# Overview

# Finding Stationary Points

the problem of finding a stationary point of a function is the same as finding a zero of the first derivative of that function

# Finding Stationary Points



the problem of finding a stationary point of a function is the same as finding a zero of the first derivative of that function

the methods we will study are very similar to root-finding methods

# Finding Stationary Points



the problem of finding a stationary point of a function is the same as finding a zero of the first derivative of that function

the methods we will study are very similar to root-finding methods

special care must be taken if we are only looking for minima or maxima

# Golden Section Search

1. choose two points $(a, b)$ that bracket a minimum;

# Golden Section Search



1. choose two points $(a, b)$ that bracket a minimum;
2. choose two probe points $(c, d)$ on the bracket;

# Golden Section Search

1. choose two points $(a, b)$ that bracket a minimum;
2. choose two probe points $(c, d)$ on the bracket;
3. proceed with whichever interval $(a, d)$ or $(c, b)$ brackets the minimum;

# Golden Section Search



1. choose two points $(a, b)$ that bracket a minimum;
2. choose two probe points $(c, d)$ on the bracket;
3. proceed with whichever interval $(a, d)$ or $(c, b)$ brackets the minimum;
4. continue until $|a - b| < \delta$

# Golden Section Search



1. choose two points $(a, b)$ that bracket a minimum;
2. choose two probe points $(c, d)$ on the bracket;
3. proceed with whichever interval $(a, d)$ or $(c, b)$ brackets the minimum;
4. continue until $|a - b| < \delta$

Q. We want a reliable, efficient method; how should we choose the points $(c, d)$?

# Golden Section Search



Q. We want a reliable, efficient method; how should we choose the points $(c, d)$?

# Golden Section Search

Q. We want a reliable, efficient method; how should we choose the points $(c, d)$?

A1. Such that $(a, d)$ and $(c, b)$ have the same width (reliable),

$$c = b - \frac{b-a}{x}; \quad d = a + \frac{b-a}{x}$$

# Golden Section Search

Q. We want a reliable, efficient method; how should we choose the points $(c, d)$?

A1. Such that $(a, d)$ and $(c, b)$ have the same width (reliable),

$$c = b - \frac{b-a}{x}; \quad d = a + \frac{b-a}{x}$$

A2. and such $c$ is a probe point on $(a, d)$, and that $d$ is a probe point on $(c, b)$ (efficient),

$$\frac{b-a}{d-a} = \frac{d-a}{c-a}$$

# Golden Section Search



Q. We want a reliable, efficient method; how should we choose the points $(c, d)$?

A1. Such that $(a, d)$ and $(c, b)$ have the same width (reliable),

$$c = b - \frac{b-a}{x}; \quad d = a + \frac{b-a}{x}$$

A2. and such $c$ is a probe point on $(a, d)$, and that $d$ is a probe point on $(c, b)$ (efficient),

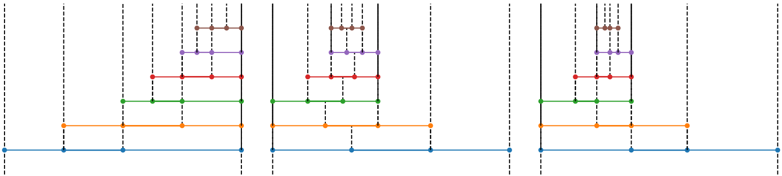
$$\frac{b-a}{d-a} = \frac{d-a}{c-a}$$

the conditions are satisfied when

$$x = \frac{\sqrt{5}+1}{2},$$

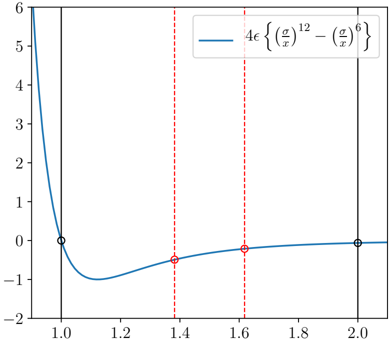
the golden ratio, hence the name, "golden section search."

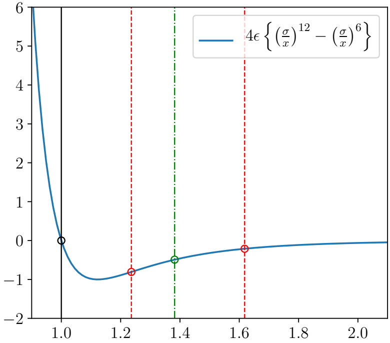
# Golden Section Search



- Uneven intervals (left), even, but suboptimal intervals (centre), and golden section search (right)

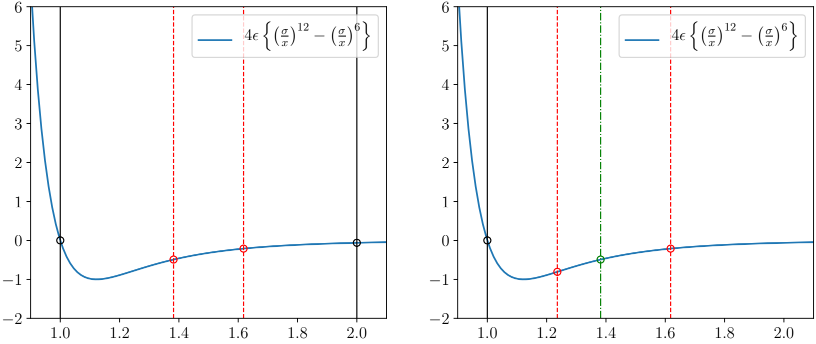
# Golden Section Search




- One iteration of the golden section search applied to the Lennard-Jones potential.

# Golden Section Search




- One iteration of the golden section search applied to the Lennard-Jones potential.
- The "c" point in the first bracket is reused as the "d" point in the second bracket.

# Successive parabolic interpolation



1. choose three points $(a, b, c)$ that need not bracket a stationary point;

# Successive parabolic interpolation

1. choose three points $(a, b, c)$ that need not bracket a stationary point;
2. find the unique parabola to the curve passing through those points,

$$p(x) = \frac{(x-b)(x-c)}{(a-b)(a-c)} f_a + \frac{(x-a)(x-c)}{(b-a)(b-c)} f_b + \frac{(x-a)(x-b)}{(c-a)(c-b)} f_c$$

# Successive parabolic interpolation



1. choose three points $(a, b, c)$ that need not bracket a stationary point;
2. find the unique parabola to the curve passing through those points,

$$p(x) = \frac{(x-b)(x-c)}{(a-b)(a-c)} f_a + \frac{(x-a)(x-c)}{(b-a)(b-c)} f_b + \frac{(x-a)(x-b)}{(c-a)(c-b)} f_c$$

3. find the unique stationary point of the parabola,

$$d = \frac{1}{2} \frac{a^2(f_c - f_b) + b^2(f_a - f_c) + c^2(f_b - f_a)}{a(f_c - f_b) + b(f_a - f_c) + c(f_b - f_a)}$$

# Successive parabolic interpolation



1. choose three points $(a, b, c)$ that need not bracket a stationary point;
2. find the unique parabola to the curve passing through those points,

$$p(x) = \frac{(x-b)(x-c)}{(a-b)(a-c)} f_a + \frac{(x-a)(x-c)}{(b-a)(b-c)} f_b + \frac{(x-a)(x-b)}{(c-a)(c-b)} f_c$$

3. find the unique stationary point of the parabola,

$$d = \frac{1}{2} \frac{a^2 (f_c - f_b) + b^2 (f_a - f_c) + c^2 (f_b - f_a)}{a (f_c - f_b) + b (f_a - f_c) + c (f_b - f_a)}$$

