First Name: Peter
Last Name: Gruber
Date: 01.12.03
Homework Number: 2
Homework Title: Exercise 2.13

Problem 2.13: *How would you solve a partitioned linear system of the form*

$$\begin{bmatrix} L_1 & O \\ B & L_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix},$$

where L_1 and L_2 are nonsingular lower triangular matrices, and the solution and right-hand-side vectors are partitioned accordingly? Show the specific steps you would perform in terms of the given submatrices and vectors.

Solution:

Let L_1 be a nxn-matrix and L_2 a kxk-matrix. Consequently, O will be a nxk-matrix and B a kxn-matrix.
The main idea is to eliminate all entries in O and B, that is to perform elementary matrix manipulations so that all those entries will become zero in the end. Eliminating these entries, we will make sure that L_1 and L_2 remain trinagular matrices (that means, all exisiting zeros may remain untouched). The algorithm secures this by eliminating rows of O and columns of B alternatively (meaning last row of O followed by last column of B followed by last but one row of O and so on). After this, we will have two Linear Systems with lower triangular matrices $L1$ and $L2$ which may be solved independently e.g. by forward substitution.
The algorithm might look like this:

for i = n..1
eliminate all rows of O (from down to top) / all columns of B (from right to left)

for j = k..1
eliminate all entries in i^{th} row of O / in i^{th} column of B

for s = j..1
eliminating $O(i,j)$, update all other entries in i^{th} row of O
$O(i,s) = O(i,s) - \frac{L_2(j,s)}{L_2(j,j)} O(i,j)$

end;

for s = i..1
update entries in L_1 as well (entries in upper triangular part are zero all time and need not to be updated, as all columns in B "right from index i" should already be set to zero!)
$L_1(i,s) \;=\; L_1(i,s) - \frac{B(j,s)}{L_2(j,j)} O(i,j)$
end;

for s = 1..k
eliminate all entries in i^{th} column of B

for r = 1..i
update all entries in s^{th} row (all entries right from index i are already zero and need not to be updated (as they remain untouch because L_1 is lower triangular)!)
$B(r,s) \;=\; B(r,s) - \frac{L_1(i,s)}{L_1(i,i)} B(r,i)$
end;

end;
end;
end;

If we suppose that the matrix O is already a zero matrix, this means that $\begin{bmatrix} L_1 & O \\ B & L_2 \end{bmatrix}$ is already a lower triangular matrix. So we could solve the Linear System simply by applying the forward-substitution-method to the whole matrix.
Alternatively, we may also split the whole process into a few steps, dealing only with one of the given submatrices in each step.
The steps we would perform are as follows:

1. Solve the sub-system $L_1\, x \;=\; b$. (Most likely, we would use forward substitution here.)
2. Knowing the solution for the sub-vector x, compute the vector $B\, x$.
3. Compute the vector $c' \;:=\; c \;-\; B\, x$.
4. Solve the sub-system $L_2\, y \;=\; c'$.

This finally leads to a solution for the vector $\begin{bmatrix} x \\ y \end{bmatrix}$.

This paritioned algorithm will require $O\left(\frac{k^2+n^2}{2} + n * k\right)$ steps (don't forget we also have to calculate the vector $B\ x$, which leads to $n*k$ multiplications!), compared to $O\left(\frac{(k+n)^2}{2}\right)$ when applying Forward Substitution to the whole matrix. Thus, both methods require the same amount of operations.
Therefore, the partitioned algorithm won't reduce the amount of operations we have to perform (compared to applying forward-substitution to the whole matrix), but as only one of the submatrices needs to be present in each step, this can bring an advantage concerning the requested memory space if the submatrices themselves are very large.