Newton-Raphson line search algorithm

Finding the minimizer(s) of f is resemble to finding the root(s) of f'. From

$$f(x) = 10(x-1)^4 - 4\sin(3x)$$

Then, a derivative of this function w.r.t. x is given here:

$$f'(x) = 40(x-1)^3 - 12\cos(3x)$$

Let $g(x) = f'(x)$. Now, we will use the Newton-Raphson method to find the root of $g(x)$. Then, the formula will be:

$$\begin{aligned} x_{k+1} &= x_k + g(x_k)/g'(x_k) \\ &= x_k + \frac{40(x-1)^3 - 12\cos(3x)}{120(x-1)^2 + 36\sin(3x)} \end{aligned}$$

The `FunctionName.m` is modified to

```
function [f,J] = FunctionName(xin)

    % Declare the functions.
    f_fun = @(x) 40*(x-1)^3 - 12*cos(3*x);
    J_fun = @(x) 120*(x-1)^2 + 36*sin(3*x);

    % Evaluate numerical values.
    f = f_fun(xin);
    J = J_fun(xin);

end
```

The `newton.m` is provided here.

```
function [xsolution, Xk, Fk, Jk, IFLAG, IterationsUsed] = newton_1var(FunctionName, x0, epsilon, IterationMax)

    Xk = x0; % list to store x_k.
    Fk = []; % list to store f_k.
    Jk = []; % list to store J_k.
    IFLAG = 0; % Flag to indicate whether x_k converge.

    for i = 1:IterationMax
        [f, J] = FunctionName(x0); % generate f and J matrix at x0.
        s = J\(-f); % solve for step value s.
        x1 = x0 + s; % find next x_k.

        Xk(:,i+1) = x1; % store new x_k.
        Fk(:,i) = f; % store new f_k.
        Jk(:,:,i) = J; % store new J_k.

        if norm(x1-x0) <= epsilon % stop if the step is small enough.
            IterationsUsed = i; % return a number of iterations.
            xsolution = x1; % return the root of function.
            [f, J] = FunctionName(x1); % finding new f, J final value.
            Fk(:,i+1) = f; % store last f_k.
            Jk(:,:,i+1) = J; % store last J_k.
            break
        end

        x0 = x1; % set next x_k to be x_(k+1).
    end

    if IterationsUsed == IterationMax % to indicate x_k is not converge.
        IFLAG = 1;
    end

end
```

The result is printed by this script.

```
[xsolution, Xk, Fk, Jk, IFLAG, IterationsUsed] = newton_1var(@FunctionName, 1, 1e-10, 100);

% report the values as a table.
fprintf('% 5s % 20s % 10s \n', 'Iter', 'x_k', 'f_k');
for i = 1:IterationsUsed+1
    fprintf('% 5.2d % 20.10f % 10.3f \n', i, Xk(i), Fk(i));
end
```

And the printed result is written below:

```
 Iter                  x_k         f_k
   01         1.0000000000      11.880
   02        -1.3384175171    -503.773
   03        -0.6016751899    -161.570
   04        -0.0094674957     -53.142
   05         0.4287791259     -10.823
   06         0.5756154046      -1.192
   07         0.5964689434      -0.026
   08         0.5969428628      -0.000
   09         0.5969431100      -0.000
   10         0.5969431100      -0.000
```

So, the value of a minimizer of f is 0.5969431100.

To determine whether f has only one minimizer, let's analyze from the function derivative:

$$f'(x) = 40(x-1)^3 - 12\cos(3x)$$

Plot the graph of $40(x-1)^3$ and $12\cos(3x)$ in the same area.

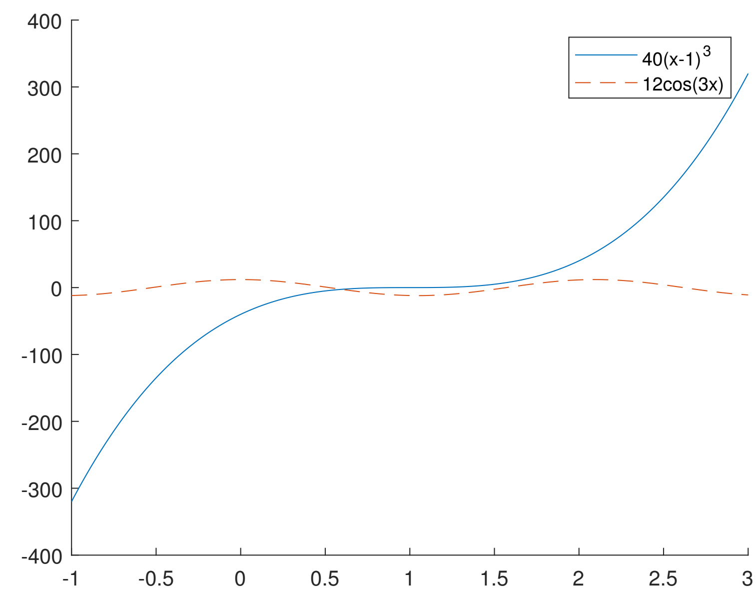

Figure 1: The graph of $40(x-1)^3$ and $12\cos(3x)$.

From the graph, for $x < 0.597$, $40(x-1)^3 < 12\cos(3x)$ so $f'(x)$ is always negative. For $x \geq 0.597$, $40(x-1)^3 \geq 12\cos(3x)$ so $f'(x)$ is always positive. Therefore $f'(x)$ has only one root, then $f(x)$ has only one minimizer.