4. $(a, b, c) \leftarrow (b, c, d)$

# Successive parabolic interpolation

1. choose three points $(a, b, c)$ that need not bracket a stationary point;
2. find the unique parabola to the curve passing through those points,

$$p(x) = \frac{(x-b)(x-c)}{(a-b)(a-c)} f_a + \frac{(x-a)(x-c)}{(b-a)(b-c)} f_b + \frac{(x-a)(x-b)}{(c-a)(c-b)} f_c$$

3. find the unique stationary point of the parabola,

$$d = \frac{1}{2} \frac{a^2(f_c - f_b) + b^2(f_a - f_c) + c^2(f_b - f_a)}{a(f_c - f_b) + b(f_a - f_c) + c(f_b - f_a)}$$

4. $(a, b, c) \leftarrow (b, c, d)$
5. continue until $|c - b| < \delta$

# Successive parabolic interpolation



1. choose three points $(a, b, c)$ that need not bracket a stationary point;
2. find the unique parabola to the curve passing through those points,

$$p(x) = \frac{(x-b)(x-c)}{(a-b)(a-c)} f_a + \frac{(x-a)(x-c)}{(b-a)(b-c)} f_b + \frac{(x-a)(x-b)}{(c-a)(c-b)} f_c$$

3. find the unique stationary point of the parabola,

$$d = \frac{1}{2} \frac{a^2 (f_c - f_b) + b^2 (f_a - f_c) + c^2 (f_b - f_a)}{a (f_c - f_b) + b (f_a - f_c) + c (f_b - f_a)}$$

4. $(a, b, c) \leftarrow (b, c, d)$
5. continue until $|c - b| < \delta$
6. the order of convergence $q \approx 1.324$

# Successive parabolic interpolation



What if we only want to converge to minima?

# Successive parabolic interpolation



What if we only want to converge to minima?

1. find a bracket $(a, c, b)$;

# Successive parabolic interpolation

What if we only want to converge to minima?

1. find a bracket $(a, c, b)$;
2. find the new point $d$;

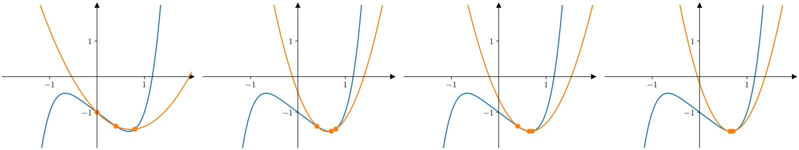
# Successive parabolic interpolation

What if we only want to converge to minima?

1. find a bracket $(a, c, b)$;
2. find the new point $d$;
3. update with the new, narrower bracket,

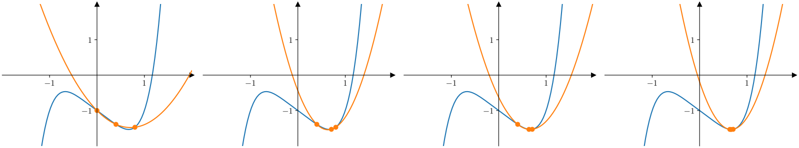
$$(a, c, b) \leftarrow \begin{cases} (a, d, c) & d < c, \quad f(d) < f(c) \\ (d, c, b) & d < c, \quad f(d) > f(c) \\ (c, d, b) & d > c, \quad f(d) < f(c) \\ (a, c, d) & d > c, \quad f(d) > f(c) \,. \end{cases}$$

# Successive parabolic interpolation



- Successive parabolic interpolation applied to the quintic with $(a, b, c) = (0.0, 0.4, 0.8)$.

# Successive parabolic interpolation




- Successive parabolic interpolation applied to the quintic with $(a, b, c) = (0.0, 0.4, 0.8)$.
- What happens if you set $(a, b, c) = (0.0, 0.5, 1.0)$?

# Newton–Raphson (again)

1. pick an initial guess $x_0$ close to the stationary point;

# Newton–Raphson (again)



1. pick an initial guess $x_0$ close to the stationary point;
2. find the extremum of the parabola that at $x_0$ has the same function value, derivative, and curvature as the objective function

# Newton–Raphson (again)



1. pick an initial guess $x_0$ close to the stationary point;
2. find the extremum of the parabola that at $x_0$ has the same function value, derivative, and curvature as the objective function
3. *i.e.*, solve for the extremum of a second-order Taylor expansion of the function,

$$f\left(\xi\right) = f\left(x_0 + \epsilon_0\right) = f_0 + \epsilon_0 f_0' + \mathcal{O}\left(\epsilon_0^2\right)$$

# Newton–Raphson (again)



1. pick an initial guess $x_0$ close to the stationary point;
2. find the extremum of the parabola that at $x_0$ has the same function value, derivative, and curvature as the objective function
3. *i.e.*, solve for the extremum of a second-order Taylor expansion of the function,

$$f'(\xi) = f'(x_0 + \epsilon_0) = f_0' + \epsilon_0 f_0'' + \mathcal{O}\left(\epsilon_0^2\right)$$

# Newton–Raphson (again)

1. pick an initial guess $x_0$ close to the stationary point;
2. find the extremum of the parabola that at $x_0$ has the same function value, derivative, and curvature as the objective function
3. *i.e.*, solve for the extremum of a second-order Taylor expansion of the function,

$$0 = f_0' + \epsilon_0 f_0'' + \mathcal{O}\left(\epsilon_k^2\right)$$

# Newton–Raphson (again)

1. pick an initial guess $x_0$ close to the stationary point;
2. find the extremum of the parabola that at $x_0$ has the same function value, derivative, and curvature as the objective function
3. *i.e.*, solve for the extremum of a second-order Taylor expansion of the function,

$$\epsilon_0 = -\frac{f_0'}{f_0''} + \mathcal{O}\left(\epsilon_k^2\right)$$

# Newton–Raphson (again)

1. pick an initial guess $x_0$ close to the stationary point;
2. find the extremum of the parabola that at $x_0$ has the same function value, derivative, and curvature as the objective function
3. *i.e.*, solve for the extremum of a second-order Taylor expansion of the function,

$$\epsilon_0 = -\frac{f_0'}{f_0''} + \mathcal{O}\left(\epsilon_k^2\right)$$

4. which point is,

$$x_1 = x_0 - \frac{f_0'}{f_0''}$$

# Newton–Raphson (again)



1. pick an initial guess $x_0$ close to the stationary point;
2. find the extremum of the parabola that at $x_0$ has the same function value, derivative, and curvature as the objective function
3. *i.e.*, solve for the extremum of a second-order Taylor expansion of the function,

$$\epsilon_0 = -\frac{f_0'}{f_0''} + \mathcal{O}\left(\epsilon_k^2\right)$$

4. which point is,

$$x_1 = x_0 - \frac{f_0'}{f_0''}$$

5. $x_0 \leftarrow x_1$

# Newton–Raphson (again)

1. pick an initial guess $x_0$ close to the stationary point;
2. find the extremum of the parabola that at $x_0$ has the same function value, derivative, and curvature as the objective function
3. *i.e.*, solve for the extremum of a second-order Taylor expansion of the function,

$$\epsilon_0 = -\frac{f_0'}{f_0''} + \mathcal{O}\left(\epsilon_k^2\right)$$

4. which point is,

$$x_1 = x_0 - \frac{f_0'}{f_0''}$$

5. $x_0 \leftarrow x_1$
6. continue until $|x_1 - x_0| < \delta$

# Newton–Raphson (again)



- to find only minima, take the absolute value of the second derivative in the update step,

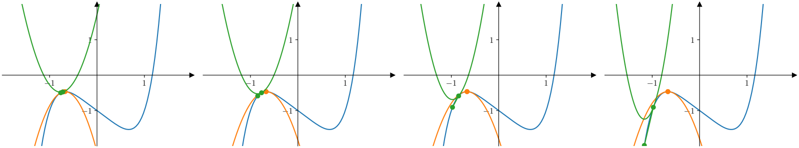
$$x_1 = x_0 - \frac{f_0'}{|f_0''|}$$

# Newton–Raphson (again)



- to find only minima, take the absolute value of the second derivative in the update step,

$$x_1 = x_0 - \frac{f_0'}{|f_0''|}$$




- Convergence to a minimum is not guaranteed...

