Webnucleo Technical Report: The Arrow Solver in the wn_matrix Module

Bradley S. Meyer

August 20, 2009

As of version 0.8 of wn_matrix, there is a Gaussian elimination solver for an "arrow" type matrix. An arrow-type matrix is one for which the non-zero elements form a band along the diagonal and two wings along the far right and bottom of the matrix.

1 The WnMatrix__Arrow Structure

The form of an arrow matrix is shown schematically in Fig. 1. The matrix has N rows, therefore, in dense form it has $N \times N$ elements. The arrow matrix has band width n and wing width m. The arrow matrix comprises four submatrices, as shown in Fig. 1. The a matrix has $N-m$ rows and n columns. The row index of a corresponds to the row index of the parent matrix while the column index gives the position relative to the diagonal. Matrix elements in the central band farthest from the diagonal are $m_b = (n-1)/2$ columns away from the diagonal. The b matrix is one of the wings of the arrow. It has m rows corresponding to the columns beyond the column $N-m$ and $N-m$ columns corresponding to the row of the element. The c matrix has m rows corresponding to rows beyond the row $N-m$ in the parent matrix and $N-m$ columns corresponding to the columns in the parent matrix. Finally, the d matrix is an $m \times m$ matrix with indices corresponding to the most rightward and bottom elements in the parent matrix.

The mapping between an element in the parent matrix $M(i, j)$ at row i and column j and the corresponding element in the arrow matrix is determined as follows. If $i \leq N-m$ and $j \leq N-m$, then $a(i, m_b+1-i+j) = M(i, j)$. If $i \leq N-m$ and $j > N-m$, then $b(j-N+m, i) = M(i, j)$. If $i > N-m$ and $j \leq N-m$, then $c(i-N+m, j) = M(i, j)$. Finally, if $i > N-m$ and $j > N-m$, then $d(i-N+m, j-N+m) = M(i, j)$. The above mappings assume 1-based indexing. Notice that the full range on the second index of the a matrix can put the index outside the parent matrix. This array must therefore be checked to ensure it remains in bounds. It should be clear that the arrow matrix form requires storage of $(N-m)*(n+2m)+m^2$ elements. If $n, m << N$, the arrow format can result in substantial storage savings over the dense form.

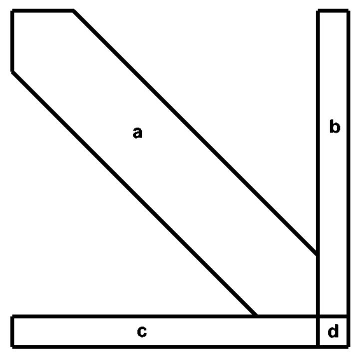

Figure 1: Arrow matrix.

The data for the arrow matrix are stored in a WnMatrix__Arrow structure, which is obtained from a WnMatrix structure by the API routine with prototype

WnMatrix__Arrow *
WnMatrix__getArrow(WnMatrix *p_matrix, size_t i_wing).

Here i_wing is the desired wing width for the arrow matrix (the quantity m). The routine determines the appropriate band width from the parent matrix. Diagnostics on the arrow matrix are available from the API routines

size_t
WnMatrix__Arrow__getNumberOfRows(WnMatrix__Arrow * p_arrow_matrix),

which returns N, the number of rows,

size_t
WnMatrix__Arrow__getWingWidth(WnMatrix__Arrow * p_arrow_matrix),

which returns m, the wing width, and

size_t
WnMatrix__Arrow__getBandWidth(WnMatrix__Arrow * p_arrow_matrix),

which returns n, the band width.

2 Solving an Arrow Matrix Equation

We suppose now that we have the matrix equation $Ax = b$. If the matrix A is stored as a WnMatrix__Arrow matrix p_arrow_matrix and the right-hand-side

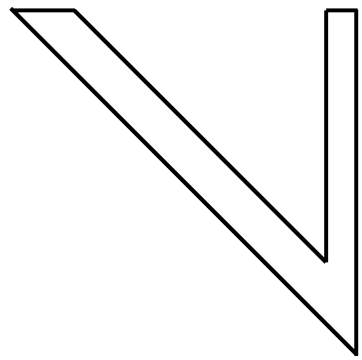

Figure 2: Arrow matrix after Gaussian elimination.

vector b is stored as a gsl vector pointed to by p_rhs, the user may find the solution vector p_solution from an API routine as:

p_solution = WnMatrix__Arrow__solve(p_arrow_matrix, p_rhs).

p_solution is another gsl vector. The routine solves the matrix by Gaussian elimination. Through row operations, the routine reduces the matrix to the form shown in Fig. 2. Simultaneous operations change the rhs vector. From this form, simple backsubstitution gives the solution vector. The arrow matrix and the rhs vector returned from WnMatrix__Arrow are those after the row operations. This permits diagnostics on the solution. For example, after the solution is obtained, the user can get a WnMatrix form for the arrow from the API routine

p_solved_matrix = WnMatrix__Arrow__getWnMatrix(p_arrow_matrix)

which may be studied or output to text with other API routines. Since the solver modifies the input matrix and rhs vector, the user should make copies of those prior to solving the matrix equation if he or she requires those later. Once the user is done with the arrow matrix, he or she should free it with the API routine WnMatrix__Arrow__free().