Consider the problem to minimize

$$\min \quad f(\boldsymbol{x}) \triangleq \frac{1}{2}\boldsymbol{x}^T \boldsymbol{H}\boldsymbol{x} - \boldsymbol{c}^T \boldsymbol{x}$$

where $\boldsymbol{H}$ is a symmetric positive definite matrix.

- *Condition number* of the Hessian matrix controls the convergence rate of steepest descent method.
- Faster convergence if the Hessian matrix is $\boldsymbol{I}$
- Let $\boldsymbol{H} = \boldsymbol{L}\boldsymbol{L}^T$ be the Cholesky decomposition of $\boldsymbol{H}$
- Define $\boldsymbol{y} = \boldsymbol{L}^T\boldsymbol{x}$. Therefore, the function $f(\boldsymbol{x})$ is transformed to the function $h(\boldsymbol{y})$.

$$h(\boldsymbol{y}) \triangleq f(\boldsymbol{L}^{-T}\boldsymbol{y})$$

$$
\begin{aligned}
h(\boldsymbol{y}) &= f(\boldsymbol{L}^{-T}\boldsymbol{y}) \\
&= \frac{1}{2}\boldsymbol{y}^T\boldsymbol{L}^{-1}\boldsymbol{H}\boldsymbol{L}^{-T}\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{L}^{-T}\boldsymbol{y} \\
&= \frac{1}{2}\boldsymbol{y}^T\boldsymbol{L}^{-1}\boldsymbol{L}\boldsymbol{L}^T\boldsymbol{L}^{-T}\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{L}^{-T}\boldsymbol{y} \\
&= \frac{1}{2}\boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{L}^{-T}\boldsymbol{y}
\end{aligned}
$$

- The Hessian matrix of $h(\boldsymbol{y})$ is $\boldsymbol{I}$
- Let us apply steepest descent method in $\boldsymbol{y}$-space

$$
\begin{aligned}
\boldsymbol{y}^{k+1} &= \boldsymbol{y}^k - \nabla h(\boldsymbol{y}^k) \\
&= \boldsymbol{y}^k - \boldsymbol{L}^{-1}\nabla f(\boldsymbol{L}^{-T}\boldsymbol{y}^k) \\
\therefore \boldsymbol{L}^{-T}\boldsymbol{y}^{k+1} &= \boldsymbol{L}^{-T}\boldsymbol{y}^k - \boldsymbol{L}^{-T}\boldsymbol{L}^{-1}\nabla f(\boldsymbol{L}^{-T}\boldsymbol{y}^k) \\
\therefore \boldsymbol{x}^{k+1} &= \boldsymbol{x}^k - \boldsymbol{H}^{-1}\nabla f(\boldsymbol{x}^k)
\end{aligned}
$$

Newton Method

Consider the problem,

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x})$$

- Let $f \in \mathcal{C}^2$ and f be bounded below.
- Use second order information to find a descent direction
- At every iteration, use Taylor series to approximate f at $\boldsymbol{x}^k$ by a quadratic function and find the minimum of this quadratic function to get $\boldsymbol{x}^{k+1}$

$$\begin{aligned} f(\boldsymbol{x}) \approx f_q(\boldsymbol{x}) &= f(\boldsymbol{x}^k) + \boldsymbol{g}^{k^T}(\boldsymbol{x} - \boldsymbol{x}^k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^k)^T \boldsymbol{H}^k (\boldsymbol{x} - \boldsymbol{x}^k) \\ \boldsymbol{x}^{k+1} &= \arg\min_{\boldsymbol{x}} \; f_q(\boldsymbol{x}) \end{aligned}$$

- $\nabla f_q(\boldsymbol{x}) = 0 \;\Rightarrow\; \boldsymbol{x}^{k+1} = \boldsymbol{x}^k - (\boldsymbol{H}^k)^{-1}\boldsymbol{g}^k$ (assuming $\boldsymbol{H}^k$ is invertible)

$$x^{k+1} = x^k - (H^k)^{-1} g^k \quad \text{is of the form,} \quad x^{k+1} = x^k + \alpha^k d^k$$

