TTK4130 MODELING AND SIMULATION	ASSIGNMENT 8
ODIN ALEKSANDER SEVERINSEN	MTTK

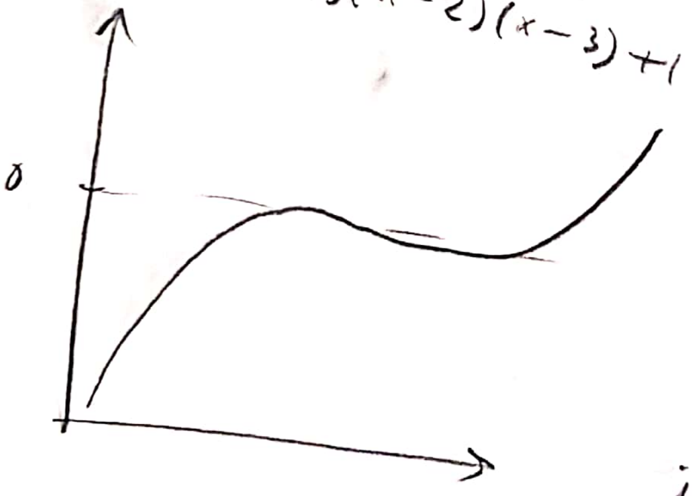
a) See attached code.

b) Rewrites as $f(x) = \begin{bmatrix} xy - 2 \\ \frac{x^4}{4} + \frac{y^3}{3} - 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

See attached code.

The first iteration overshoots correct answer, but quickly converges after this.

c) $f(x) = (x-1)(x-2)(x-3)+1$ looks like this:



Thus, $f'(x) \approx 0$ where $f(x) = 0$. When Newton's approaches 0, ~~it~~ the linearization it does will start to overshoot. This can be fixed with a variable step size.

Scanned with CamScanner

d) The Jacobian It is singular at the root $x^* = \begin{bmatrix} 1 \\ \pi \end{bmatrix}$

Newton converges to the closest root.

e) ~~When the Newton's method gets close to the solution, the gradient approaches 0, which makes each step smaller and smaller, which breaks the convergence.~~

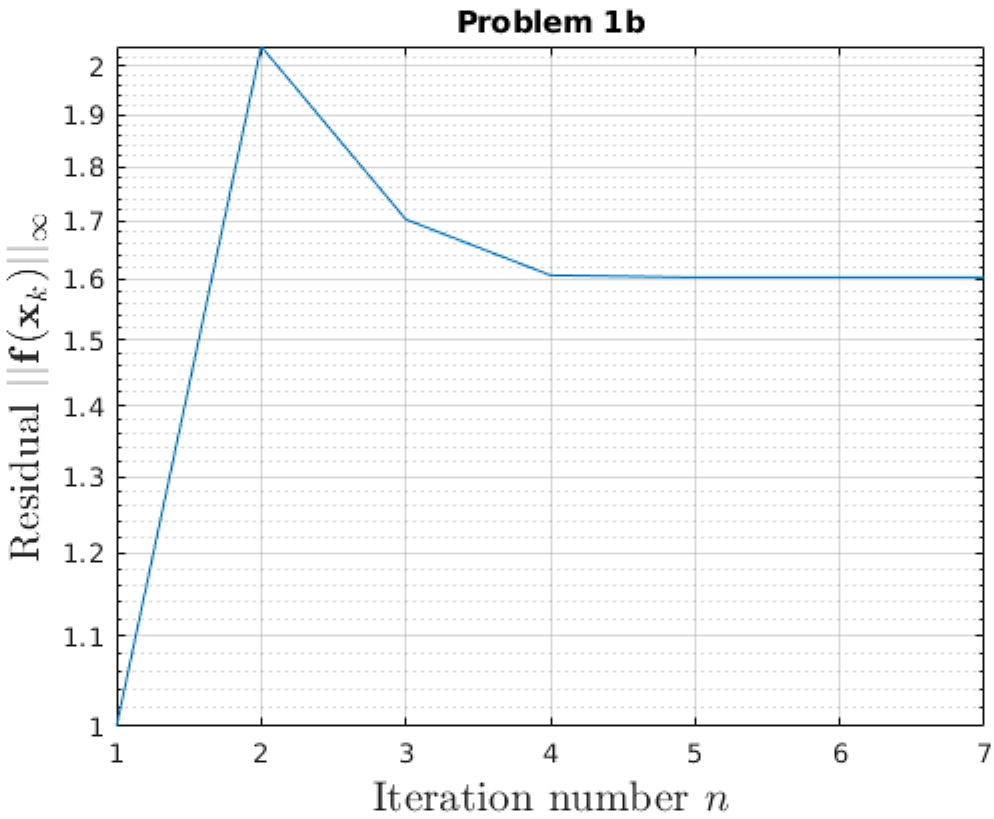
The convergence is quadratic. ~~The~~ Newton's has the benefit over other methods of finding mins, such as gradiant descent, that it uses information in the Hessian to not stall in a zig-zag pattern when it gets close to a solution.

