Metody Obliczeniowe w Nauce i Technice
Laboratorium 1
Arytmetyka Komputerowa
Sprawozdanie

Yurii Vyzhha

5 listopada 2017

Zadanie 2 Faktoryzacja LU

```
function [p, A] = LU(A)
    [n,~] = size(A);
    s = zeros(1,5);
    p = zeros(1,n);
    for i = 1:n
        s(i) = max(abs(A(i,:)));
        p(i) = i;
    end
    for k = 1:n-1
        rmax = 0;
        for i = k+1:n
            r = abs(A(i,k)/s(i));
            if r > rmax
                rmax = r;
                imax = i;
            end
        end
        if k ~= imax
            tmp = P(k);
            p(k) = p(imax);
            p(imax) = tmp;
            A([k imax],:) = A([imax k],:);
        end
        for i = k+1:n
            xmult = A(i,k)/A(k,k);
            A(i,k) = xmult;
            for j = k+1:n
                A(i,j) = A(i,j) - xmult*A(k,j);
            end
        end
    end
end
```

Funkcja działa w następujący sposób. Najpierw tworzymy dwa wektory: s oraz p. s jest wektorem skalowania takim, że $s_i = \max\limits_{1 \leqslant j \leqslant n} |A_{ij}|$. p jest wektorem reprezentującym macierz zamiany rzędów P. Tą macierz można otrzymać najpierw tworząc macierz zerową $n \times n$ oraz w miejscach tej macierzy $(i, p(i))$ wpisać jedynki. W głównej pętli $1 \leqslant k \leqslant n-1$, gdzie n to rozmiar macierzy A, szukamy elementu *rmax* takiego, że rmax $= \max\limits_{k+1 \leqslant i \leqslant n} |\frac{A_{ik}}{s_i}|$. W danej interacji pętli, rmax jest pivotem, a imax jest indexem rzędu, w którym go znaleźliśmy. Jeśli imax $\neq$ k to zamieniamy rzędy miejscami, zapisując zmiany w vector P. Następnie w pętlin $i = k + 1 : n$ obliczamy współczynnik xmult, za pomocą którego redukujemy rzędy od $k + 1$ do n. Ten współczynnik wpisujemy na miejsce A_{ik}; on będzie elementem macierzy L. Po wykonianiu się programu dostajemy na wyjściu wektor P oraz macierz zmienioną macierz A, w której są jednocześnie zapisane macierze L oraz U.
Dla sprawdzenia poprawności danej funkcji, napisałem program, który sprawdza, czy dla P, L i U, otrzymanych jako wyniki funkcji faktoryzującej, zachodzi $P * A = L * U$.

```
function a = CheckLU(A)
    [n,~] = size(A);
    [p, lu] = LU(A);
    P = zeros(n);
    for i = 1:n
        P(i,p(i)) = 1;
    end
    L = eye(n);
    U = zeros(n);
    for i = n:-1:1
        for j = n:-1:i
            U(n-j+1,n-i+1) = lu(n-j+1,n-i+1);
        end
    end
    for i = 2:n
        for j = 1:i-1
            L(i,j) = lu(i,j);
        end
    end
    X = P*A;
    Y = L*U;
    a = true;
    for i = 1:n
        for j = 1:n
            if abs(X(i,j)-Y(i,j)) > 1e4*eps(min(abs(X(i,j)),abs(Y(i,j))))
                a = false;
                break
            end
        end
    end
end
```

W tej funkcji oblicamy P, L i U za pomocą wspomnianej wyżej funkcji oraz tworzymy nowe macierze $X = P * A$ i $Y = L * U$. Dalej porównujemy każdy odpowiedni element macierzy X oraz Y i zwracamy "prawdę", jeśli X_{ij} jest prawie równe Y_{ij} dla każdego $1 \leqslant i \leqslant n$ oraz $1 \leqslant j \leqslant n$.