$$
-E = \begin{pmatrix}
0 & 0 & 0 & \cdots & 0 & 0 \\
a_{21} & 0 & 0 & \cdots & 0 & 0 \\
a_{31} & a_{32} & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
a_{n-1\,1} & a_{n-1\,2} & a_{n-1\,3} & \ddots & 0 & 0 \\
a_{n\,1} & a_{n\,2} & a_{n\,3} & \cdots & a_{n\,n-1} & 0
\end{pmatrix}, \; -F = \begin{pmatrix}
0 & a_{12} & a_{13} & \cdots & a_{1n-1} & a_{1n} \\
0 & 0 & a_{23} & \cdots & a_{2n-1} & a_{2n} \\
0 & 0 & 0 & \ddots & a_{3n-1} & a_{3n} \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 0 & a_{n-1\,n} \\
0 & 0 & 0 & \cdots & 0 & 0
\end{pmatrix}.
$$

In *Jacobi's method*, we assume that *all* diagonal entries in A are nonzero, and we pick

$$
\begin{aligned}
M &= D \\
N &= E + F,
\end{aligned}
$$

so that by $(*)$,

$$B = M^{-1}N = D^{-1}(E + F) = I - D^{-1}A.$$

As a matter of notation, we let

$$J = I - D^{-1}A = D^{-1}(E + F),$$

which is called *Jacobi's matrix*. The corresponding method, *Jacobi's iterative method*, computes the sequence (u_k) using the recurrence

$$u_{k+1} = D^{-1}(E + F)u_k + D^{-1}b, \quad k \geq 0.$$

In practice, we iteratively solve the systems

$$Du_{k+1} = (E + F)u_k + b, \quad k \geq 0.$$

If we write $u_k = (u_1^k, \ldots, u_n^k)$, we solve iteratively the following system:

$$
\begin{array}{ccccccccc}
a_{11}u_1^{k+1} & = & & -a_{12}u_2^k & -a_{13}u_3^k & & \cdots & -a_{1n}u_n^k & +\, b_1 \\
a_{22}u_2^{k+1} & = & -a_{21}u_1^k & & -a_{23}u_3^k & & \cdots & -a_{2n}u_n^k & +\, b_2 \\
\vdots & \vdots & \vdots & & & & & & \\
a_{n-1\,n-1}u_{n-1}^{k+1} & = & -a_{n-1\,1}u_1^k & \cdots & -a_{n-1\,n-2}u_{n-2}^k & & & -a_{n-1\,n}u_n^k & +\, b_{n-1} \\
a_{n\,n}u_n^{k+1} & = & -a_{n\,1}u_1^k & -a_{n\,2}u_2^k & \cdots & -a_{n\,n-1}u_{n-1}^k & & & +\, b_n
\end{array}.
$$

In `Matlab` one step of Jacobi iteration is achieved by the following function:

```
function v = Jacobi2(A,b,u)
   n = size(A,1);
   v = zeros(n,1);
   for i = 1:n
      v(i,1)  = u(i,1) + (-A(i,:)*u + b(i))/A(i,i);
   end
end
```