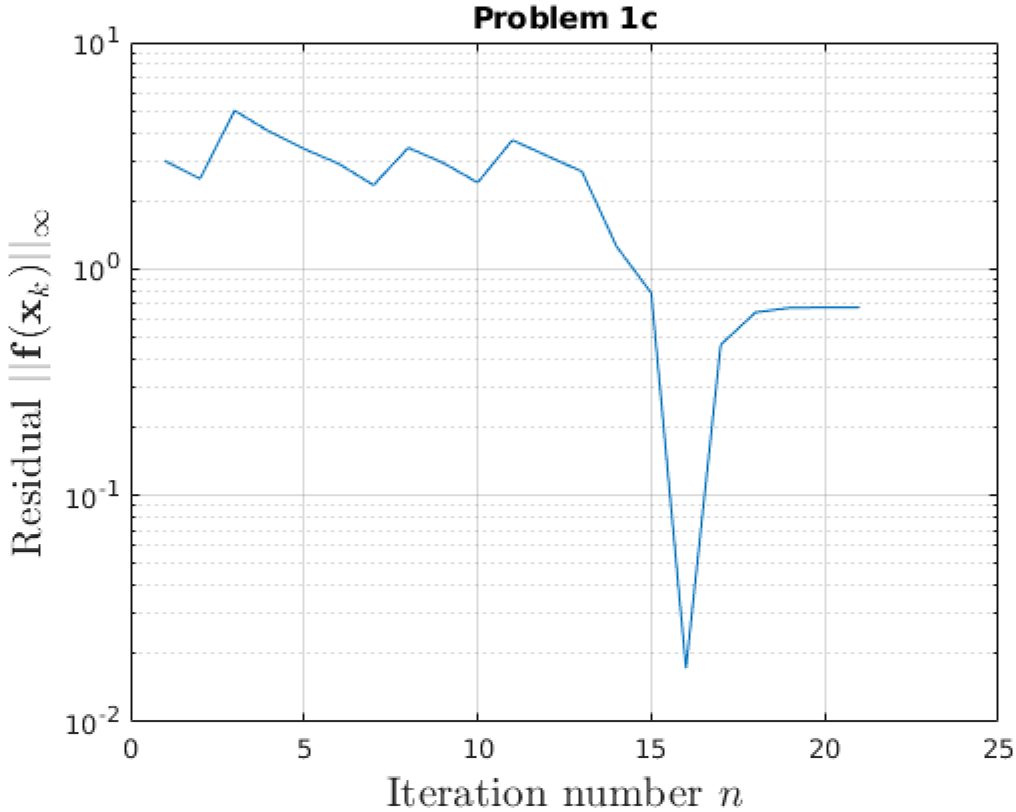
$\boxed{2}$

a) See attached code.

