Creating a Tridiagonal matrix in matlab

How can I create a tridiagonal matrix that I can use for Crout factorization? And, I don't have any codes on how to create one since I am new to matlab.

Consider the $n \times n$ tridiagonal system of equations $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}$$

and

$$\mathbf{b}^{\mathrm{T}} = \begin{bmatrix} 1 + \frac{1^2}{(n+1)^4} & \frac{2^2}{(n+1)^4} & \frac{3^2}{(n+1)^4} & \cdots & \frac{(n-1)^2}{(n+1)^4} & 6 + \frac{n^2}{(n+1)^4} \end{bmatrix}.$$

Ok, please help me understand what does the sentence "The program should output the ∞ norm of the residual of your computed solution and the number of iterations used" mean in this case? I am all confused figuring this out.

(matlab)

edited Apr 22 '14 at 18:37

asked Apr 22 '14 at 1:07

user136422
117 2 3 11

3 Answers

```
>> n = 10;
>> full(gallery('tridiag',n,-1,2,-1))

ans =

     2    -1     0     0     0     0     0     0     0     0
    -1     2    -1     0     0     0     0     0     0     0
     0    -1     2    -1     0     0     0     0     0     0
     0     0    -1     2    -1     0     0     0     0     0
     0     0     0    -1     2    -1     0     0     0     0
     0     0     0     0    -1     2    -1     0     0     0
     0     0     0     0     0    -1     2    -1     0     0
     0     0     0     0     0     0    -1     2    -1     0
     0     0     0     0     0     0     0    -1     2    -1
     0     0     0     0     0     0     0     0    -1     2
```

Crout:

```
% Source: http://users.csc.tntech.edu/~mjkosa/3020/matlab/crout.m
% MATLAB implementation of Crout reduction algorithm (p. 140 of your book)
function [L,U] = crout(A,n)  % returns two matrices

for i = 1:n
    L(i,1) = A(i,1);
end

for j = 1:n
    U(1,j) = A(1,j)/L(1,1);
end

for j = 2:n
    for i = j:n
        sum = 0.0;
        for k = 1:(j-1)
            sum = sum + L(i,k) * U(k,j);
        end
```

```
        L(i,j) = A(i,j) - sum;
    end

    U(j,j) = 1;

    for i = (j+1):n
        sum = 0.0;
        for k = 1:(j-1)
            sum = sum + L(j,k) * U(k,i);
        end
        U(j,i) = (A(j,i) - sum)/L(j,j);
    end
end
```

answered Apr 22 '14 at 1:14

Franck Dernoncourt
906 2 9 26

Thanks. And, I can also perform Crout factorization on that and get L and U? – user136422 Apr 22 '14 at 1:16

I am getting this error while trying to execute the codes. function [L,U] = crout(A,n) | Error: Function definitions are not permitted in this context. – user136422 Apr 22 '14 at 2:05

Use $[L,U] = crout(A,n)$ then hit enter. $function[L,U] = crout(A,n)$ defines a function; which is *normally* done in a script not at the command line. – K. Rmth Apr 22 '14 at 19:51

The tridiagonal part can be created using sums of calls to diag()

```
n = 5 ;
nOnes = ones(n, 1) ;
x = diag(2 * nOnes, 0) - diag(nOnes(1:n-1), -1) - diag(nOnes(1:n-1), 1)

x =

     2    -1     0     0     0
    -1     2    -1     0     0
     0    -1     2    -1     0
     0     0    -1     2    -1
     0     0     0    -1     2
```

answered Nov 4 '14 at 16:13

Peeter Joot
485 3 9

In your case

```
toeplitz([2 -1 zeros(1, N-2)], [2 -1 zeros(1, N-2)])
```

or even

```
toeplitz([2 -1 zeros(1, N-2)])
```

edited Sep 26 '16 at 19:05

Clarissa G
103 4

answered Aug 6 '15 at 4:43

Yola
738 6 20