```
function x = ImplicitEulerTemplate(f, dfdx, T, x0)
    % Returns the iterations of the implicit Euler method
    % f: Function handle
    %    Vector field of ODE, i.e., x_dot = f(t,x)
    % dfdx: Function handle
    %       Jacobian of f w.r.t. x
    % T: Vector of time points, 1 x Nt
    % x0: Initial state, Nx x 1
    % x: Implicit Euler iterations, Nx x Nt
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Define variables
    % Allocate space for iterations (x)

    Nx = size(x0, 1);
    Nt = size(T, 2);
    x = zeros(Nx, Nt);
    x(:, 1) = x0;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    xt = x0; % initial iteration
    % Loop over time points
    for nt=2:Nt
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Update variables
        % Define the residual function for this time step
        % Define the Jacobian of this residual
        % Call your Newton's method function
        % Calculate and save next iteration value xt

        dt = T(nt) - T(nt - 1);
        tk = T(nt);
        phi = @(x) xt + dt * f(tk, x) - x;
        J_phi = @(x) dt * dfdx(tk, x) - eye(Nx);
        X_newton = NewtonsMethodTemplate(phi, J_phi, xt);
        xt = X_newton(:, end);
        x(:, nt) = xt;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end
end
```

```
Not enough input arguments.

Error in ImplicitEulerTemplate (line 14)
    Nx = size(x0, 1);
```

Published with MATLAB® R2019a