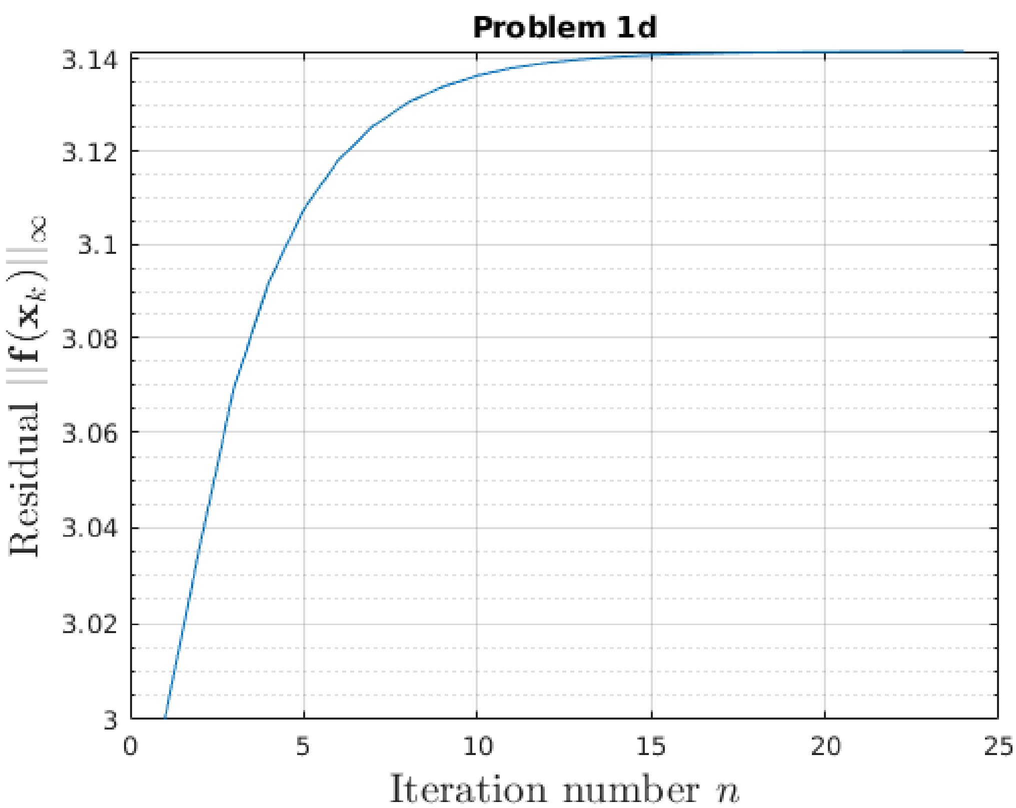
b) The results are pretty good.

```
run('p1b.m');
run('p1c.m');
run('p1d.m');
run('p1e.m');
run('p2b.m');
```