- Classical Newton Method:
  - Newton Direction: $d_N^k = -(H^k)^{-1} g^k$
  - Step Length: $\alpha^k = 1$
- Is d_N^k a descent direction?
  $g^{kT} d_N^k = -g^{kT} (H^k)^{-1} g^k < 0$ if H^k is positive definite.

  d_N^k is a descent direction if H^k is positive definite
- Consider the problem to minimize, $f(x) = \frac{1}{2} x^T H x - c^T x$
  where H is a symmetric positive definite matrix.
  $g(x) = 0 \;\Rightarrow\; x^* = H^{-1} c$ is a strict local minimum
  Let $x^0 \in \mathbb{R}^n$ be any point. $g(x^0) = Hx^0 - c,\; H(x^0) = H$.
  Using classical Newton method,
  $x^1 = x^0 - H^{-1}(Hx^0 - c) = H^{-1} c = x^*$.

  Using classical newton method, the minimum of a strictly convex quadratic function (with invertible Hessian matrix) is attained in one iteration from *any starting point*.

Classical Newton Algorithm

(1) Initialize $\boldsymbol{x}^0$ and ϵ, set $k := 0$.

(2) **while** $\|\boldsymbol{g}^k\| > \epsilon$

  (a) $\boldsymbol{d}^k = -(\boldsymbol{H}^K)^{-1}\boldsymbol{g}^k$

  (b) $\alpha^k = 1$

  (c) $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k$

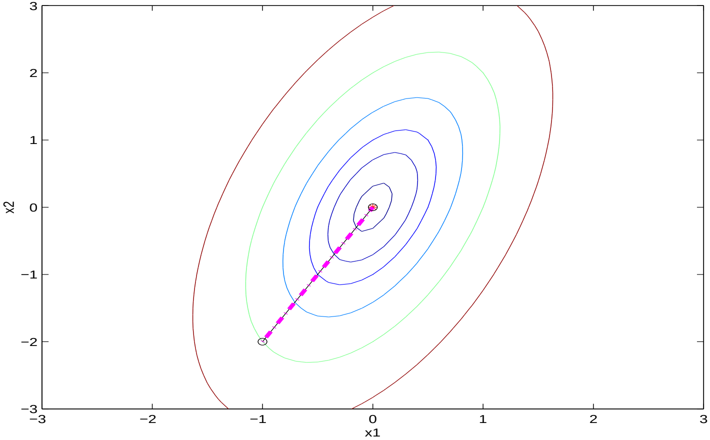
  (d) $k := k + 1$

endwhile

Output : $\boldsymbol{x}^* = \boldsymbol{x}^k$, a stationary point of $f(\boldsymbol{x})$.

Example:

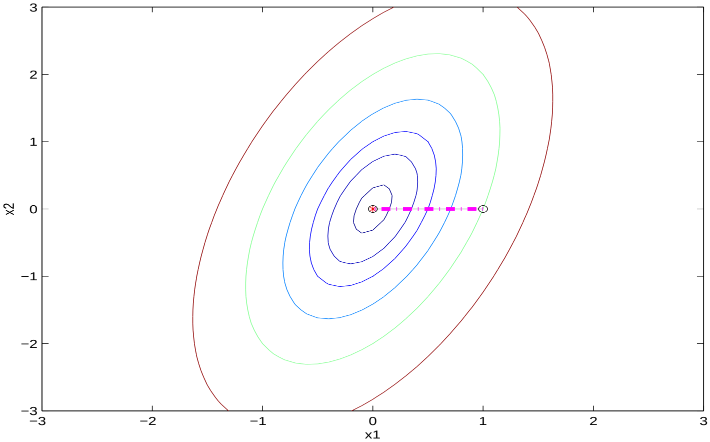
$$\min \quad f(\boldsymbol{x}) \triangleq 4x_1^2 + x_2^2 - 2x_1x_2$$



