Below is a list of problems for our lab session

Wednesday, January 7th, (kl. 13.00-16.00), Room MA436-446.

These problems center the solution of linear systems.
Problem 1 (Hand calculations only) Solve the linear system

\[ Ax = f \]

where

\[
A = \begin{bmatrix}
1 & -1 & 1 & -1 \\
2 & 0 & 0 & 0 \\
3 & 5 & -2 & 2 \\
5 & 7 & 14 & -10
\end{bmatrix}, \quad \text{and} \quad f = \begin{bmatrix}
-2 \\
2 \\
15 \\
21
\end{bmatrix}.
\]
Problem 2 (Hand calculations only) Consider the matrices $L$ and $U$ given by

$$L = \begin{bmatrix}
1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 \\
3 & 4 & 1 & 0 \\
5 & 6 & 7 & 1 \\
\end{bmatrix}, \quad \text{and} \quad U = \begin{bmatrix}
1 & -1 & 1 & -1 \\
0 & 2 & -2 & 2 \\
0 & 0 & 3 & -3 \\
0 & 0 & 0 & 4 \\
\end{bmatrix}. $$

1. Show that $A = LU$, where $A$ is the matrix given in Problem 1.
2. Solve the linear system $Ly = f$ where $f$ is given in Problem 1.
3. Solve the linear system $Ux = y$ and compare $x$ to the solution of Problem 1.
Problem 3 Consider the following MATLAB routine which is designed to solve a nonsingular upper triangular linear system

\[ Ux = f \]

using forward substitution.

```matlab
function x=backwardsub(u,f);
n=size(u,1); x=f;
for i=n:-1:1
    x(i)=x(i)/u(i,i);
    x(1:i-1)=x(1:i-1)-u(1:i-1,i)*x(i);
end
```

Verify experimentally that it actually accomplishes this task! How? Pick a upper triangular matrix \( U \) and a solution \( x \) and compute \( f = Ux \) and feed \( U \) and \( f \) to the function. Do you recover \( x \) exactly? What is the normwise relative error, i.e. compute \( \|x - \hat{x}\|_\infty / \|x\|_\infty \). Explain how the function works.
Problem 4 Consider the following MATLAB routine which is designed to solve a nonsingular lower triangular linear system
\[ Lx = f \]
using forward substitution.

```matlab
function x = forwardsub(l, f)
    n = size(l, 1); x = f;
    for i = 1:n
        x(i) = x(i) / l(i, i);
        x(i+1:n) = x(i+1:n) - l(i+1:n, i) .* x(i);
    end
```

Verify experimentally that it actually accomplishes this task! How? Pick a lower triangular matrix \( L \) and a solution \( x \) and compute \( f = LX \) and feed \( L \) and \( f \) to the function. Do you recover \( x \) exactly? What is the normwise relative error, i.e. compute \( \|x - \tilde{x}\|_\infty /\|x\|_\infty \). Explain how the function works.
Problem 3 Consider the following MATLAB function which is designed to compute the LU factorization of a nonsingular matrix $A$.

```matlab
function [l u sigma]=factor(a)
% LU factorization routine for m by m matrices
% It will not work for general m by n matrices!
[m,n]=size(a); l=eye(m,m); u=zeros(m,n); b=a;
sigma=linspace(1,m,m); aux=linspace(1,m,m);
for j=1:m-1
    ii=j; piv=abs(b(ii,j));
    % Search for a better pivot in jth column.
    for i=j+1:m
        if (abs(b(i,j))>piv)
            ii=i; piv=abs(b(i,j));
        end
    end
    % Is the matrix singular?
    if (piv>0)
        if (ii>j)
            % Swap the physical rows
            temp=b(ii,:); b(ii,:)=b(j,:); b(j,:)=temp;
            % Record the swap
            temp=sigma(ii); sigma(ii)=sigma(j); sigma(j)=temp;
        end
        % Perform the update of the matrix
        for i=j+1:m
            b(i,j)=b(i,j)/b(j,j); % Record the multipliers
            for k=j+1:n
                b(i,k)=b(i,k)-b(i,j)*b(j,k);
            end
        end
    else % Singular matrix
        fprintf('The matrix is singular 

'); break;
    end
end
% Extract the factors
l=l+tril(b,-1); u=triu(b,0);
```

Verify experimentally that it will actually get the job done.

1. Construct a matrix $A=rand(5,5)$, examine your output. Does $A$ equal $LU$? Does $A(\sigma,:) = LU$? Compute $\|A(\sigma,:) - LU\|_\infty/\|A\|_\infty$. You should get a fairly small number.

2. Systematically repeat the above experiment with matrices of size 100$i$ for $i = 1, 2, \ldots, 10$. 
Problem 5 Create a MATLAB function “solve.m” which uses the output of “factor.m” from Problem 3 to solve a nonsingular linear system $Ax = f$. Verify experimentally that your routine will actually get the job done. How? Construct a sequence of linear systems of increasing dimension for which you know the solutions $x$ and compare the actual solutions to the computed solutions $\hat{x}$. 
Problem 6 Extremely bad matrices are easy to find! Let $A$ be the Hilbert matrix of dimension 20. ($A = \text{hilb}(20)$ in MATLAB.) Use your routines to compute the $LU$ factorization of $A$. Compute $\frac{\|A\sigma, -LU\|_\infty}{\|A\|_\infty}$. (You should get a small number.) Then let $x = (1, 1, \ldots, 1)^T \in \mathbb{R}^{20}$. Compute $f = Ax$ and feed the linear system $Ax = f$ to your solver! Examine the components of the computed solution $\hat{x}$. Are they even remotely similar to $x$? Is it even theoretically possible to solve this linear system on your computer? (Hint: What is the condition number of the matrix?)