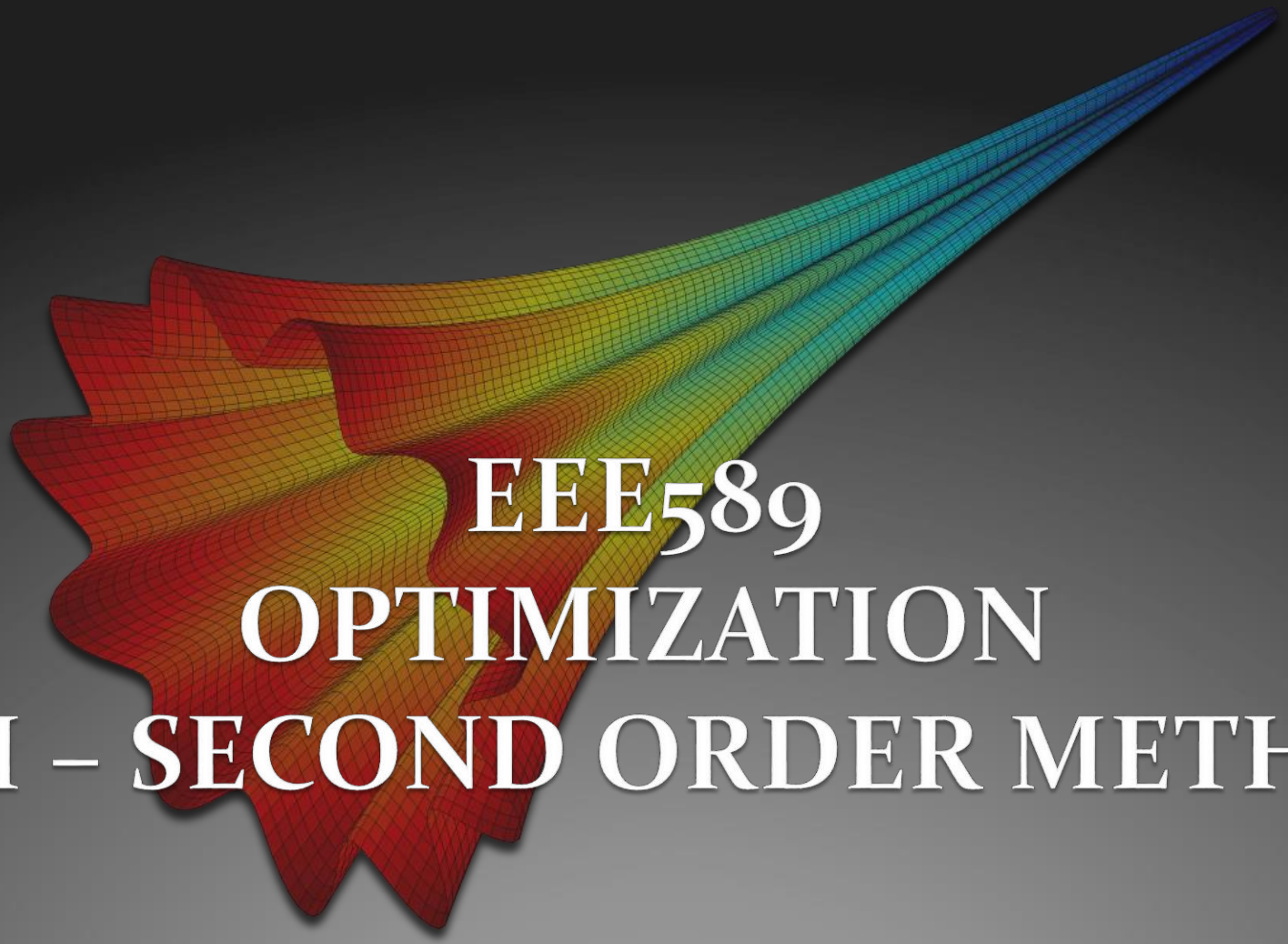


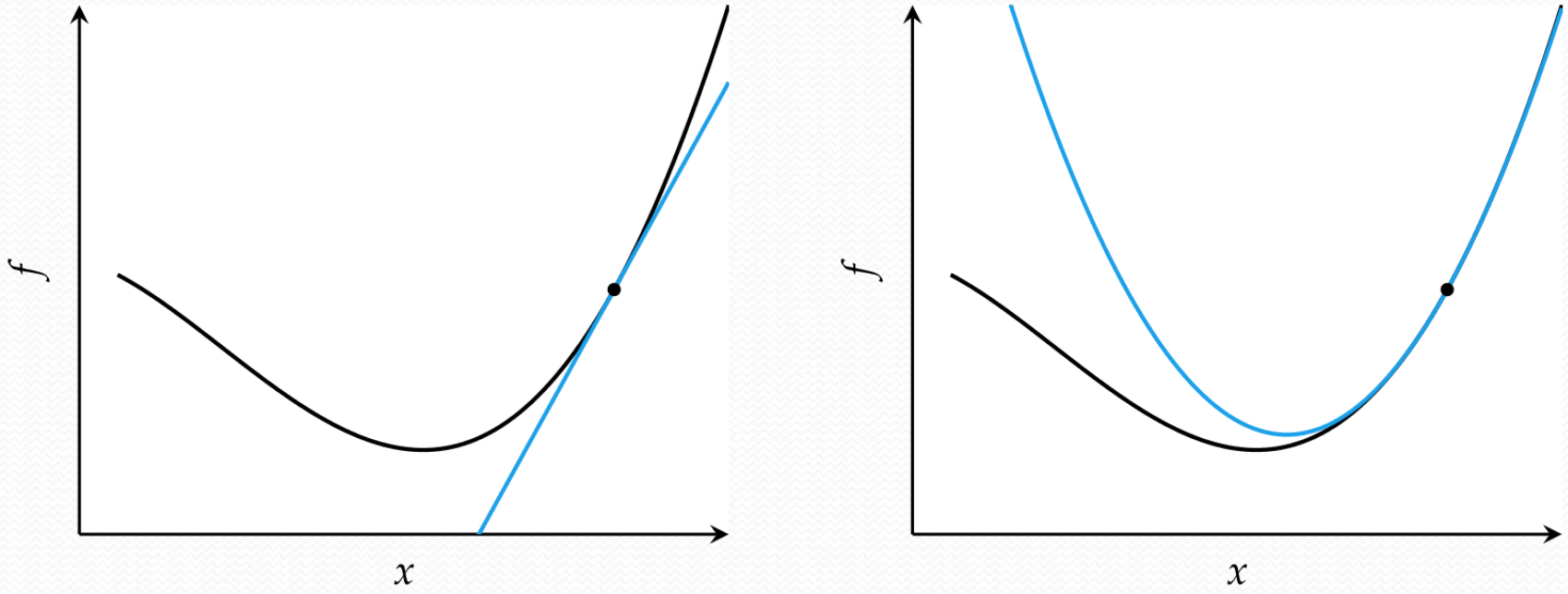
EEE589
OPTIMIZATION
CH VI – SECOND ORDER METHODS

Second-Order Methods

- This chapter focuses on leveraging *second-order* approximations that use the second derivative in univariate optimization or the Hessian in multivariate optimization to direct the search.
- This additional information can help improve the local model used for informing the selection of directions and step lengths in descent algorithms.

Second-Order Methods

- A comparison of firstorder and second-order approximations. Bowl-shaped quadratic approximations have unique locations where the derivative is zero.



Newton's Method

- Knowing the function value and gradient for a design point can help determine the direction to travel, but this first-order information does not directly help determine how far to step to reach a local minimum.
- Second-order information, on the other hand, allows us to make a quadratic approximation of the objective function and approximate the right step size to reach a local minimum.
- As we have seen with quadratic fit search in chapter 3, we can analytically obtain the location where a quadratic approximation has a zero gradient.
- We can then use that location as the next iteration to approach a local minimum.

