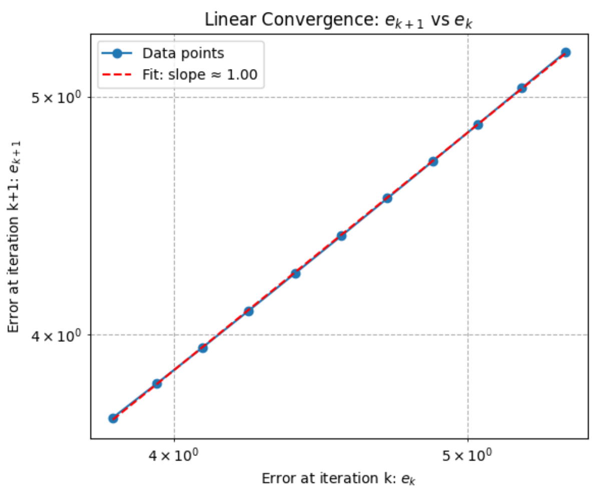
## Numerical Optimization – Homework 6

Problem 1: Based on the code linesearch_steepest.html from Week 4 and your own choice of function f, design a numerical experiment that concludes that the rate of convergence p of the steepest descent method is 1.

Answer:

In this problem, I used the steepest descent method on the Rosenbrock function to check its rate of convergence. The initial point was set to $(-4,3)$ and a fixed step size of $\alpha = 0.2$ was used. The goal was to see how the error between the current point and the true minimum (1,1) changes in each iteration and to find whether the convergence rate is equal to 1. To test this, I calculated the error at each iteration and plotted $e_{k+1}$ versus $e_k$ on a log-log scale. If the method converges linearly, we expect the errors to follow $e_{k+1} = Ce_k$. By taking the logarithm, this becomes a straight line with slope 1. From my plot, the fitted line had a slope of about 1.00, which confirms that the steepest descent method converges linearly. This means that in each step, the error decreases by roughly the same factor.




Problem 2: Based on the code linesearch_newton.html from Week 4 and your own choice of function f, design a numerical experiment that concludes that the rate of convergence p of the Newton's method is 2.

Answer:

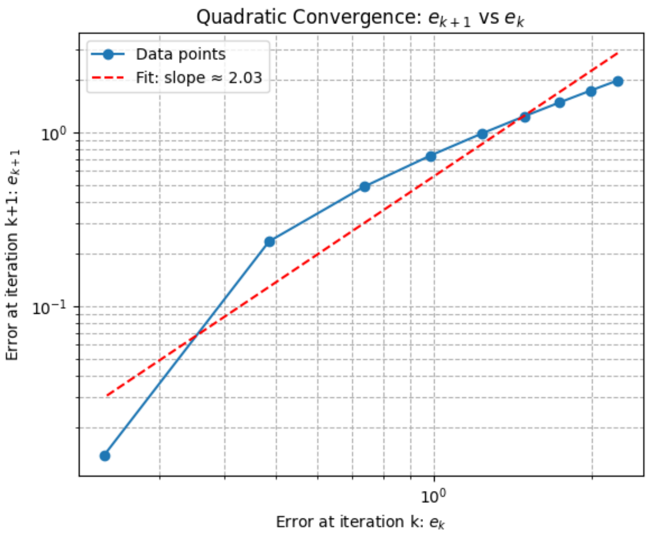
The goal of this problem is to confirm that Newton's method converges quadratically to the true minimum of a smooth function. Quadratic convergence means that once the iterations are close enough to the minimum, the error between consecutive iterations satisfies the relation: $e_{k+1} = Ce_k^2$ or equivalently, we can take the logarithm of both sides of the equation:

$$\log(e_{k+1}) = \log(C) + 2\log(e_k)$$

This equation shows that if we plot $\log(e_{k+1})$ versus $\log(e_k)$, the data should lie roughly on a straight line whose slope is 2. Therefore, by performing a simple linear fit on the log-log data, we can estimate the rate of convergence from the slope of the line.

In this experiment, the chosen objective function was a quartic bowl which has its global minimum at (1, -2). The gradient and Hessian were derived analytically and used in the Newton update rule. The initial guess was set to (2,−4) and a step size of $\alpha = 0.25$ was used. During each iteration, the current point, function value, and distance to the true minimum were recorded. The errors $e_k$ and $e_{k+1}$ were plotted against each other on a logarithmic scale to visualize the convergence pattern.

Result:




In this plot the slope of the fitted line was about 2.03, which is very close to 2. This confirms that the method has quadratic convergence, as expected for Newton's method. It also means that once the algorithm gets close enough to the minimum, the error roughly squares at every step. For example, if one step has an error of 0.1, the next step's error becomes around 0.01. From the plot, we can also see that the early steps (when the point is still far from the minimum) don't follow the line perfectly, meaning the convergence is slower at first. But after a few iterations, the points line up nicely, and the convergence becomes very fast. This shows the transition from the slow early phase to the rapid, quadratic phase that happens near the true solution.