# Secant Method (again)



- You may have noticed that the secant method for zeros is equivalent to the NR method for zeros with the derivative approximated by first backward differences

# Secant Method (again)



- You may have noticed that the secant method for zeros is equivalent to the NR method for zeros with the derivative approximated by first backward differences
- the secant method for stationary points works on a similar basis, with the second derivative term replaced by a first backward differences approximation of the first derivative...of the first derivative

# Gradient Descent (something new!)



1. pick an initial guess $x_0$ close to the stationary point;

# Gradient Descent (something new!)

1. pick an initial guess $x_0$ close to the stationary point;
2. move a small amount in the direction that the function is decreasing,

$$x_1 = x_0 + \Delta x_0 = x_0 - \alpha \boldsymbol{p}$$

($\alpha$ is a small, positive, adjustable parameter, chosen to guarantee function decrease)

# Gradient Descent (something new!)



1. pick an initial guess $x_0$ close to the stationary point;
2. move a small amount in the direction that the function is decreasing,

$$x_1 = x_0 + \Delta x_0 = x_0 - \alpha \boldsymbol{p}$$

   ($\alpha$ is a small, positive, adjustable parameter, chosen to guarantee function decrease)
3. $x_0 \leftarrow x_1$

# Gradient Descent (something new!)



1. pick an initial guess $x_0$ close to the stationary point;
2. move a small amount in the direction that the function is decreasing,

$$x_1 = x_0 + \Delta x_0 = x_0 - \alpha \boldsymbol{p}$$

   ($\alpha$ is a small, positive, adjustable parameter, chosen to guarantee function decrease)
3. $x_0 \leftarrow x_1$
4. continue until $|x_1 - x_0| < \delta$

# Gradient Descent (something new!)

1. pick an initial guess $x_0$ close to the stationary point;
2. move a small amount in the direction that the function is decreasing,

$$x_1 = x_0 + \Delta x_0 = x_0 - \alpha \boldsymbol{p}$$

   ($\alpha$ is a small, positive, adjustable parameter, chosen to guarantee function decrease)
3. $x_0 \leftarrow x_1$
4. continue until $|x_1 - x_0| < \delta$

- By construction, this method only finds minima

# Gradient Descent (something new!)



1. pick an initial guess $x_0$ close to the stationary point;
2. move a small amount in the direction that the function is decreasing,

$$x_1 = x_0 + \Delta x_0 = x_0 - \alpha \boldsymbol{p}$$

   ($\alpha$ is a small, positive, adjustable parameter, chosen to guarantee function decrease)
3. $x_0 \leftarrow x_1$
4. continue until $|x_1 - x_0| < \delta$

- By construction, this method only finds minima
- to find maxima, a negative value of $\alpha$ should be employed

# Homework

## The gradient-descent (one dimensional)

Choosing the value of $\alpha$ is a critical consideration: if $\alpha$ is too small, convergence is slow; if $\alpha$ is too large, then we risk jumping back and forth over the minimum, which also results in slow convergence.

1. devise a simple algorithm to choose $\alpha$;
2. implement the gradient-descent method, using this algorithm.

# Backtracking line search



- consider the update

$$x_1 = x_0 - \alpha p$$

where $\alpha$ is a positive, adjustable parameter, and $p = f'(x) / |f'(x)|$ is either 1 or $-1$

# Backtracking line search

- consider the update

$$x_1 = x_0 - \alpha p$$

where $\alpha$ is a positive, adjustable parameter, and $p = f'(x) / |f'(x)|$ is either 1 or $-1$

- start with a linear approximation to $f$,

$$f(x_0 - \alpha p) = f(x_0) - \alpha p f'(x_0) + \mathcal{O}(\alpha^2)$$

# Backtracking line search



- consider the update

$$x_1 = x_0 - \alpha p$$

where $\alpha$ is a positive, adjustable parameter, and $p = f'(x) / |f'(x)|$ is either 1 or $-1$

- start with a linear approximation to $f$,

$$f(x_0 - \alpha p) = f(x_0) - \alpha p f'(x_0) + \mathcal{O}(\alpha^2)$$

- if $f$ really was linear, then

$$f(x_0 - \alpha p) - f(x_0) = -\alpha p f'(x_0)$$

# Backtracking line search

- consider the update

$$x_1 = x_0 - \alpha p$$

where $\alpha$ is a positive, adjustable parameter, and $p = f'(x) / |f'(x)|$ is either 1 or $-1$

- start with a linear approximation to $f$,

$$f(x_0 - \alpha p) = f(x_0) - \alpha p f'(x_0) + \mathcal{O}(\alpha^2)$$

- if $f$ really was linear, then

$$f(x_0 - \alpha p) - f(x_0) = -\alpha p f'(x_0)$$

- so we say the function value has decreased sufficiently if

$$f(x_0 - \alpha p) \leq f(x_0) - \alpha c m \tag{1}$$

where $m = p f'(x_0) = |f'(x_0)|$ and $c \in (0, 1)$ is a control parameter

# Backtracking line search

- consider the update
$$x_1 = x_0 - \alpha p$$
where $\alpha$ is a positive, adjustable parameter, and $p = f'(x) / |f'(x)|$ is either 1 or $-1$
- start with a linear approximation to $f$,
$$f(x_0 - \alpha p) = f(x_0) - \alpha p f'(x_0) + \mathcal{O}(\alpha^2)$$
- if $f$ really was linear, then
$$f(x_0 - \alpha p) - f(x_0) = -\alpha p f'(x_0)$$
- so we say the function value has decreased sufficiently if
$$f(x_0 - \alpha p) \leq f(x_0) - \alpha c m \tag{1}$$
where $m = pf'(x_0) = |f'(x_0)|$ and $c \in (0, 1)$ is a control parameter
- if condition 1 is not satisfied, update $\alpha$ as $\alpha \leftarrow \tau\alpha$ where $\tau \in (0, 1)$ is another control parameter

# Backtracking line search

- consider the update

$$x_1 = x_0 - \alpha p$$

where $\alpha$ is a positive, adjustable parameter, and $p = f'(x) / |f'(x)|$ is either 1 or $-1$

- start with a linear approximation to $f$,

$$f(x_0 - \alpha p) = f(x_0) - \alpha p f'(x_0) + \mathcal{O}(\alpha^2)$$

- if $f$ really was linear, then

$$f(x_0 - \alpha p) - f(x_0) = -\alpha p f'(x_0)$$

- so we say the function value has decreased sufficiently if

$$f(x_0 - \alpha p) \leq f(x_0) - \alpha c m \tag{1}$$

where $m = pf'(x_0) = |f'(x_0)|$ and $c \in (0, 1)$ is a control parameter

- if condition 1 is not satisfied, update $\alpha$ as $\alpha \leftarrow \tau\alpha$ where $\tau \in (0, 1)$ is another control parameter
- continue until condition 1 is satisfied

# Backtracking line search

- consider the update

$$x_1 = x_0 - \alpha p$$

where $\alpha$ is a positive, adjustable parameter, and $p = f'(x) / |f'(x)|$ is either 1 or $-1$

- start with a linear approximation to $f$,

$$f(x_0 - \alpha p) = f(x_0) - \alpha p f'(x_0) + \mathcal{O}(\alpha^2)$$

- if $f$ really was linear, then

$$f(x_0 - \alpha p) - f(x_0) = -\alpha p f'(x_0)$$

- so we say the function value has decreased sufficiently if

$$f(x_0 - \alpha p) \leq f(x_0) - \alpha c m \tag{1}$$

where $m = pf'(x_0) = |f'(x_0)|$ and $c \in (0, 1)$ is a control parameter

- if condition 1 is not satisfied, update $\alpha$ as $\alpha \leftarrow \tau\alpha$ where $\tau \in (0, 1)$ is another control parameter
- continue until condition 1 is satisfied
- typically, $c = 0.5$, $\tau = 0.5$

