```
   // Create solution vector
   Make/FREE/D/N=(3) vector = {5,2,2}

   // Solve system of linear equations
   // Adding sparseMatrixType={TRIANGULAR,UPPER,NON_DIAG} may improve speed
   MatrixSparse rowsA=3, colsA=3, csrA={values,columns,ptrB}, vectorX=vector,
           operation=TRSV
   WAVE M_TRSVOut       // Output from TRSV

   Print M_TRSVOut      // Should be {-15, 8, 2}
End
```

Clustering

The following operations perform cluster analysis of various kinds:

FastGaussTransform, **FPClustering**, **ImageThreshold**, **KMeans**, **HCluster**

For hierarchical clustering, see the next section.

Hierarchical Clustering

Hierarchical clustering builds a hierarchy of clusters which provides the information needed to create a cluster dendrogram. You can perform hierarchical clustering using the **HCluster** operation added in Igor Pro 9.00. HCluster, based on code developed by Daniel Müllner, uses an agglomerative hierarchical clustering algorithm.

You provide as input either a square dissimilarity matrix or a matrix in which rows represent vectors in some data space as the input wave. A dissimilarity matrix contains values measuring the degree of difference between every pair of vectors in the original data set. It is common to use "distance" instead of "dissimilarity" because for most purposes Euclidean distance is used to measure dissimilarity. We use "dissimilarity" below.

HCluster creates one or two output waves, depending on the output type that you request. The output waves are a square dissimilarity matrix wave and a dendrogram wave. The format of the dendrogram wave is discussed below under **Dendrogram Wave Format**.

Agglomerative hierarchical clustering proceeds by iteratively finding the pair of nodes that have the minimum dissimilarity amongst all pairs. The node pair with the smallest value of dissimilarity is joined into a single replacement node. A node can represent a single original data vector or a set of vectors that have already been joined into a cluster. This process is repeated until all original data vectors are represented by a single cluster. The output dendrogram wave describes a tree identifying the nodes that were combined and the dissimilarity between those nodes. This description is sufficient for drawing a dendrogram illustrating the clusters.

If the input wave is a dissimilarity matrix, then those dissimilarities are used to form pair-wise nodes in a dendrogram expressing the degree of dissimilarity between pairs of nodes in the tree.

If the input wave is a matrix of raw data vectors, the HCluster operation can be used to either create a dissimilarity matrix or to create a dendrogram directly, without outputting a dissimilarity matrix. It is also possible to get both the dissimilarity matrix and the dendrogram wave from a single call. If you don't need the dissimilarity matrix output, in certain cases vector data can be processed using an algorithm that minimizes memory use.

There are two types of dissimilarity that HCluster need to calculate:

- The dissimilarity between two input vectors
  We call this a "vector dissimilarity" calculation.
  You specify how to calculate vector dissimilarity using the HCluster /DISS flag.