```
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)
    N_x = size(x0,1);
    N_t = size(T,2);
    x = zeros(N_x,N_t);
    %
    %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    x_t = x0; % initial iteration
    x(:,1) = x_t;
    fn = f(T(1),x_t);
    % Loop over time points
    for n_t=2:N_t
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % 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(n_t) - T(n_t -1);
        x(:, n_t) = x(:, n_t - 1) + dt*fn;
        r = @(F) x(:,n_t-1) + dt * f(T(n_t), F)-F;
        J_ie = @(F) dt*dfdx(F) - eye(size(fn,1), N_x);
        [S_ie, infNorm_ie] = NewtonsMethodTemplate(r, J_ie, x(:,n_t));
        x(:,n_t) = S_ie(:,end);
        fn = f(T(n_t), x(:,n_t));
        %
        %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end
end
```

```
Not enough input arguments.

Error in ImplicitEulerTemplate (line 13)
    N_x = size(x0,1);
```

Published with MATLAB® R2019a