Problem 3: Implementing the full Wolfe condition to select the step length

Answer:

The full Wolfe condition contains two parts:

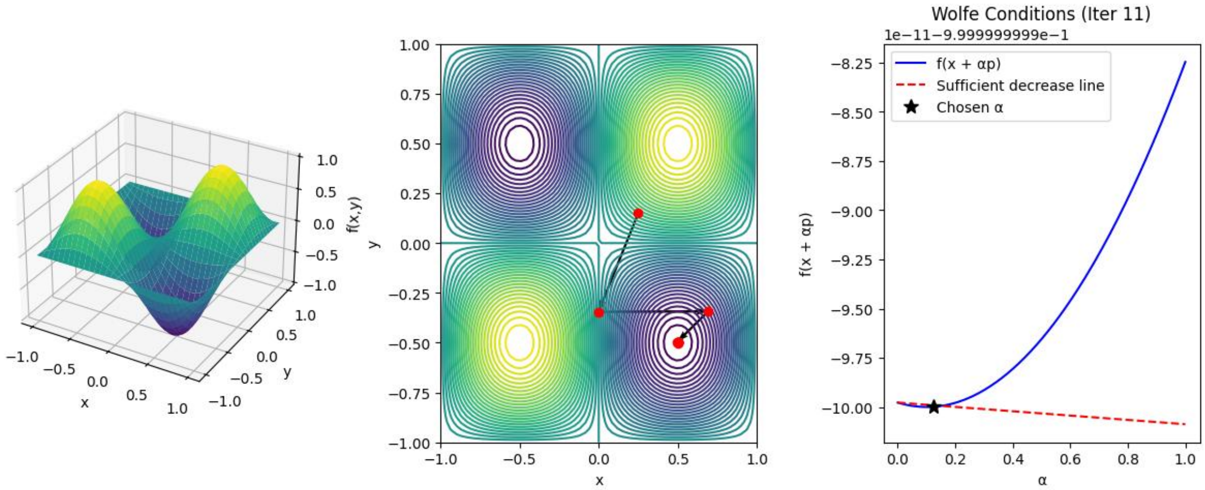
1. Sufficient decrease condition: ensures the function decreases enough
2. Curvature condition: ensures that the slope has flattened sufficiently at the new point

In order to implement this algorithm, I followed these steps:

1. Defining the function and gradient: I used $f(x, y) = \sin(\pi x)\sin(\pi y)$ which is a non-convex function with many peaks and valleys. The gradient is computed analytically for both variables and is used to find the steepest descent direction.
2. Setting up the plots: I created three plots: a 3D surface, a contour map, and a Wolfe condition plot, to visualize the optimization process and how the step length is chosen at each iteration.
3. Initialization: The starting point is $x_k = [0.25, 0.15]$. I also set the Wolfe parameters $c_1 = 0.25$ and $c_2 = 0.9$ with the maximum possible step size $\alpha = 1$
4. Main iteration loop: In this part I used a while loop that keeps doing iterations as long as the stopping criteria are satisfied.

5. Updating the point: Once the correct $\alpha_k$ is found, the next point is updated as $x_{k+1} = x_k + \alpha_k p_k$ and plotted on the contour map as a red dot with an arrow showing the step direction.
6. Printing results: Each iteration printed the current coordinates, function value, and the chosen step length, so I could check how the step size changed across iterations.

<u>Results:</u>




1. 3D Surface plot: This plot shows the overall shape of the objective function. It has multiple hills and valleys that repeat periodically because of the sine terms. From this plot, we can see that the optimizer can get stuck in one of the local minima depending on where it starts.
2. Contour plot with iterations: This plot is a top-down view of the same function. Each contour line connects points with equal function values, so darker areas represent valleys (lower f values). The red dots represent successive points, and the black arrows show the directions of motion and the computed step length. We can see that the points gradually move downhill toward a nearby valley, showing that the Wolfe conditions successfully control the step size to ensure smooth convergence.
3. Wolfe condition plot: This plot visualizes how the step length is chosen in one iteration. The blue curve shows how $f(x + \alpha p)$ changes as we vary $\alpha$ along the search direction. The red dashed line represents the sufficient decrease condition, which defines the region where the function decreases enough to be acceptable. The black star marks the chosen α that satisfies both the Armijo condition (enough decrease) and the curvature condition (slope small enough). This plot proves that the selected α is valid according to the Wolfe criteria; the blue curve at that point lies below the red line, and the slope is sufficiently flat.