
```

%{
Name: modified_newton
Inputs:
    x_0 - the initial iterate
    f - the once differentiable function whose root we are trying to
    find
    f_prime - the derivative of f
    p - the multiplicity of the root we are trying to find
    tol - the tolerance for the relative error in the iterates
    n_max - the maximum number of Newton iterations to perform
Outputs:
    r - the approximate root found by modified Newton's method
    iters
    ier - an error message
        If ier = 0, then Newton's method successfully converged to a
        root
            of f starting from x_0 within n_max iterations
            If ier /= 0, then Newton's method did not successfully
            converge
%}
function [r, iters, ier] = modified_newton(x_0, f, f_prime, p, tol,
n_max)
    iters = 1;
    ier = 1;

    % Set the initial iterate and compute the derivative of f at it
    r = zeros(n_max, 1);
    r(iters) = x_0;
    df = f_prime(r(iters));

    % Perform Newton iterations while the max number of iterations
    % has not been reached yet
    while iters < n_max
        % If the derivative at the current iterate is 0, stop the
        iteration
        if df == 0
            ier = 2;
            break;
        end

        % Compute new iterate via the modified Newton method
        r(iters + 1) = r(iters) - p*f(r(iters))./df;
        iters = iters + 1;

        % If the relative error between the two most recent iterates
        is
        % below the provided tolerance, stop the iteration
        if abs((r(iters) - r(iters-1))/r(iters-1)) < tol
            ier = 0;
            break;
        end
    end

```

```
        % Compute derivative at most recent iterate
        df = f_prime(r(iters));
    end

    % If the max number of iterations were performed, indicate this to
    the
    % user
    if iters == n_max
        ier = 1;
    end
end

Not enough input arguments.

Error in modified_newton (line 23)
    r = zeros(n_max, 1);
```

Published with MATLAB® R2021a