The first two of these waves can be read as ordered pairs: (5,0), (8,1), (3,2), and (6,1). Each ordered pair specifies a value (e.g., 6) and a column (e.g., 1) but does not tell us in which row the value appears.

The third wave, W_CSRPointerB, is used to determine in which row a given value appears. It contains one index for each row in the represented matrix plus an additional index which is nnz, the number of non-zero values in the matrix. W_CSRPointerB[i] is the zero-based index in W_CSRValues of the first non-zero value in row i.

In this example, we can interpret the values of W_CSRPointerB as follows:

```
W_CSRPointerB[0] = 0    // First value for row 0 is at index 0 in W_CSRValues *
W_CSRPointerB[1] = 0    // First value for row 1 is at index 0 in W_CSRValues
W_CSRPointerB[2] = 2    // First value for row 2 is at index 2 in W_CSRValues
W_CSRPointerB[3] = 3    // First value for row 3 is at index 3 in W_CSRValues
W_CSRPointerB[4] = 4    // Number of non-zero values in W_CSRValues is 4
```

* There are no non-zero values in row 0 so W_CSRPointerB[0] is the same as W_CSRPointerB[1].

When you specify a sparse matrix in CSR format to the MatrixSparse operation, the last element of W_CSRPointerB, specifying nnz, is optional and you can omit it.

The wave names W_CSRValues, W_CSRColumns, and W_CSRPointerB are used by MatrixSparse when it creates an output sparse matrix in CSR format. When you specify an input sparse matrix, you can use any wave names.

CSC Sparse Matrix Storage Format

"CSC" is the shorthand name for "compressed sparse column". It is more efficient in terms of memory use and computational speed than COO.

In CSC format, the three 1D waves store the non-zero values, the zero-based row indices, and a "pointer" vector which is used to determine in which column each value is to be stored.

In Igor terminology, CSC format uses the following three waves:

W_CSCValues stores each non-zero value in the matrix.

W_CSCRows stores the zero-based row indices for each non-zero value in the matrix.

W_CSCPointerB stores indices into W_CSCValues which are used to determine in which column a particular value appears.

The W_CSCPointerB wave works in CSC in a manner analogous to how W_CSRPointerB in CSR.

When you specify a sparse matrix in CSC format to the MatrixSparse operation, the last element of W_CSCPointerB, specifying nnz, is optional and you can omit it.

The wave names W_CSCValues, W_CSCRows, and W_CSCPointerB are used by MatrixSparse when it creates an output sparse matrix in CSC format. When you specify an input sparse matrix, you can use any wave names.

Sparse Matrix Example

To help you get a feel for how the MatrixSparse operation works, here is a simple example showing multiplication of a sparse matrix by a vector using the MatrixSparse MV operation.

```
Function DemoSparseMatrixMV()
   // Define Wikipedia example sparse matrix in CSR format
   Make/FREE/D values = {5, 8, 3, 6}   // Double-precision floating point
   Make/FREE/L columns = {0, 1, 2, 1}  // 64-bit signed integer
   Make/FREE/L ptrB = {0, 0, 2, 3, 4}  // 64-bit signed integer

   // Create a vector
```