对如下线性方程组：

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ -1 \\ 3 \\ 2 \end{bmatrix}$$

利用如下方法求解该线性方程组的解（精确解$\mathbf{X} = \begin{bmatrix} 0.2 & 0.2 & -0.5 & 0.8 & 0.3 \end{bmatrix}^T$）：

（1） 高斯消元法，最终得到的解以 DOS 窗口截图展示。

（2） 雅克比迭代法，其中初始解 $\mathbf{X}_k = (0,0,0,0,0)^T$，最终迭代解误差要求 $\left\| \mathbf{X}_k - \mathbf{X}_{k-1} \right\|_2 \le 10^{-3}$，最终得到的解以 DOS 窗口截图展示。并用 MATLAB 绘制 $\left\| \mathbf{X}_k - \mathbf{X}_{k-1} \right\|_2$ 随迭代步变化曲线图。；

（3） 阅读后面的附录材料，利用追赶法或者 Thomas 算法求解该线性方程组的解，最终得到的解以 DOS 窗口截图展示。

附录：三对角线性方程组(tridiagonal systems of equations)

三对角线性方程组经常出现在微分方程的数值求解和三次样条函数题中。三对角线性方程组可描述为以下方程组：

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i$$

其中$1 \le i \le n$，$a_1 = 0$，$c_n = 0$。以上方程组写成矩阵形式为$\mathbf{Ax} = \mathbf{d}$，即：

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}$$

三对角线性方程组的求解采用追赶法或者 Thomas 算法，它是高斯消元法在三对角线性方程组这种特殊情形下的应用，因此，主要思想还是高斯消元法，只是会更加简单些。我们将在下面的算法详述中给出该算法的具体求解过程。

当然，该算法并不总是稳定的，但当系数矩阵 A 为严格对角占优矩阵（Strictly Diagonally Dominant, SDD）或矩阵（Symmetric Positive Definite, SPD）时，该算法稳定。

追赶法或者 Thomas 算法详述

（1） 创建新系数c_i^*和d_i^*来代替原先的a_i，b_i，c_i，公式如下：

$$c_i^* = \begin{cases} \frac{c_1}{b_1} & ; i = 1 \\ \frac{c_i}{b_i - a_i c_{i-1}^*} & ; i = 2, 3, \ldots, n-1 \end{cases}$$

$$d_i^* = \begin{cases} \frac{d_1}{b_1} & ; i = 1 \\ \frac{d_i - a_i d_{i-1}^*}{b_i - a_i c_{i-1}^*} & ; i = 2, 3, \ldots, n-1 \end{cases}$$

（2） 改写原先的方程组 $\mathbf{Ax} = \mathbf{d}$ 如下：

$$\begin{bmatrix} 1 & c_1^* & 0 & 0 & \dots & 0 \\ 0 & 1 & c_2^* & 0 & \dots & 0 \\ 0 & 0 & 1 & c_3^* & 0 & 0 \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & c_{n-1}^* \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_k \end{bmatrix} = \begin{bmatrix} d_1^* \\ d_2^* \\ d_3^* \\ . \\ . \\ . \\ d_n^* \end{bmatrix}$$

（3） 计算解向量 $\mathbf{x}$，如下：

$$x_n = d_n^*, \qquad x_i = d_i^* - c_i^* x_{i+1}, \qquad i = n-1, n-2, \ldots, 2, 1$$

以上算法得到的解向量 $\mathbf{x}$ 即为原方程 $\mathbf{Ax} = \mathbf{d}$ 的解。