Newton's Method

- In univariate optimization, the quadratic approximation about a point $x^{(k)}$ comes from the second-order Taylor expansion:

$$q(x) = f(x^{(k)}) + (x - x^{(k)})f'(x^{(k)}) + \frac{(x - x^{(k)})^2}{2}f''(x^{(k)})$$

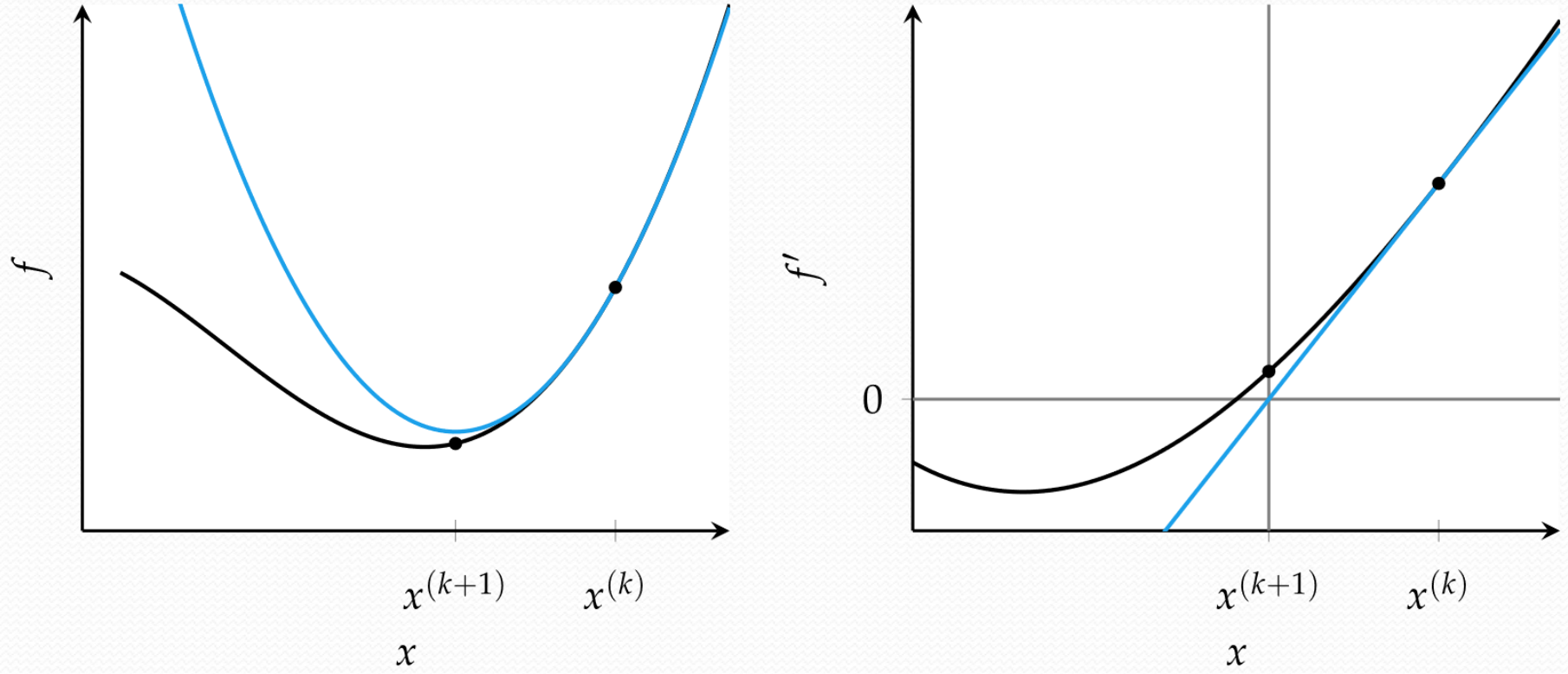
- Setting the derivative to zero and solving for the root yields the update equation for *Newton's method*:

$$\frac{\partial}{\partial x}q(x) = f'(x^{(k)}) + (x - x^{(k)})f''(x^{(k)}) = 0$$

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

Newton's Method

- This update is shown in figure. **Newton's method for optimization is equivalent to finding the roots of the derivative function**

