```
function [X, iter] = jacobi_method(A, b, X0, tol, max_iter)
    % Jacobi Method for solving AX = b
    % Inputs:
    % A - Coefficient matrix
    % b - Right-hand side vector
    % X0 - Initial guess
    % tol - Convergence tolerance
    % max_iter - Maximum number of iterations
    % Outputs:
    % X - Solution vector
    % iter - Number of iterations performed

    n = length(b);
    X = X0;
    X_new = X0;
    iter = 0;

    while iter < max_iter
        for i = 1:n
            sum1 = A(i, 1:i-1) * X(1:i-1);
            sum2 = A(i, i+1:n) * X(i+1:n);
            X_new(i) = (b(i) - sum1 - sum2) / A(i, i);
        end

        % Check for convergence
        if norm(X_new - X, inf) < tol
            break;
        end

        X = X_new;
        iter = iter + 1;
    end
end
```