
```

function run_full_bfgs(n_values,toll,max_iters)
    for i = 1:length(n_values)
        n = n_values(i);
        x0 = 2 * ones(n, 1);
        B0 = eye(n);

        % Rosenbrock function, gradient, and Hessian
        [f, g, H] = rosenbrock(x0);

        % Run the BFGS algorithm
        [x, k, g_norms] = bfgs_backtracking(f, g, x0, B0, toll, max_iters);

        disp(['Dimension n = ', num2str(n)]);
        disp(['Number of iterations: ', num2str(k)]);
        disp(['Final point x: ', mat2str(x', 4)]);

        % Plotting the convergence rate (semi-log plot for gradient norms)
        figure;
        semilogy(1:length(g_norms), g_norms);
        title(['Convergence rate for n = ', num2str(n)]);
        xlabel('kth iteration');
        ylabel('Gradient norm');

        % Plotting the ratio of consecutive gradient norms
        g_ratio = g_norms(2:end) ./ g_norms(1:end-1);
        figure;
        plot(1:length(g_ratio), g_ratio);
        title(['Gradient norm ratio for n = ', num2str(n)]);
        xlabel('kth iteration');
        ylabel('$$\frac{||g_{k+1}||}{||g_k||}$$', 'interpreter', 'latex');
    end
end

function [f, g, H] = rosenbrock(x0)
    syms x [length(x0) 1];
    f_sym = 0;
    for i = 1:(length(x) - 1)
        f_sym = f_sym + 100*(x(i+1) - x(i)^2)^2 + (1 - x(i))^2;
    end

    grad_sym = gradient(f_sym, x);
    Hess_sym = hessian(f_sym, x);

    f = matlabFunction(f_sym, 'Vars', {x});
    g = matlabFunction(grad_sym, 'Vars', {x});
    H = matlabFunction(Hess_sym, 'Vars', {x});
end

function [x, k, g_norms] = bfgs_backtracking(f, g, x0, B0, toll, max_iters)
    x = x0;
    B = B0;
    k = 0;

```

```

g_norms = []; % Initialize an array to store the norm of the gradients

while true
    gx = g(x);
    fx = f(x);
    g_norms = [g_norms; norm(gx)]; % Store the current gradient norm

    if norm(gx) <= toll
        break;
    end

    pk = -B * gx;
    alpha = 1;
    c1 = 0.1;
    rho = 0.5;

    % Backtracking line search
    while true
        x_next = x + alpha * pk;
        f_next = f(x_next); % Get the next function value
        if f_next > fx + c1 * alpha * gx' * pk
            alpha = rho * alpha;
        else
            break;
        end
    end

    g_next = g(x_next);
    s = x_next - x;
    y = g_next - gx;

    if s' * y >= 0.2 * s' * B * s
        theta_k = 1;
    else
        theta_k = (0.8 * s' * B * s) / (s' * B * s - s' * y);
    end

    r = theta_k * y + (1 - theta_k) * (B * s);

    rho = 1 / (y' * s);
    B = (eye(length(x0)) - rho * s * y') * B * (eye(length(x0)) - rho * y
    * s') + rho * s * s';

    x = x_next;
    k = k + 1;

    if k >= max_iters
        break;
    end
end
end

```

Not enough input arguments.

Error in run_full_bfgs (line 2)
for i = 1:length(n_values)

Published with MATLAB® R2023b