

```
function [Xf, operations] = jacobi(A, n, B)
    % Declaring XI, Xf, and operations
    operations = 0;
    Xi = zeros(n,1);
    Xf = zeros(n,1);
    error = 10^(-6);

    % iteration loop
    while 1>0
        for i = 1:n
            sum = 0;
            for j = 1:n
                if(i~=j)
                    sum = sum + (A(i,j)*Xi(j));
                    operations = operations + 1;
                end
            end

            % setting values to Xf by using previous X's
            Xf(i) = (B(i) - sum)/A(i,i);
            operations = operations + 2;
        end

        isConverging = 1; % like boolean value for checking convergence

        % checking convergence for every element
        for k = 1:n
            if abs((Xf(k) - Xi(k))/Xf(k)) > error
                isConverging = 0;
            end
        end

        if(isConverging == 1) % if converging, then break
            break;
        end
        Xi = Xf;
    end
    return
end
```