# Overview

# Solving linear equations

- Consider the system of equations,

$$\begin{aligned} 2x + y - z &= 8; \\ -3x - y + 2z &= -11; \\ -2x + y + 2z &= -3, \end{aligned}$$

- we can summarise these equations in an augmented matrix

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array}\right]$$

# Solving linear equations



- Consider the system of equations,

$$
\begin{aligned}
2x + y - z &= 8; \\
-3x - y + 2z &= -11; \\
-2x + y + 2z &= -3,
\end{aligned}
$$

- $(R2) - \frac{3}{2}(R1)$, $(R3) + (R1)$

$$
\left[\begin{array}{ccc|c}
2 & 1 & -1 & 8 \\
0 & 0.5 & 0.5 & 1 \\
0 & 2 & 1 & 5
\end{array}\right]
$$

# Solving linear equations



- Consider the system of equations,

$$\begin{aligned} 2x + y - z &= 8; \\ -3x - y + 2z &= -11; \\ -2x + y + 2z &= -3, \end{aligned}$$

- $(R3) - 4(R2)$

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & 0.5 & 0.5 & 1 \\ 0 & 0 & -1 & 1 \end{array}\right]$$

- at this point, we could solve by back-substitution
- we could also find the determinant of the original matrix as the product of the diagona elements
- additionally, if the matrix is symmetric, the signs of the eigenvalues of the matrix are the same as the signs of the diagonal elements

# Solving linear equations

- Consider the system of equations,

$$\begin{aligned} 2x + y - z &= 8; \\ -3x - y + 2z &= -11; \\ -2x + y + 2z &= -3, \end{aligned}$$

- reflect in the antidiagonal

$$\left[\begin{array}{ccc|c} -1 & 0 & 0 & 1 \\ 0.5 & 0.5 & 0 & 1 \\ -1 & 1 & 2 & 8 \end{array}\right]$$

# Solving linear equations

- Consider the system of equations,

$$\begin{aligned} 2x + y - z &= 8; \\ -3x - y + 2z &= -11; \\ -2x + y + 2z &= -3, \end{aligned}$$

- $(R2) + \frac{1}{2}(R1)$, $(R3) - (R1)$

$$\left[\begin{array}{ccc|c} -1 & 0 & 0 & 1 \\ 0 & 0.5 & 0 & 1.5 \\ 0 & 1 & 2 & 7 \end{array}\right]$$

# Solving linear equations

- Consider the system of equations,

$$
\begin{aligned}
2x + y - z &= 8; \\
-3x - y + 2z &= -11; \\
-2x + y + 2z &= -3,
\end{aligned}
$$

- $(R3) - 2(R2)$

$$
\left[\begin{array}{ccc|c}
-1 & 0 & 0 & 1 \\
0 & 0.5 & 0 & 1.5 \\
0 & 0 & 2 & 4
\end{array}\right]
$$

# Solving linear equations



- Consider the system of equations,

$$\begin{aligned} 2x + y - z &= 8; \\ -3x - y + 2z &= -11; \\ -2x + y + 2z &= -3, \end{aligned}$$

- divide through by the diagonal

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

REDUCED ROW ECHELON FORM

# Solving linear equations



- Consider the system of equations,

$$\begin{aligned} 2x + y - z &= 8; \\ -3x - y + 2z &= -11; \\ -2x + y + 2z &= -3, \end{aligned}$$

- don't forget that we switched $x$ and $z$

$$\left[\begin{array}{ccc|r} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{array}\right]$$

# Solving linear equations

- Consider the system of equations,

$$
\begin{aligned}
2x + y - z &= 8; \\
-3x - y + 2z &= -11; \\
-2x + y + 2z &= -3,
\end{aligned}
$$

- don't forget that we switched $x$ and $z$

$$
\left[\begin{array}{ccc|c}
1 & 0 & 0 & 2 \\
0 & 1 & 0 & 3 \\
0 & 0 & 1 & -1
\end{array}\right]
$$

- the proof:

$$
\begin{aligned}
2(2) + (3) - (-1) &= 8 \\
-3(2) - (3) + 2(-1) &= -11 \\
-2(2) + (3) + 2(-1) &= -3
\end{aligned}
$$

# Solving linear equations

- solving for the inverse of a matrix amounts to applying the above procedure to

$$\left[\mathbf{A} \mid \mathbf{I}\right]$$

# Solving linear equations

- solving for the inverse of a matrix amounts to applying the above procedure to

$$\left[\mathbf{A} \mid \mathbf{I}\right]$$

- for example, our matrix,

$$\left[\begin{array}{rrr|rrr} 2 & 1 & -1 & 1 & 0 & 0 \\ -3 & -1 & 2 & 0 & 1 & 0 \\ -2 & 1 & 2 & 0 & 0 & 1 \end{array}\right],$$

- solving for the inverse of a matrix amounts to applying the above procedure to

$$\left[\mathbf{A} \mid \mathbf{I}\right]$$

- for example, our matrix,

$$\left[\begin{array}{ccc|ccc} 2 & 1 & -1 & 1 & 0 & 0 \\ -3 & -1 & 2 & 0 & 1 & 0 \\ -2 & 1 & 2 & 0 & 0 & 1 \end{array}\right],$$

- reduces to,

$$\left[\begin{array}{ccc|ccc} -1 & 0 & 0 & -5 & -4 & 1 \\ 0 & \frac{1}{2} & 0 & -1 & -1 & \frac{1}{2} \\ 0 & 0 & 2 & 8 & 6 & -2 \end{array}\right],$$

- solving for the inverse of a matrix amounts to applying the above procedure to

$$\left[\mathbf{A} \mid \mathbf{I}\right]$$

- for example, our matrix,

$$\left[\begin{array}{rrr|rrr} 2 & 1 & -1 & 1 & 0 & 0 \\ -3 & -1 & 2 & 0 & 1 & 0 \\ -2 & 1 & 2 & 0 & 0 & 1 \end{array}\right],$$

- reduces to,

$$\left[\begin{array}{rrr|rrr} -1 & 0 & 0 & -5 & -4 & 1 \\ 0 & \frac{1}{2} & 0 & -1 & -1 & \frac{1}{2} \\ 0 & 0 & 2 & 8 & 6 & -2 \end{array}\right],$$

- and finally solves to,

$$\left[\begin{array}{rrr|rrr} 1 & 0 & 0 & 4 & 3 & -1 \\ 0 & 1 & 0 & -2 & -2 & 1 \\ 0 & 0 & 1 & 5 & 4 & -1 \end{array}\right].$$

# Non-linear equations—Newton's method—again (2)



- Newton's method in many variables is derived in a similar way to single variable case

# Non-linear equations—Newton's method—again (2)

- Newton's method in many variables is derived in a similar way to single variable case
- We start with the many variable Taylor expansion of a function at the stationary point, $\xi$,

$$f\left(\xi\right) = f\left(\mathbf{x}_k + \boldsymbol{\epsilon}_k\right) = f\left(\mathbf{x}_k\right) + \nabla f\left(\mathbf{x}_k\right)^T \cdot \boldsymbol{\epsilon}_k + \mathcal{O}\left(\left|\boldsymbol{\epsilon}_k^2\right|\right)$$

# Non-linear equations—Newton's method—again (2)



- Newton's method in many variables is derived in a similar way to single variable case
- We start with the many variable Taylor expansion of a function at the stationary point, $\xi$,

$$f(\xi) = f(\mathbf{x}_k + \boldsymbol{\epsilon}_k) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \cdot \boldsymbol{\epsilon}_k + \mathcal{O}\left(\left|\boldsymbol{\epsilon}_k^2\right|\right)$$