Classical Newton algorithm applied to $f(\boldsymbol{x})$ converges to $\boldsymbol{x}^*$ in one iteration from any starting point

Example:

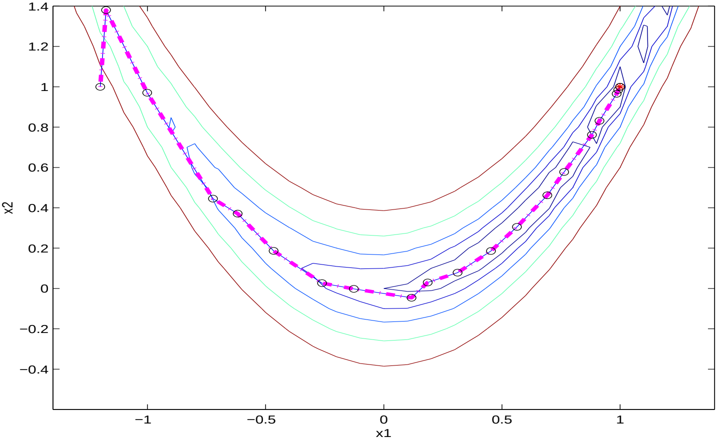
$$\min \quad f(\boldsymbol{x}) \triangleq 4x_1^2 + x_2^2 - 2x_1x_2$$



Classical Newton algorithm applied to $f(\boldsymbol{x})$ converges to $\boldsymbol{x}^*$ in one iteration from any starting point

Example (Rosenbrock function):

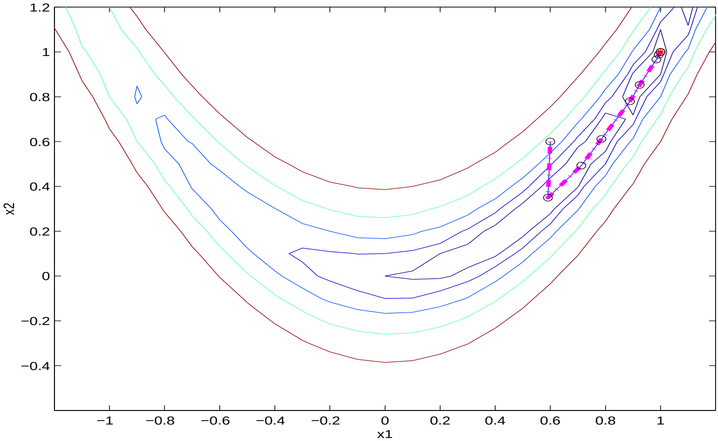
$$\min \;\; f(\boldsymbol{x}) \triangleq 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$



Behaviour of classical Newton algorithm (with backtracking line search) applied to $f(\boldsymbol{x})$ using $\boldsymbol{x}^0 = (-1.2, 1)^T$

Example (Rosenbrock function):

$$\min \quad f(\boldsymbol{x}) \stackrel{\Delta}{=} 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$



Behaviour of classical Newton algorithm (with backtracking line search) applied to $f(\boldsymbol{x})$ using $\boldsymbol{x}^0 = (0.6, 0.6)^T$

Classical Newton Algorithm

(1) Initialize $\boldsymbol{x}^0$ and ϵ, set $k := 0$.

(2) **while** $\|\boldsymbol{g}^k\| > \epsilon$

  (a) $\boldsymbol{d}^k = -(\boldsymbol{H}^K)^{-1}\boldsymbol{g}^k$

  (b) $\alpha^k = 1$

  (c) $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k$

  (d) $k := k + 1$

endwhile

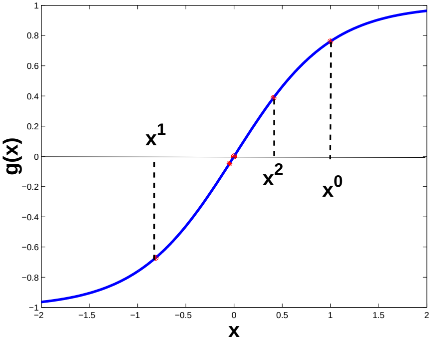
Output : $\boldsymbol{x}^* = \boldsymbol{x}^k$, a stationary point of $f(\boldsymbol{x})$.

