

```

function [solutionVector] = GaussSeidel( ...
    coefMatrix, ...
    constantVector, ...
    initialGuess, ...
    maxIterations, ...
    errorLevel, ...
    relaxation...
)

% GAUSSSEIDEL This function takes in the coef matrix and constants vector,
% and solves it using the gauss-seidel method

% renaming variables for shorter equations
A = coefMatrix;
B = constantVector;
P = initialGuess;

% N is the number of variables that we are working in
% This works out to be the length of b
N = length(B);

solutionVector = zeros(N,1);
errorVector = zeros(N,1);

% initial guess vector
for j=1:maxIterations
    for i=1:N
        solutionVector(i) = ...
            ((B(i)-(A(i,[1:i-1, i+1:N])*P([1:i-1,i+1:N])))/A(i,i)) * relaxation ...
            + (1 - relaxation) * solutionVector(i) ...
        ;

        % this accounts for the case when P(i) = 0
        if solutionVector(i) == P(i)
            errorVector(i) = 0;
        else
            errorVector(i) = abs(solutionVector(i) - P(i)) / solutionVector(i);
        end
        P(i) = solutionVector(i);
    end
    if max(errorVector) < errorLevel
        disp("Iteration: ")
        disp(j)
        return
    end
end
warning("Warning: GaussSeidel was not able to find a solution within the number of
iterations specified.")
end

```