- Taking derivatives, and setting the derivative at the stationary point to zero,

$$\nabla f(\xi) = \mathbf{0} = \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k) \cdot \boldsymbol{\epsilon}_k + \mathcal{O}\left(\left|\boldsymbol{\epsilon}_k^2\right|\right)$$

# Non-linear equations—Newton's method—again (2)

- Newton's method in many variables is derived in a similar way to single variable case
- We start with the many variable Taylor expansion of a function at the stationary point, $\xi$,

$$f\left(\xi\right) = f\left(\mathbf{x}_k + \boldsymbol{\epsilon}_k\right) = f\left(\mathbf{x}_k\right) + \nabla f\left(\mathbf{x}_k\right)^T \cdot \boldsymbol{\epsilon}_k + \mathcal{O}\left(\left|\boldsymbol{\epsilon}_k^2\right|\right)$$

- Taking derivatives, and setting the derivative at the stationary point to zero,

$$\nabla f\left(\xi\right) = \mathbf{0} = \nabla f\left(\mathbf{x}_k\right) + \nabla^2 f\left(\mathbf{x}_k\right) \cdot \boldsymbol{\epsilon}_k + \mathcal{O}\left(\left|\boldsymbol{\epsilon}_k^2\right|\right)$$

- rearranging,

$$\cdot\nabla^2 f\left(\mathbf{x}_k\right) \cdot \boldsymbol{\epsilon}_k = -\nabla f\left(\mathbf{x}_k\right) + \mathcal{O}\left(\left|\boldsymbol{\epsilon}_k^2\right|\right)$$

# Non-linear equations—Newton's method—again (2)



- truncating at second order in the error, we have a matrix equation of the form,

$$\mathbf{Ax} = \mathbf{b},$$

# Non-linear equations—Newton's method—again (2)



- truncating at second order in the error, we have a matrix equation of the form,

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

- where,

$$\begin{aligned} \mathbf{x} &= \boldsymbol{\epsilon}_k; \\ \mathbf{b} &= -\nabla f\left(\mathbf{x}_k\right); \\ \mathbf{A} &= \nabla^2 f\left(\mathbf{x}_k\right). \end{aligned}$$

- truncating at second order in the error, we have a matrix equation of the form,

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

- where,

$$\begin{aligned} \mathbf{x} &= \boldsymbol{\epsilon}_k; \\ \mathbf{b} &= -\nabla f(\mathbf{x}_k); \\ \mathbf{A} &= \nabla^2 f(\mathbf{x}_k). \end{aligned}$$

- the solution for $\boldsymbol{\epsilon}_k$ is,

$$\boldsymbol{\epsilon}_k = -\nabla^2 f(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k) + \mathcal{O}\left(\left|\boldsymbol{\epsilon}_k^2\right|\right)$$

# Non-linear equations—Newton's method—again (2)

- truncating at second order in the error, we have a matrix equation of the form,

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

- where,

$$\begin{aligned} \mathbf{x} &= \boldsymbol{\epsilon}_k; \\ \mathbf{b} &= -\nabla f(\mathbf{x}_k); \\ \mathbf{A} &= \nabla^2 f(\mathbf{x}_k). \end{aligned}$$

- the solution for $\boldsymbol{\epsilon}_k$ is,

$$\boldsymbol{\epsilon}_k = -\nabla^2 f(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k) + \mathcal{O}\left(\left|\boldsymbol{\epsilon}_k^2\right|\right)$$

- giving the many-variable Newton update,

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x} - \nabla^2 f(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k)} \tag{2}$$

MANY-VARIABLE NEWTON–RAPHSON UPDATE

# Non-linear equations—Newton's method—again (2)



- truncating at second order in the error, we have a matrix equation of the form,

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

- where,

$$\begin{aligned} \mathbf{x} &= \epsilon_k; \\ \mathbf{b} &= -\nabla f(\mathbf{x}_k); \\ \mathbf{A} &= \nabla^2 f(\mathbf{x}_k). \end{aligned}$$

- the solution for $\epsilon_k$ is,

$$\epsilon_k = -\nabla^2 f(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k) + \mathcal{O}\left(\left|\epsilon_k^2\right|\right)$$

- giving the many-variable Newton update,

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x} - \nabla^2 f(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k)} \tag{2}$$

MANY-VARIABLE NEWTON–RAPHSON UPDATE

- we can compute the inverse of the matrix of second derivatives, the Hessian matrix, using the matrix methods outlined above.

# Non-linear equations—Newton's method—again (2)

- in the one dimensional version, we takes as the new point the stationary point of a parabola

# Non-linear equations—Newton's method—again (2)



- in the one dimensional version, we takes as the new point the stationary point of a parabola
- a minimum if the parabola has positive curvature, and a maximum if it has negative curvature

# Non-linear equations—Newton's method—again (2)

- in the one dimensional version, we takes as the new point the stationary point of a parabola
- a minimum if the parabola has positive curvature, and a maximum if it has negative curvature
- in many dimensions we find the stationary point of a quadratic form,

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c$$

  a linear combination of orthogonal parabolas

# Non-linear equations—Newton's method—again (2)



- in the one dimensional version, we takes as the new point the stationary point of a parabola
- a minimum if the parabola has positive curvature, and a maximum if it has negative curvature
- in many dimensions we find the stationary point of a quadratic form,

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c$$

  a linear combination of orthogonal parabolas
- the search direction is only one in which the function decreases if all the parabolas have positive curvature,

# Non-linear equations—Newton's method—again (2)



- in the one dimensional version, we takes as the new point the stationary point of a parabola
- a minimum if the parabola has positive curvature, and a maximum if it has negative curvature
- in many dimensions we find the stationary point of a quadratic form,

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c$$

  a linear combination of orthogonal parabolas
- the search direction is only one in which the function decreases if all the parabolas have positive curvature,
- *i.e.*, if the Hessian matrix is positive definite, *i.e.*,

$$\forall \mathbf{x} \neq \mathbf{0}, \quad \mathbf{x}^T\mathbf{A}\mathbf{x} > 0$$

# Non-linear equations—Newton's method—again (2)



- in the one dimensional version, we takes as the new point the stationary point of a parabola
- a minimum if the parabola has positive curvature, and a maximum if it has negative curvature
- in many dimensions we find the stationary point of a quadratic form,

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c$$

  a linear combination of orthogonal parabolas
- the search direction is only one in which the function decreases if all the parabolas have positive curvature,
- *i.e.*, if the Hessian matrix is positive definite, *i.e.*,

$$\forall \mathbf{x} \neq \mathbf{0}, \quad \mathbf{x}^T\mathbf{A}\mathbf{x} > 0$$

- this condition is true iff all of the eigenvalues of the Hessian matrix are positive

- we can shift the eigenvalues of a symmetric matrix to positive values, while leaving the eigenvectors unchanged, by adding to it a multiple of the identity matrix,

$$\mathbf{B} = \mathbf{A} + \mu \mathbf{I}$$

where $\mu + \lambda_{min} > 0$, and $\lambda_{min}$ is the smallest (most negative) eigenvalue of $\mathbf{A}$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- computing the Hessian and its inverse is expensive for large systems—we would like to compute it infrequently or never...

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- computing the Hessian and its inverse is expensive for large systems—we would like to compute it infrequently or never...
- one solution is to update the coordinates with an approximate Hessian matrix, $\mathbf{B}_k$,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k^{-1} \cdot \nabla f(\mathbf{x}_k)$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- computing the Hessian and its inverse is expensive for large systems—we would like to compute it infrequently or never...
- one solution is to update the coordinates with an approximate Hessian matrix, $\mathbf{B}_k$,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k^{-1} \cdot \nabla f(\mathbf{x}_k)$$

- and then update the the approximate Hessian,

$$\mathbf{B}_{k+1} = ??$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- computing the Hessian and its inverse is expensive for large systems—we would like to compute it infrequently or never...
- one solution is to update the coordinates with an approximate Hessian matrix, $\mathbf{B}_k$,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k^{-1} \cdot \nabla f(\mathbf{x}_k)$$

- and then update the the approximate Hessian,

$$\mathbf{B}_{k+1} = ??$$

- how do we choose the update?

