```
function [f_all, gnorm_all, m, M, L] = newtmeth(fun, x0, tol, maxit)
% Newton's method with backtracking line search.
% Also estimate the values of m, M, and L using several points.
%
% Author: Terrence Alsup
% Date: March 10, 2020
% File: newtmeth.m


% Get the starting values.
x = x0;
[f, g, H] = fun(x);
H0 = H;
L = 0;
f_all(1) = f;
gnorm = norm(g);
gnorm_all(1) = gnorm;

evals = eig(H);
m = min(evals);
M = max(evals);

% The main Newton iteration and check the stopping criteria.
it = 1;
while it <= maxit && gnorm > tol

    % Compute the Newton search direction.
    dx = -H\g;

    t = linesearch(fun, x, dx);  % Backtracking linesearch for step
 length.
    x = x + t*dx;                % The Newton update.

    % Recompute the current value of the objective function.
    [f, g, H] = fun(x);

    % Compute the min and max eigenvalues of the Hessian.
    evals = eig(H);
    m = min([min(evals), m]);
    M = max([max(evals), M]);

    % Estimate the Lipschitz constant.
    L = max([L, norm(H - H0)/norm(x - x0)]);

    % Save the values.
    gnorm = norm(g);
    f_all(it + 1) = f;
    gnorm_all(it + 1) = gnorm;

    it = it + 1; % Increment the iteration.
end
```

```
end
```

Published with MATLAB® R2019b