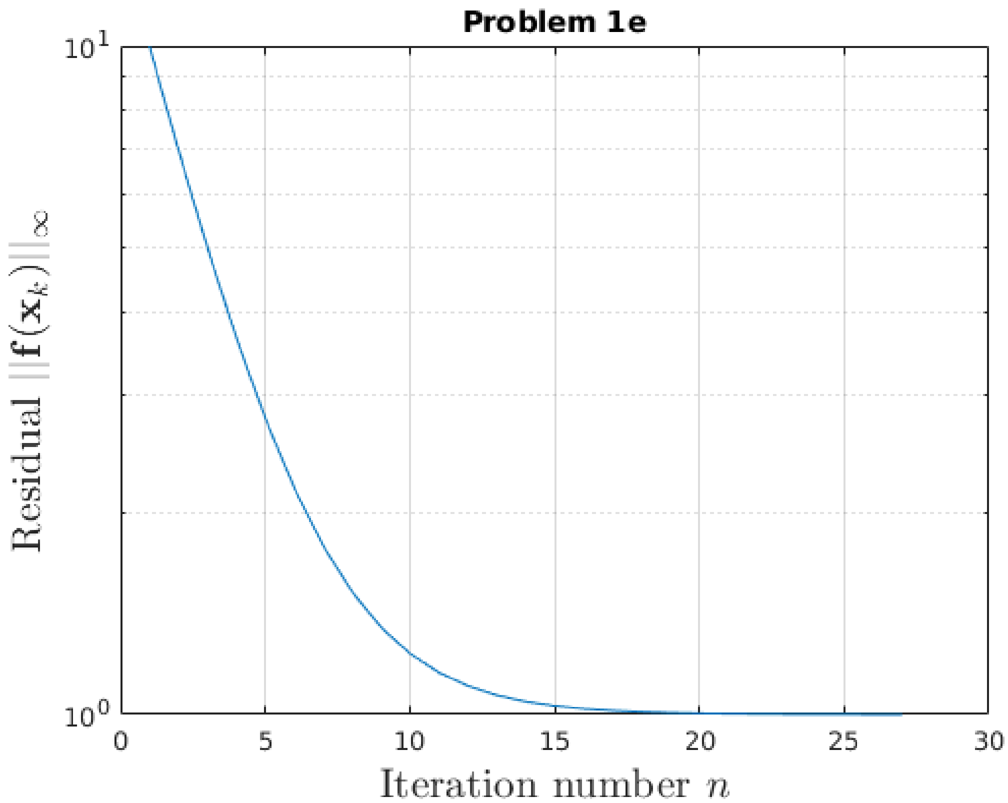


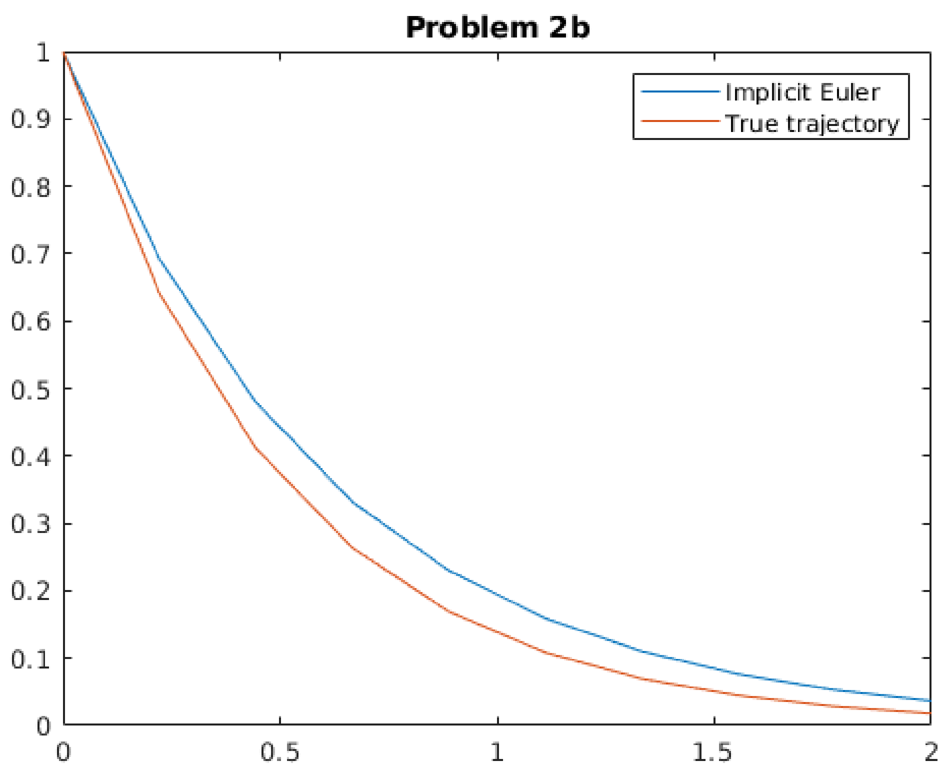


Problem 1c
10^{1}
10^{0}
10^{-1}
10^{-2}
0
5
10
15
20
25
Residual $\|\mathbf{f}(\mathbf{x}_k)\|_\infty$
Iteration number n



Problem 1d
3.14
3.12
3.1
3.08
3.06
3.04
3.02
3
0
5
10
15
20
25
Residual $\|\mathbf{f}(\mathbf{x}_k)\|_\infty$
Iteration number n


Problem 1e
10^1
10^0
0
5
10
15
20
25
30
Residual $\|\mathbf{f}(\mathbf{x}_k)\|_\infty$
Iteration number n


Problem 2b
Implicit Euler
True trajectory
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2

Published with MATLAB® R2019a

```
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

```matlab
function X = NewtonsMethodTemplate(f, J, x0, tol, N)
    % Returns the iterations of the Newton's method
    % f: Function handle
    %    Objective function, i.e. equation f(x)=0
    % J: Function handle
    %    Jacobian of f
    % x0: Initial root estimate, Nx x 1
    % tol: tolerance
    % N: Maximum number of iterations
    if nargin < 5
        N = 100;
    end
    if nargin < 4
        tol = 1e-10;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Define variables
    % Allocate space for iterations (X)

    Nx = size(x0, 1);
    X = zeros(Nx, N);
    X(:) = nan;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    xn = x0; % initial estimate
    n = 1; % iteration number
    fn = f(xn); % save calculation
    X(:, n) = xn;
    % Iterate until f(x) is small enough or
    % the maximum number of iterations has been reached
    iterate = norm(fn,Inf) > tol;
    while iterate
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Calculate and save next iteration value x

        xn = xn - J(xn) \ fn;
        n = n + 1;
        X(:, n) = xn;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        fn = f(xn); % save calculation for next iteration
        % Continue iterating?
        iterate = norm(fn,Inf) > tol && n <= N;
    end

    if norm(fn,Inf) > tol && n > N
        fprintf('Terminated early!\n')
    end

    X(:,~any(~isnan(X), 1))=[];

end
```

```
Not enough input arguments.

Error in NewtonsMethodTemplate (line 20)
    Nx = size(x0, 1);
```

Published with MATLAB® R2019a