Matrix Unit

consists of Tmatrix object
with modules
constructor init;
procedure load(row,column:integer;value:real);
procedure solve(var solution;count:integer);
function check:real;
destructor done;

Init initializes two pointers, m and n of the form ^array[1..800] of prow where prow is array[1..801] of real. If you wish to speed up the object, then reduce 800 to a smaller number but make sure prow is always one
more. (200 and 201 for example). You'll also have to change s:array[1..800] of real (a field in tmatrix) and the init's new and the done's dispose statements. Plus s1 in solve.

Load is used with two for, to , do loops to load m^ and n^. This procedure expects a square matrix with the addition of one more column which represents the output array.   new matrix
[800x801]

```
                                                      | i         i   . O |
|          |        |   |         |   |               |  i        i    .O|
| input    |   X  |  S |    =   | O|                  | i         i   . O|
|           |       |    |         |   |              S=solution array [800x1]
                                                       O=output array  [800x1]
                                                      input array[800x800] of
```

coefficients to simultaneous equations. Example of loading matrixes m^ and n^ found in mattest.pas.

Solve uses Cholesky's method (one of fastest methods for computer iteration). For solution parameter, put in solution:array[1..800] of real or solution:array[1..x] of real if row of square matrix is x. (matrix x by x in other words. In count parameter, put in number of rows of matrix, number of coefficients of simultaneous equations in other words. Example in Mattest.pas solves 20X20 matrix (20 simultaneous equations) with no adjustment to m^ and n^ (no adjustment to code).

To see answers, write your array you put in solution out.

Check verifies that solution correct. use a variable like a:=v.check where v:tmatrix. A small number in a denotes very little difference between the input matrix times the s versus the o matrix element in each case.

Done is used as a final step to dispose of m and n.

Note all elements of m and n are referred to as m^[a]^[b] where a is the row number and b is the column number.
For small matrixes, do the modifications listed above to reduce from 800 to whatever. 800x801 represents close to 4.5 meg of memory for real numbers (6 bytes). Enjoy.