Algorithm 2 Newton's Method for a System of NE (5.4)

1: $\mathbf{x}_0$ = initial guess;
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Solve $\mathbf{J}_f(\mathbf{x}_k)\mathbf{s}_k = -\mathbf{f}(\mathbf{x}_k)$ for $\mathbf{s}_k$
4: $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$
5: **end for**

In Matlab, solve it by the \ operator: $\mathbf{s}_k = -\mathbf{J}_f(x_k)\backslash\mathbf{f}(x_k)$

Here, $\mathbf{f}(x_k)$ is the difference of the **current position** (returned by the evalRobot2D function) and the **expected position [x ; y]**.

The Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

m is the number of the function
n is the number of variables