adaptive thresholding on an image

THRESHOLD – Performs either binary or grayscale global thresholding on an image

HISTOGRAM OPERATIONS

HIST_STATS – Computes six different statistical operations on image histograms

IPHISTOGRAM – Computes the density function or the cumulative density function for an image

FILTERING

SPATIAL DOMAIN LINEAR FILTERS

CANNY – Applies the Canny edge-detection method to an image

FILT_SMOOTH – Performs smoothing on 1D, 2D or 3D image arrays

GAUSS_KERNEL – Computes a 1D or 2D spatial Gaussian filter kernel

IPCONVOL – Performs 1D, 2D or 3D convolution on signals, images, and volumes

Spatial Domain Linear Filters (cont.)

IPCREATE_FILTER – Creates a spatial filter object, given a kernel and other appropriate fields

IPREAD_FILTER – Reads an ASCII text or XDR file depending on whether the file contains a spatial or spectral filter (respectively)

IPWRITE_FILTER – Saves a 2D convolution kernel to an ASCII text file

FREQUENCY DOMAIN LINEAR FILTERS

FILT_FREQ – Generates a 2D Butterworth or ideal lowpass, bandpass, bandstop or highpass spatial frequency domain filter

FILT_NOTCH – Generates a 3D ideal notch spatial frequency domain filter

FILT_WIENER – Computes and applies a parametric Wiener filter to an image that is either in the spatial or the spatial frequency domain

NONLINEAR AND ADAPTIVE FILTERS

FILT_DWMTM – Performs 1D, 2D or 3D adaptive double-window-modified trimmed mean filter

FILT_MMSE – Performs 1D, 2D or 3D adaptive minimum mean-squared error filtering

FILT_NONLIN – Performs nonlinear filtering operations on 1D, 2D, or 3D image arrays

WINDOWING

IPWIN – Computes one of several different data windows: Blackman, Chebyshev, Hamming, Hanning, Kaiser, rectangular, or triangular

MORPHOLOGICAL IMAGE PROCESSING

DIST_MAP – Computes the Euclidean distance map for an image

HIT_MISS – Performs the morphological hit-or-miss transform for shape processing

MORPH_CLOSE – Performs the morphologic-close operation for shape processing

MORPH_OPEN – Performs the morphologic-open operation for shape processing.

MORPH_OUTLINE – Performs morphologic outlining for shape processing

SKELETONIZE – Performs the morphologic skeletonizing operation for shape processing

TOP_HAT – Performs the morphologic top-hat transformations for shape processing

MENSURATION

CENTROID – Calculates the centroid of a binary region of an image

ENTROPY – Computes the entropy of an array

IPSTATS – Computes up to eight different statistical operations on an array, including the mean, variance, standard deviation, skewness, kurtosis, minimum, maximum, and range

KURTOSIS – Computes the kurtosis of an array

MAJOR_AXIS – Computes the major axis of a region in an image

MODE – Determines the mode of an array

MOMENT2D – Computes the 2D moments of an array

PERIMETER – Computes the perimeter of a region in an image

RANGE – Computes the range of values in an array

SKEWNESS – Computes the skewness of an array

UNIFORMITY – Computes the uniformity of an array

REPRESENTATION AND DESCRIPTION

TEXTURE

GLCM – Computes the gray level co-occurrence probability matrix for a graylevel image

GLCM_STATS – Calculates five statistics on the graylevel co-occurrence matrix

GLRL – Computes the graylevel run length matrix used for textural analysis of an image

GLRL_STATS – Calculates five statistics on the graylevel run length matrix

HIST_STATS – Computes six different statistical operations on image histograms

CORRELATION

IPCORRELATE – Performs direct (spatial domain) or indirect (spatial frequency domain) correlation between an array and a template

IMAGE TRANSFORMS

DCT – Performs the discrete cosine transformation on a 2D image or a 3D array of images

HAAR – Performs a Haar transform on a square image

HOUGH – Computes the line or circle Hough transform of an image

IPCT – Performs the inverse principle components transform on a multi-layered image

IPSPECTRUM – Estimates the power spectrum (power spectral density) of an image

IPWAVELET – Computes the separable wavelet transform for an image

PCT – Performs the principle components transform on a multi-layered image

POLAR_FFT – Transform the FFT of an image or images from a rectangular-coordinate space into a polar-coordinate space, and then sums the polar FFT along rho and theta