- Requires $O(n^3)$ computational effort for every iteration (Step 2(a))
- *No guarantee* that $\boldsymbol{d}^k$ is a descent direction
- *No guarantee* that $f(\boldsymbol{x}^{k+1}) < f(\boldsymbol{x}^k)$ (no line search)
- Sensitive to initial point (for non-quadratic functions)

Consider the problem,

$$\min_{x \in \mathbb{R}} \quad \log(e^x + e^{-x})$$

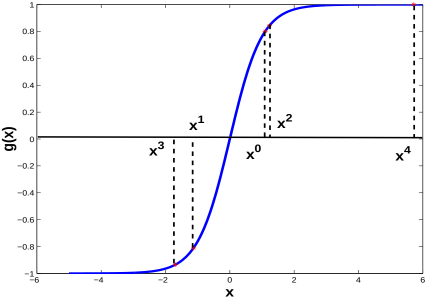
- $f(x) = \log(e^x + e^{-x})$
- $g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$



Consider the problem,

$$\min_{x \in \mathbb{R}} \quad \log(e^x + e^{-x})$$

- $f(x) = \log(e^x + e^{-x})$
- $g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$



Classical Newton algorithm does not converge with this initialization of x^0

Definition

An iterative optimization algorithm is said to be locally convergent if for each solution $\boldsymbol{x}^*$, there exists $\delta > 0$ such that for any initial point $\boldsymbol{x}^0 \in B(\boldsymbol{x}^*, \delta)$, the algorithm produces a sequence $\{\boldsymbol{x}^k\}$ which converges to $\boldsymbol{x}^*$.

- Classical Newton algorithm is locally convergent

Let $f : \mathbb{R} \to \mathbb{R}$, $f \in \mathcal{C}^2$.
Consider the problem:

$$\min \ f(x)$$

Let $x^* \in \mathbb{R}$ be such that $g(x^*) = 0$ and $g'(x^*) > 0$.
Assume that x^0 is *sufficiently* close to x^*.
Suppose we apply classical Newton algorithm to minimize $f(x)$.

Also, we want $\beta|x^k - x^*| < 1 \;\forall\; k$. That is,

$$\begin{aligned} |x^k - x^*| &< 1/\beta \;\forall\; k \\ \Rightarrow \quad x^k &\in (x^* - 1/\beta, x^* + 1/\beta) \end{aligned}$$

Therefore, choose $x^0 \in (x^* - \eta, x^* + \eta) \cap (x^* - 1/\beta, x^* + 1/\beta)$

Does $\{x^k\}$ converge to x^* if x^0 is chosen using this approach?

We have

$$\begin{aligned} |x^k - x^*| &\leq \beta|x^{k-1} - x^*|^2 \\ \therefore \beta|x^k - x^*| &\leq (\beta|x^0 - x^*|)^{2^k} \\ \therefore |x^k - x^*| &\leq \frac{1}{\beta}(\underbrace{\beta|x^0 - x^*|}_{<1})^{2^k} \end{aligned}$$

Therefore,

$$\lim_{k\to\infty} |x^k - x^*| = 0$$

Not a practical approach to choose x^0

Theorem

Let $f : \mathbb{R} \to \mathbb{R}$, $f \in \mathcal{C}^3$. Let $x^* \in \mathbb{R}$ be such that $g(x^*) = 0$ and $g'(x^*) > 0$. Then, provided x^0 is sufficiently close to x^*, the sequence $\{x^k\}$ generated by classical Newton algorithm converges to x^* with an order of convergence two.

Initialization of x^0 requires knowledge of x^*!