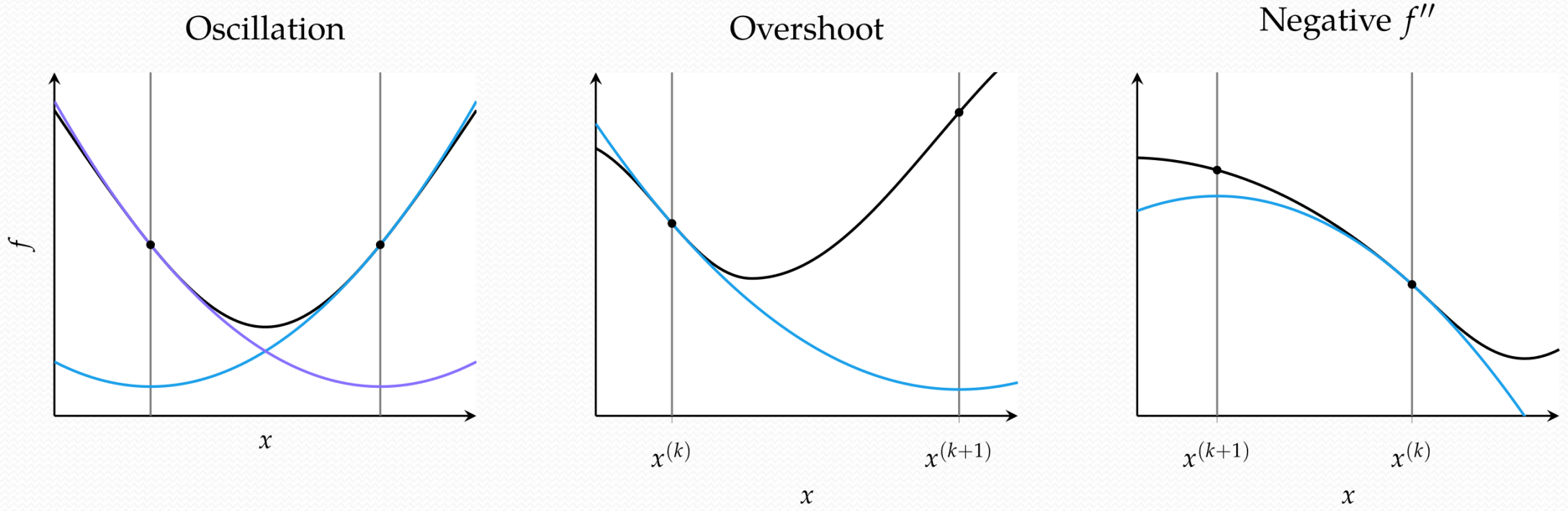


Newton's Method

- The update rule in Newton's method involves dividing by the second derivative.
- The update is undefined if the second derivative is zero, which occurs when the quadratic approximation is a horizontal line.
- Instability also occurs when the second derivative is very close to zero, in which case the next iterate will lie very far from the current design point, far from where the local quadratic approximation is valid.
- Poor local approximations can lead to poor performance with Newton's method.

Newton's Method

- Figure shows three kinds of failure cases. **Common causes of error in Newton's method**



Newton's Method

- Newton's method does tend to converge quickly when in a bowl-like region that is sufficiently close to a local minimum.
- It has *quadratic convergence*, meaning the difference between the minimum and the iterate is approximately squared with every iteration.
- This rate of convergence holds for Newton's method starting from $x^{(1)}$ within a distance δ of a root x^* if
  - $f''(x) \neq 0$ for all points in I,
  - $f'''(x)$ is continuous on I, and
  - $\frac{1}{2}\left|\frac{f'''(x^{(1)})}{f''(x^{(1)})}\right| < c\left|\frac{f'''(x^*)}{f''(x^*)}\right|$ for some $c < \infty$

  for an interval $I = [x^* - \delta, x^* + \delta]$.

The final condition guards against overshoot.

Newton's Method

- Newton's method can be extended to multivariate optimization (algorithm 6.1). The multivariate second-order Taylor expansion at $x^{(k)}$ is:

$$f(\mathbf{x}) \approx q(\mathbf{x}) = f(\mathbf{x}^{(k)}) + (\mathbf{g}^{(k)})^\top(\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})^\top \mathbf{H}^{(k)}(\mathbf{x} - \mathbf{x}^{(k)})$$

where $g^{(k)}$ and $H^{(k)}$ are the gradient and Hessian at $x^{(k)}$, respectively.

