Rozwiązanie układnu równań sprowadzać się będzie do rozwiązania układu $Ay = b$ Można zauważyć,że macież A ma pewną stałą postać zależną od h. W dodadku jest to macierz **trójdiagonalna**`

$$\frac{y_{n-1} - 2y_n + y_{n+1}}{h^2} + y_n = 0$$

$$y_{n-1} + (h^2 - 2)y_n + y_{n+1} = 0$$

$$y_0 = 1$$

$$y_N = 0$$

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & h^2-2 & 1 & \cdots & 0 \\ 0 & 1 & h^2-2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

$$y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix} b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

W związku z tym zaimplementowanie samej macierzy A staje się bardzo proste. Instotnym elementem jest jednak nie zapełnianie nadmierniej ilosci pamięci. Macierz trójdianonalną przechowywać możemy w formie 3 wektorów.

$$c_{i,j+1} = [0, \quad 1, \quad 1, \cdots \quad 1]$$

$$d_{i,j} = [1, \quad h^2-2, \quad h^2-2, \quad \cdots \quad h^2-2, \quad 1]$$

$$b_{i+1,j} = [1, \quad 1, \quad 1, \cdots \quad 0]$$

Aby uniknąć liczenia macierzy odwrotnej które zajęło by czas $O(n^3)$

$$y = A^{-1}b$$

Można zostosować algorytm Thomasa w celu zoopimalizowania faktoryzacji macierzy

- Faktoryzacja LU macierzy **trójdiagonalnej** dokonana zostanie w czasie $O(n)$

In [ ]:
```
#Procedura faktoryzacji:
for i in 2:n
    factor = v2[i-1] / diagonal[i-1]
    diagonal[i] -= factor * v1[i-1]
    v1[i-1] = factor
end
```

Następnie znając już macierze L oraz U zapisane w formie wektorowej algorytmem **forward substitution** oraz **backward substitution** obliczam wektor wartosci wedle wzorów :

$$Lx = b$$

$$Uy = x$$

Całościowe rozwiązanie takiego układu ma złożoność $O(n)$