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- the one-dimensional secant equation (from first backward differences),

$$f''(x) \cdot h = f'(x) - f'(x-h) + \mathcal{O}\left(h^2\right)$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- the one-dimensional secant equation (from first backward differences),

$$f''(x) \cdot h = f'(x) - f'(x-h) + \mathcal{O}\left(h^2\right)$$

- the many-variable secant equation is analogously

$$\nabla^2 f(\mathbf{x}) \cdot \mathbf{h} = \nabla f(\mathbf{x}) - \nabla f(\mathbf{x} - \mathbf{h}) + \mathcal{O}\left(\mathbf{h}^2\right)$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- the one-dimensional secant equation (from first backward differences),

$$f''(x) \cdot h = f'(x) - f'(x-h) + \mathcal{O}\left(h^2\right)$$

- the many-variable secant equation is analogously

$$\nabla^2 f(\mathbf{x}) \cdot \mathbf{h} = \nabla f(\mathbf{x}) - \nabla f(\mathbf{x} - \mathbf{h}) + \mathcal{O}\left(\mathbf{h}^2\right)$$

- re-writing in the above terms, we get

$$\nabla^2 f(\mathbf{x}_{k+1}) \cdot (\mathbf{x}_{k+1} - \mathbf{x}_k) \approx \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$$

- choose an update that is consistent with the secant equation!

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- the one-dimensional secant equation (from first backward differences),

$$f''(x) \cdot h = f'(x) - f'(x - h) + \mathcal{O}\left(h^2\right)$$

- the many-variable secant equation is analogously

$$\nabla^2 f(\mathbf{x}) \cdot \mathbf{h} = \nabla f(\mathbf{x}) - \nabla f(\mathbf{x} - \mathbf{h}) + \mathcal{O}\left(\mathbf{h}^2\right)$$

- re-writing in the above terms, we get

$$\nabla^2 f(\mathbf{x}_{k+1}) \cdot (\mathbf{x}_{k+1} - \mathbf{x}_k) \approx \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$$

- choose an update that is consistent with the secant equation!
- but... there is no unique solution to the secant equation!

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- if we want to do minimisation, we require that $\mathbf{B}_k$ be symmetric and positive definite
- similarly, the updated matrix $\mathbf{B}_{k+1}$ must also be symmetric and positive definite

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- if we want to do minimisation, we require that $\mathbf{B}_k$ be symmetric and positive definite
- similarly, the updated matrix $\mathbf{B}_{k+1}$ must also be symmetric and positive definite
- does solving the secant equation directly ensure this?

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- if we want to do minimisation, we require that $\mathbf{B}_k$ be symmetric and positive definite
- similarly, the updated matrix $\mathbf{B}_{k+1}$ must also be symmetric and positive definite
- does solving the secant equation directly ensure this?
- taking $\mathbf{B}_{k+1} \approx \nabla^2 f\left(\mathbf{x}_{k+1}\right)$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \nabla f\left(\mathbf{x}_{k+1}\right) - \nabla f\left(\mathbf{x}_k\right)$, the secant equation is expressed as,

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- if we want to do minimisation, we require that $\mathbf{B}_k$ be symmetric and positive definite
- similarly, the updated matrix $\mathbf{B}_{k+1}$ must also be symmetric and positive definite
- does solving the secant equation directly ensure this?
- taking $\mathbf{B}_{k+1} \approx \nabla^2 f(\mathbf{x}_{k+1})$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$, the secant equation is expressed as,

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$$

- rearranging,

$$\mathbf{B}_{k+1} = \left(\mathbf{y}_k \cdot \mathbf{s}_k^T\right) \cdot \left(\mathbf{s}_k \cdot \mathbf{s}_k^T\right)^{-1}$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- if we want to do minimisation, we require that $\mathbf{B}_k$ be symmetric and positive definite
- similarly, the updated matrix $\mathbf{B}_{k+1}$ must also be symmetric and positive definite
- does solving the secant equation directly ensure this?
- taking $\mathbf{B}_{k+1} \approx \nabla^2 f(\mathbf{x}_{k+1})$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$, the secant equation is expressed as,

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$$

- rearranging,

$$\mathbf{B}_{k+1} = \left(\mathbf{y}_k \cdot \mathbf{s}_k^T\right) \cdot \left(\mathbf{s}_k \cdot \mathbf{s}_k^T\right)^{-1}$$

- this matrix is in general non-symmetric

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- one solution is Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- one solution is Broyden–Fletcher–Goldfarb–Shanno (BFGS) method
- assume the update is of the form,

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \alpha \mathbf{u}\mathbf{u}^{\mathbf{T}} + \beta \mathbf{v}\mathbf{v}^{\mathbf{T}}$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- one solution is Broyden–Fletcher–Goldfarb–Shanno (BFGS) method
- assume the update is of the form,

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \alpha \mathbf{u}\mathbf{u}^{\mathbf{T}} + \beta \mathbf{v}\mathbf{v}^{\mathbf{T}}$$

- the added term is a symmetric, positive semi-definite rank-two matrix

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- one solution is Broyden–Fletcher–Goldfarb–Shanno (BFGS) method
- assume the update is of the form,

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \alpha \mathbf{u}\mathbf{u}^{\mathbf{T}} + \beta \mathbf{v}\mathbf{v}^{\mathbf{T}}$$

- the added term is a symmetric, positive semi-definite rank-two matrix
- we require,

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- one solution is Broyden–Fletcher–Goldfarb–Shanno (BFGS) method
- assume the update is of the form,

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \alpha \mathbf{u}\mathbf{u}^{\mathbf{T}} + \beta \mathbf{v}\mathbf{v}^{\mathbf{T}}$$

- the added term is a symmetric, positive semi-definite rank-two matrix
- we require,

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$$

- choosing $\mathbf{u} = \mathbf{y}_k$ and $\mathbf{v} = \mathbf{B}_k\mathbf{s}_k$, and solving for $\alpha, \beta$, we get

$$\alpha = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}; \quad \beta = \frac{1}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k},$$

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- one solution is Broyden–Fletcher–Goldfarb–Shanno (BFGS) method
- assume the update is of the form,

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \alpha \mathbf{u}\mathbf{u}^{\mathbf{T}} + \beta \mathbf{v}\mathbf{v}^{\mathbf{T}}$$

- the added term is a symmetric, positive semi-definite rank-two matrix
- we require,

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$$

- choosing $\mathbf{u} = \mathbf{y}_k$ and $\mathbf{v} = \mathbf{B}_k\mathbf{s}_k$, and solving for $\alpha, \beta$, we get

$$\alpha = \frac{1}{\mathbf{y}_k^T\mathbf{s}_k}; \quad \beta = \frac{1}{\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k},$$

- so the update to $\mathbf{B}$ is,

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k} - \frac{\mathbf{B}_k\mathbf{s}_k\mathbf{s}_k^T\mathbf{B}_k^T}{\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k}$$

(verify this for yourselves)

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- collecting these ideas together, recalling that $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, and $\mathbf{y}_k = \nabla f\left(\mathbf{x}_{k+1}\right) - \nabla f\left(\mathbf{x}_k\right)$,

$$\boxed{\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \mathbf{B}_k^{-1} \cdot \nabla f\left(\mathbf{x}_k\right) \qquad (3) \\ \mathbf{B}_{k+1} &= \mathbf{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k^T}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k}. \qquad (4) \end{aligned}}$$

BFGS UPDATES

- however, since we only need the inverse of **B**, we can rewrite the update using the Sherman–Morrison formula,

$$\mathbf{B}_{k+1}^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) \mathbf{B}_k^{-1} \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \tag{5}$$