- We evaluate the gradient and set it to zero:

$$\nabla q(\mathbf{x}^{(k)}) = \mathbf{g}^{(k)} + \mathbf{H}^{(k)}(\mathbf{x} - \mathbf{x}^{(k)}) = \mathbf{0}$$

- We then solve for the next iterate, thereby obtaining Newton's method in multivariate form:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{H}^{(k)})^{-1}\mathbf{g}^{(k)}$$

Newton's Method

- If f is quadratic and its Hessian is positive definite, then the update converges to the global minimum in one step.
- For general functions, Newton's method is often terminated once x ceases to change by more than a given tolerance.
- Newton's method can also be used to supply a descent direction to line search or can be modified to use a step factor. Smaller steps toward the minimum or line searches along the descent direction can increase the method's robustness.The descent direction is:

$$\mathbf{d}^{(k)} = -\left(\mathbf{H}^{(k)}\right)^{-1}\mathbf{g}^{(k)}$$

- Next example shows how Newton's method can be used to minimize a function.

With $\mathbf{x}^{(1)} = [9, 8]$, we will use Newton's method to minimize Booth's function:

$$f(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

The gradient of Booth's function is:

$$\nabla f(\mathbf{x}) = [10x_1 + 8x_2 - 34, 8x_1 + 10x_2 - 38]$$

The Hessian of Booth's function is:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 10 & 8 \\ 8 & 10 \end{bmatrix}$$

The first iteration of Newton's method yields:

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \left(\mathbf{H}^{(1)}\right)^{-1} \mathbf{g}^{(1)} = \begin{bmatrix} 9 \\ 8 \end{bmatrix} - \begin{bmatrix} 10 & 8 \\ 8 & 10 \end{bmatrix}^{-1} \begin{bmatrix} 10 \cdot 9 + 8 \cdot 8 - 34 \\ 8 \cdot 9 + 10 \cdot 8 - 38 \end{bmatrix}$$

$$= \begin{bmatrix} 9 \\ 8 \end{bmatrix} - \begin{bmatrix} 10 & 8 \\ 8 & 10 \end{bmatrix}^{-1} \begin{bmatrix} 120 \\ 114 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

The gradient at $\mathbf{x}^{(2)}$ is zero, so we have converged after a single iteration. The Hessian is positive definite everywhere, so $\mathbf{x}^{(2)}$ is the global minimum.

Secant Method

- Newton's method for univariate function minimization requires the first and second derivatives f' and f''.
- In many cases, f' is known but the second derivative is not.
- The *secant method* applies Newton's method using estimates of the second derivative and thus only requires f'.
- This property makes the secant method more convenient to use in practice.

Secant Method

- The secant method uses the last two iterates to approximate the second derivative:

$$f''(x^{(k)}) \approx \frac{f'(x^{(k)}) - f'(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

- This estimate is substituted into Newton's method:

$$x^{(k+1)} \leftarrow x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f'(x^{(k)}) - f'(x^{(k-1)})} f'(x^{(k)})$$

- The secant method requires an additional initial design point. It suffers from the same problems as Newton's method and may take more iterations to converge due to approximating the second derivative.

Quasi-Newton Methods

- Just as the secant method approximates f ″ in the univariate case, *quasi-Newton* methods approximate the inverse Hessian. Quasi-Newton method updates have the form:

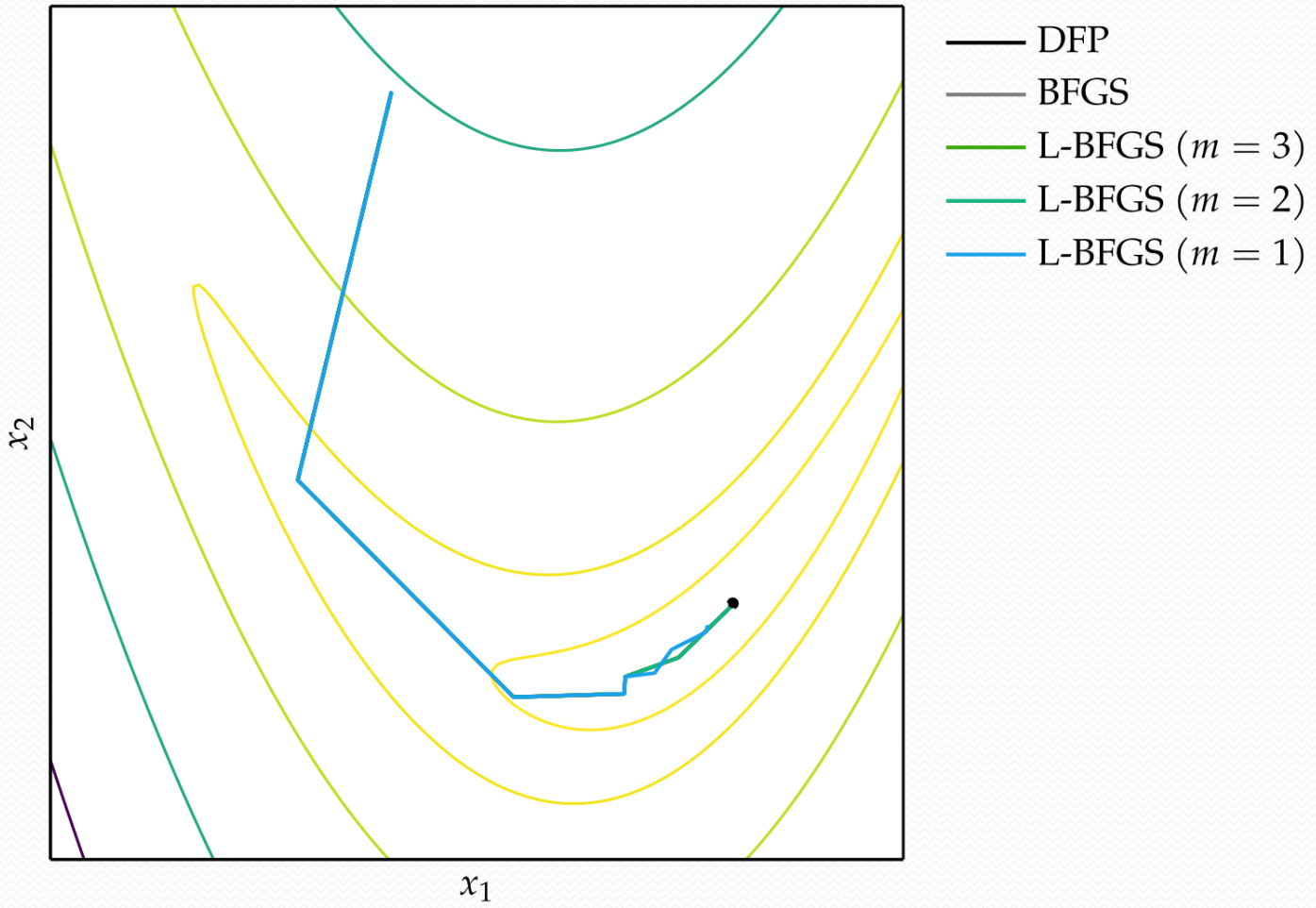
$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - \alpha^{(k)} \mathbf{Q}^{(k)} \mathbf{g}^{(k)}$$

where $\alpha^{(k)}$ is a scalar step factor and $\mathbf{Q}^{(k)}$ approximates the inverse of the Hessian at $\mathbf{x}^{(k)}$.

Quasi-Newton Methods

- For multivariate functions, the inverse Hessian may not be available or infeasible to compute, so can be approximated using a variety of Quasi-Newton methods, each appropriate in different circumstances
- Davidon-Fletcher-Powell (DFP) method
- Broyden-Fletcher-Goldfarb-Shanno (BFGS) method
- Limited-memory BFGS (L-BFGS) method

Quasi-Newton Methods



Summary

- Incorporating second-order information in descent methods often speeds convergence.
- Newton's method is a root-finding method that leverages second-order information to quickly descend to a local minimum.
- The secant method and quasi-Newton methods approximate Newton's method when the second-order information is not directly available.