```
1 function x = ImplicitEuler(f, dfdx, T, x0)
     % Returns the iterations of the implicit Euler method
 3
     % f: Function handle
     % Vector field of ODE, i.e., x \text{ dot} = f(t,x)
 4
 5
     % dfdx: Function handle
           Jacobian of f w.r.t. x
 6
 7
     % T: Vector of time points, 1 x Nt
8
     % x0: Initial state, Nx x 1
9
     % x: Implicit Euler iterations, Nx x Nt
10
    11
     % Define variables
12
     % Allocate space for iterations (x)
13
    nx = size(x0,1);
14
    Nt = size(T, 2);
15
16
    x = zeros(nx,Nt);
17
     x(:,1) = x0;
18
19
    20
     xt = x0;
21
    % Loop over time points
22
     for nt=2:Nt
23
         24
        % Update variables
25
        % Define the residual function for this time step
26
         % Define the Jacobian of this residual
27
        % Call your Newton's method function
28
        % Calculate and save next iteration value xt
29
        dt = T(nt) - T(nt-1);
30
31
        r = 0(xt1) xt + dt * f(nt, xt1) - xt1;
32
        drdx = @(xt1) dt * dfdx(nt, xt1) - eye(nx);
33
34
        xt1 = NewtonsMethod(r, drdx, xt);
35
        xt1 = xt1(:,end); % last element is solution
36
37
        x(:,nt) = xt + dt * f(nt, xt1);
38
        xt = x(:,nt);
         39
40
     end
41 end
42
```