- however, since we only need the inverse of **B**, we can rewrite the update using the Sherman–Morrison formula,

$$\mathbf{B}_{k+1}^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) \mathbf{B}_k^{-1} \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \quad (5)$$

- as $\mathbf{B}_k^{-1}$ is symmetric, and both $\mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k$ and $\mathbf{s}_k^T \mathbf{y}_k$ are scalars, we can write

$$\boxed{\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \mathbf{B}_k^{-1} \cdot \nabla f(\mathbf{x}_k) \\ \mathbf{B}_{k+1}^{-1} &= \mathbf{B}_k^{-1} + \frac{(\mathbf{s}_k^T \mathbf{y}_k + \mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k)(\mathbf{s}_k \mathbf{s}_k^T)}{(\mathbf{s}_k^T \mathbf{y}_k)^2} - \frac{\mathbf{B}_k^{-1} \mathbf{y}_k \mathbf{s}_k^T + \mathbf{s}_k \mathbf{y}_k^T \mathbf{B}_k^{-1}}{\mathbf{s}_k^T \mathbf{y}_k} \end{aligned}} \quad \begin{aligned} (6) \\ (7) \end{aligned}$$

BFGS UPDATES (IMPROVED)

and we never need to invert a matrix ever again.

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

- however, since we only need the inverse of **B**, we can rewrite the update using the Sherman–Morrison formula,

$$\mathbf{B}_{k+1}^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) \mathbf{B}_k^{-1} \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \tag{5}$$

- as $\mathbf{B}_k^{-1}$ is symmetric, and both $\mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k$ and $\mathbf{s}_k^T \mathbf{y}_k$ are scalars, we can write

$$\boxed{\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \mathbf{B}_k^{-1} \cdot \nabla f(\mathbf{x}_k) & (6)\\ \mathbf{B}_{k+1}^{-1} &= \mathbf{B}_k^{-1} + \frac{(\mathbf{s}_k^T \mathbf{y}_k + \mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k)(\mathbf{s}_k \mathbf{s}_k^T)}{(\mathbf{s}_k^T \mathbf{y}_k)^2} - \frac{\mathbf{B}_k^{-1} \mathbf{y}_k \mathbf{s}_k^T + \mathbf{s}_k \mathbf{y}_k^T \mathbf{B}_k^{-1}}{\mathbf{s}_k^T \mathbf{y}_k} & (7) \end{aligned}}$$

BFGS UPDATES (IMPROVED)

and we never need to invert a matrix ever again.

- we can use the real Hessian matrix as a starting guess, but it is often sufficient to start with the identity matrix...

# Broyden–Fletcher–Goldfarb–Shanno (BFGS) method



- however, since we only need the inverse of **B**, we can rewrite the update using the Sherman–Morrison formula,

$$\mathbf{B}_{k+1}^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) \mathbf{B}_k^{-1} \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \tag{5}$$

- as $\mathbf{B}_k^{-1}$ is symmetric, and both $\mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k$ and $\mathbf{s}_k^T \mathbf{y}_k$ are scalars, we can write

$$\boxed{\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \mathbf{B}_k^{-1} \cdot \nabla f(\mathbf{x}_k) && (6) \\ \mathbf{B}_{k+1}^{-1} &= \mathbf{B}_k^{-1} + \frac{(\mathbf{s}_k^T \mathbf{y}_k + \mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k)(\mathbf{s}_k \mathbf{s}_k^T)}{(\mathbf{s}_k^T \mathbf{y}_k)^2} - \frac{\mathbf{B}_k^{-1} \mathbf{y}_k \mathbf{s}_k^T + \mathbf{s}_k \mathbf{y}_k^T \mathbf{B}_k^{-1}}{\mathbf{s}_k^T \mathbf{y}_k} && (7) \end{aligned}}$$

BFGS UPDATES (IMPROVED)

and we never need to invert a matrix ever again.

- we can use the real Hessian matrix as a starting guess, but it is often sufficient to start with the identity matrix...
- ...at the first iteration, the update behaves like gradient descent, but in favourable circumstances, the approximate Hessian quickly approaches the values of the real Hessian

# Non-linear equations—Gradient Descent—again



- we can easily generalise the gradient-descent update to the many-variable case

# Non-linear equations—Gradient Descent—again



- we can easily generalise the gradient-descent update to the many-variable case
- our update shall be

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{p}} \tag{8}$$

GRADIENT DESCENT UPDATE

where $\alpha$ is a small, positive, adjustable parameter, and $\mathbf{p}$ is a unit vector chosen as a direction of function decrease

# Non-linear equations—Gradient Descent—again

- we can easily generalise the gradient-descent update to the many-variable case
- our update shall be

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{p}} \tag{8}$$

GRADIENT DESCENT UPDATE

where $\alpha$ is a small, positive, adjustable parameter, and $\mathbf{p}$ is a unit vector chosen as a direction of function decrease

- expanding the function around the new point,

$$f\left(\mathbf{x}_{k+1}\right) = f\left(\mathbf{x}_k\right) - \alpha \nabla f\left(\mathbf{x}_k\right)^T \mathbf{p} + \mathcal{O}\left(\alpha^2\right)$$

# Non-linear equations—Gradient Descent—again

- we can easily generalise the gradient-descent update to the many-variable case
- our update shall be

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{p}} \tag{8}$$

GRADIENT DESCENT UPDATE

where $\alpha$ is a small, positive, adjustable parameter, and **p** is a unit vector chosen as a direction of function decrease

- expanding the function around the new point,

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) - \alpha \nabla f(\mathbf{x}_k)^T \mathbf{p} + \mathcal{O}(\alpha^2)$$

- we find that, if the function were linear, the function value would decrease by an amount,

$$\begin{aligned} f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) &= -\alpha \nabla f(\mathbf{x}_k)^T \mathbf{p} \\ &= -\alpha m \end{aligned}$$

where $m = f(\mathbf{x}_k)^T \mathbf{p}$

- as a condition for choosing $\alpha$, we say that the function value has decreased by a sufficient amount if it decreases by some amount,

$$\boxed{f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) < -\alpha c m} \tag{9}$$

ARMIJO–GOLDSTEIN CONDITION

where $0 < c < 1$ is a control parameter (non-adjustable)

- as a condition for choosing $\alpha$, we say that the function value has decreased by a sufficient amount if it decreases by some amount,

$$\boxed{f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) < -\alpha c m} \tag{9}$$

ARMIJO–GOLDSTEIN CONDITION

where $0 < c < 1$ is a control parameter (non-adjustable)

- if the Armijo–Goldstein condition is not satisfied, $\alpha$ is reduced

$$\alpha \leftarrow \tau\alpha$$

where $0 < \tau < 1$ is another control parameter.

# Non-linear equations—Gradient Descent—again

- as a condition for choosing $\alpha$, we say that the function value has decreased by a sufficient amount if it decreases by some amount,

$$\boxed{f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) < -\alpha c m} \tag{9}$$

ARMIJO–GOLDSTEIN CONDITION

where $0 < c < 1$ is a control parameter (non-adjustable)

- if the Armijo–Goldstein condition is not satisfied, $\alpha$ is reduced

$$\alpha \leftarrow \tau\alpha$$

where $0 < \tau < 1$ is another control parameter.

- this method is known as a backtracking line search, because we start far from the original point with relatively large $\alpha$, and backtrack, decreasing $\alpha$ until the condition is satisfied.

# Non-linear equations—Gradient Descent—again

- as a condition for choosing $\alpha$, we say that the function value has decreased by a sufficient amount if it decreases by some amount,

$$\boxed{f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) < -\alpha c m} \tag{9}$$

ARMIJO–GOLDSTEIN CONDITION

where $0 < c < 1$ is a control parameter (non-adjustable)

- if the Armijo–Goldstein condition is not satisfied, $\alpha$ is reduced

$$\alpha \leftarrow \tau\alpha$$

where $0 < \tau < 1$ is another control parameter.