RADON – Computes the forward radon transform of an image

SLANT – Performs a Slant transform on a 2D square image

GEOMETRIC TRANSFORMS

IPSCALE – Scales an image by shrinking or expanding it

SHIFT_EDGE – Performs edge enhancement on an image using either the shift and subtract, or the shift and XOR technique

COLOR IMAGE PROCESSING

DENSITY_SLICE – Performs color density slicing on individual 2D images

COLOR QUANTIZATION

IPCOLOR_24_8 – Converts a 24-bit color image to an 8-bit color image; or, converts an 8-bit color image to a 24-bit color image

IPLINEAR_GRAY – Transforms an image with a nonlinear grayscale colormap to a linear grayscale image

IS_GRAY_CMAP – Determines if a colormap is grayscale

CLASSIFICATION AND SEGMENTATION

IPCLASSIFY – Performs supervised classification using the maximum likelihood classifier

IPCLUSTER – Performs image segmentation using K-means clustering based on regional statistical measures of the mean, mode, minimum, maximum and/or range of the image pixels

REGION_COUNT – Computes the number of regions in an image

REGION_FIND – Locates a possible seed point for a region in an image

REGION_GROW – Grows homogeneous regions in an image, where homogeneity is based on the region average

REGION_MERGE – Merges homogeneous regions in an image, where homogeneity is based on the region average

REGION_SPLIT – Splits homogeneous regions in an image, where the homogeneity is based on the region range

REGION_STATS – Performs several statistical calculations on specified regions in an image

THRESH_ADAP – Performs adaptive thresholding on an image

THRESHOLD – Performs either binary or grayscale global thresholding on an image

UTILITIES

NOISE_GEN – Returns an m-dimensional array of the desired noise distribution

NOISE_IMPULSE – Generates an array of impulse noise, also known as salt-and-pepper noise, in a blank array with a specified background; or applies the impulse noise to an existing image array

NOISE_PERIODIC – Generates an array of periodic (coherent) noise

NOISE_RAYLEIGH – Generates an array of Rayleigh distribution noise

PAD_IMAGE – Places a constant border around a volume, image or signal

Database Connection for Oracle and Sybase

DB_CONNECT – Connects PV-WAVE to a database.

DB_GET_BINARY – Returns binary large objects (BLOBS) from a DBMS (database management system) server.

DB_SQL – Queries the database currently connected to PV-WAVE.

DB_DISCONNECT – Disconnects PV-WAVE from an external database.

NULL_PROCESSOR – Facilitates the use of the Null_Info keyword for the DB_SQL function by extracting the list of rows containing missing for one or more columns.

Database Connection for ODBC

NULL_PROCESSOR PROCESSOR – Facilitates the use of the Null_Info keyword for the DB_SQL function by extracting the list of rows containing missing for one or more columns.

ODBC_BIND_PARAMS – ODBC_BIND_PARAMS is used to bind PV-WAVE variables to the parameter markers in an SQL statement.

ODBC_COMMIT – Saves changes for an ODBC transaction.

ODBC_CONNECT – Connect to an ODBC compliant data source.

ODBC_DISCONNECT – Terminate an ODBC connection.

ODBC_DRIVER_CONNECT – Connect to an ODBC compliant data source.

ODBC_EXIT – Exit an ODBC session.

ODBC_FETCH – Initiates a fetch operation.

ODBC_INIT – Initiates an ODBC session.

ODBC_LEVEL – Tests for ODBC level compliance for specific drivers related to fetch operations only.

ODBC_META – Search for data source objects.

ODBC_PREPARE – Initiate an SQL command and prepare for fetch (cursor) operations.

ODBC_ROLLBACK – Cancels changes for an ODBC transaction.

ODBC_SQL – Initiate an SQL command.

GTDESTROYSURF – Releases all allocated memory space associated with a specified surface.

GTGETBOUNDS – Retrieves the 3D bounding box from a gridded surface.

GTGETFAULTS – Retrieves the fault lines from a surface.

GTGETGRID – Retrieves the grid matrix from a surface.

GTGETSURFS – Lists the IDs of all currently active surfaces.

GTGETVALUE – Returns the value at a specified coordinate on the surface.

GTGRID – Produces an evenly sampled grid from scattered data.

GTINSTALLSURF – Creates a surface from grid and fault lines.

GTRESTORESURF – Restores a previously saved surface.

GTSAVESURF – Saves a surface to disk.