```
function [t, Y] = backward_euler_vec(f, tspan, Y0, nsteps)
    t0 = tspan(1);
    tf = tspan(2);
    h = (tf - t0) / nsteps; % Step size
    t = linspace(t0, tf, nsteps + 1); % Time vector
    Y = zeros(length(Y0), nsteps + 1); % Solution matrix
    Y(:, 1) = Y0; % Initial condition

    for i = 1:nsteps
        % The implicit update equation to solve
        func = @(nextY) nextY - Y(:, i) - h*f(t(i) + h, nextY);
        % Solve for the next Y using fsolve
        Y(:, i+1) = fsolve(func, Y(:, i), optimoptions('fsolve', 'Display', 'none'));
    end
end
```