- this method is known as a backtracking line search, because we start far from the original point with relatively large $\alpha$, and backtrack, decreasing $\alpha$ until the condition is satisfied.
- in practice, the Newton and BFGS updates aren't always used directly—we use those equations to find a suitable search direction, then use a line search to find a suitable step size

# First-order saddles—eigenvector following

- sometimes we want to find stationary points that are not minima or maxima—they are generally harder to converge to

# First-order saddles—eigenvector following



- sometimes we want to find stationary points that are not minima or maxima—they are generally harder to converge to
- to find index-1 saddles (Hessian index 1, *i.e.*, one negative eigenvalue), we can use eigenvector following

# First-order saddles—eigenvector following

- sometimes we want to find stationary points that are not minima or maxima—they are generally harder to converge to
- to find index-1 saddles (Hessian index 1, *i.e.*, one negative eigenvalue), we can use eigenvector following
- the idea is to, starting at a point with Hessian index 1, maximise parallel to the negative eigenvector while minimising in all orthogonal directions

# First-order saddles—eigenvector following

- sometimes we want to find stationary points that are not minima or maxima—they are generally harder to converge to
- to find index-1 saddles (Hessian index 1, *i.e.*, one negative eigenvalue), we can use eigenvector following
- the idea is to, starting at a point with Hessian index 1, maximise parallel to the negative eigenvector while minimising in all orthogonal directions
- we did this in the one variable case of Newton's method by taking the absolute value of the second derivative

# First-order saddles—eigenvector following

- sometimes we want to find stationary points that are not minima or maxima—they are generally harder to converge to
- to find index-1 saddles (Hessian index 1, *i.e.*, one negative eigenvalue), we can use eigenvector following
- the idea is to, starting at a point with Hessian index 1, maximise parallel to the negative eigenvector while minimising in all orthogonal directions
- we did this in the one variable case of Newton's method by taking the absolute value of the second derivative
- to do so, we find the relevant eigenvector $\mathbf{v}_{\min}$ and take a small step in the direction the function is increasing, using, *e.g.*, a line search...

# First-order saddles—eigenvector following

- sometimes we want to find stationary points that are not minima or maxima—they are generally harder to converge to
- to find index-1 saddles (Hessian index 1, *i.e.*, one negative eigenvalue), we can use eigenvector following
- the idea is to, starting at a point with Hessian index 1, maximise parallel to the negative eigenvector while minimising in all orthogonal directions
- we did this in the one variable case of Newton's method by taking the absolute value of the second derivative
- to do so, we find the relevant eigenvector $\mathbf{v}_{\text{min}}$ and take a small step in the direction the function is increasing, using, *e.g.*, a line search...
- then minimise in the direction obtained by projecting the eigenvector out of the gradient vector, using *e.g.*, BFGS,

$$\boxed{\nabla f_{\perp} = \nabla f - \left(\nabla f^{T} \cdot \mathbf{v}_{\text{min}}\right) \mathbf{v}_{\text{min}}} \tag{10}$$

EIGENVECTOR FOLLOWING PROJECTED GRADIENT

# Eigenvalues and eigenvectors—Rayleigh–Ritz ratio

- how do we find this crucial eigenvector?

# Eigenvalues and eigenvectors—Rayleigh–Ritz ratio



- how do we find this crucial eigenvector?
- or large systems, inverting the Hessian to find its eigenvectors can become a bottleneck; to avoid this problem, we can make use of the Rayleigh–Ritz ratio,

$$\boxed{\begin{aligned} \lambda(\mathbf{v}) &= \frac{\mathbf{v}^T\mathbf{H}(\mathbf{x})\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \qquad (11) \\ \nabla\lambda(\mathbf{v}) &= \frac{\left(\mathbf{v}^T\mathbf{v}\right)\left(\mathbf{H}\mathbf{v}+\mathbf{H}^T\mathbf{v}\right) - 2\left(\mathbf{v}^T\mathbf{H}\mathbf{v}\right)\mathbf{v}}{\left(\mathbf{v}^T\mathbf{v}\right)^2} \qquad (12) \end{aligned}}$$

RAYLEIGH–RITZ RATIO & GRADIENT THEREOF

# Eigenvalues and eigenvectors—Rayleigh–Ritz ratio

- how do we find this crucial eigenvector?
- or large systems, inverting the Hessian to find its eigenvectors can become a bottleneck; to avoid this problem, we can make use of the Rayleigh–Ritz ratio,

$$\boxed{\begin{aligned} \lambda(\mathbf{v}) &= \frac{\mathbf{v}^T \mathbf{H}(\mathbf{x})\mathbf{v}}{\mathbf{v}^T\mathbf{v}} && (11) \\ \nabla\lambda(\mathbf{v}) &= \frac{\left(\mathbf{v}^T\mathbf{v}\right)\left(\mathbf{H}\mathbf{v} + \mathbf{H}^T\mathbf{v}\right) - 2\left(\mathbf{v}^T\mathbf{H}\mathbf{v}\right)\mathbf{v}}{\left(\mathbf{v}^T\mathbf{v}\right)^2} && (12) \end{aligned}}$$

RAYLEIGH–RITZ RATIO & GRADIENT THEREOF

- $\lambda(\mathbf{v})$ has a unique minimum when $\mathbf{v} \parallel \mathbf{v}_{\text{min}}$

# Eigenvalues and eigenvectors—Rayleigh–Ritz ratio

- how do we find this crucial eigenvector?
- or large systems, inverting the Hessian to find its eigenvectors can become a bottleneck; to avoid this problem, we can make use of the Rayleigh–Ritz ratio,

$$\boxed{\begin{aligned} \lambda(\mathbf{v}) &= \frac{\mathbf{v}^T \mathbf{H}(\mathbf{x})\mathbf{v}}{\mathbf{v}^T\mathbf{v}} & (11)\\ \nabla\lambda(\mathbf{v}) &= \frac{\left(\mathbf{v}^T\mathbf{v}\right)\left(\mathbf{H}\mathbf{v}+\mathbf{H}^T\mathbf{v}\right)-2\left(\mathbf{v}^T\mathbf{H}\mathbf{v}\right)\mathbf{v}}{\left(\mathbf{v}^T\mathbf{v}\right)^2} & (12)\end{aligned}}$$

RAYLEIGH–RITZ RATIO & GRADIENT THEREOF

- $\lambda(\mathbf{v})$ has a unique minimum when $\mathbf{v} \parallel \mathbf{v}_{\text{min}}$
- we can search for the minimum with gradient descent, BFGS...

# Eigenvalues and eigenvectors—Rayleigh–Ritz ratio



- how do we find this crucial eigenvector?
- or large systems, inverting the Hessian to find its eigenvectors can become a bottleneck; to avoid this problem, we can make use of the Rayleigh–Ritz ratio,

$$\boxed{\begin{aligned} \lambda(\mathbf{v}) &= \frac{\mathbf{v}^T \mathbf{H}(\mathbf{x})\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \qquad (11) \\ \nabla\lambda(\mathbf{v}) &= \frac{\left(\mathbf{v}^T\mathbf{v}\right)\left(\mathbf{H}\mathbf{v} + \mathbf{H}^T\mathbf{v}\right) - 2\left(\mathbf{v}^T\mathbf{H}\mathbf{v}\right)\mathbf{v}}{\left(\mathbf{v}^T\mathbf{v}\right)^2} \qquad (12) \end{aligned}}$$

RAYLEIGH–RITZ RATIO & GRADIENT THEREOF

- $\lambda(\mathbf{v})$ has a unique minimum when $\mathbf{v} \parallel \mathbf{v}_{\text{min}}$
- we can search for the minimum with gradient descent, BFGS...
- we can also use approximate Hessian matrices if exact ones are not available