Taylor's Thm — Preliminary for Newton for Systems and OPTIMIZATION.

Consider

$$f: D \subset \mathbb{R}^2 \to \mathbb{R}$$
$$(x_1, x_2) \to f(x_1, x_2)$$

Suff. diff. on D and $\vec{p} = (p_1, p_2) \in D$.

Then, for any $(x_1, x_2) = \vec{x} \in D$, there exists $\vec{\xi} = (\xi_1, \xi_2)$

Such that ξ_1 is btw. x_1 and p_1
ξ_2 is btw x_2 and p_2

and

$$f(\vec{x}) = f(x_1, x_2) = f(p_1, p_2) + \frac{\partial f}{\partial x_1}(\vec{p})(x_1 - p_1) + \frac{\partial f}{\partial x_2}(\vec{p})(x_2 - p_2)$$

$$+ \frac{\partial^2 f}{\partial x_1^2}(\vec{p}) \frac{(x_1 - p_1)^2}{2} + \frac{\partial^2 f}{\partial x_1 \partial x_2}(\vec{p})(x_1 - p_1)(x_2 - p_2) + \frac{\partial^2 f}{\partial x_2^2}(\vec{p}) \frac{(x_2 - p_2)^2}{2}$$

$$+ R_n(\vec{x}(\vec{\xi})) \quad \text{(depends on } \vec{\xi}\text{)}$$

$$+ \frac{1}{3!} \sum_{i,j,k=1}^{2} \frac{\partial^3 f}{\partial x_i \partial x_j \partial x_k} (x_i - p_i)(x_j - p_j)(x_k - p_k)$$

$$= f(\vec{p}) + \nabla f(\vec{p}) \cdot (\vec{x} - \vec{p}) + \cdots$$

If
$$\vec{F}: D \subset \mathbb{R}^2 \to \mathbb{R}^2$$
$$(x_1, x_2) \to \vec{F}(x_1, x_2) = (f_1(x_1, x_2), f_2(x_1, x_2))$$

Suff. diff. on D and $\vec{p} \in D$, then for any $\vec{x} \in D$, there is $\vec{\xi}$ with comps. in betw.

and

$$\vec{F}(x_1, x_2) = \vec{F}(p_1, p_2) + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} (\vec{p}) \begin{bmatrix} x_1 - p_1 \\ x_2 - p_2 \end{bmatrix} +$$

See next page

$$
+\begin{bmatrix} \dfrac{\partial^2 f_1}{\partial x_1^2}(\vec{p})\,\dfrac{(x_1-p_1)^2}{2} + \dfrac{\partial^2 f_1}{\partial x_1 \partial x_2}(\vec{p})\,(x_1-p_1)(x_2-p_2) + \dfrac{\partial^2 f_1}{\partial x_2^2}(\vec{p})\,\dfrac{(x_2-p_2)^2}{2} \\ \dfrac{\partial^2 f_2}{\partial x_1^2}(\vec{p})\,\dfrac{(x_1-p_1)^2}{2} + \dfrac{\partial^2 f_2}{\partial x_1 \partial x_2}(\vec{p})\,(x_1-p_1)(x_2-p_2) + \dfrac{\partial^2 f_2}{\partial x_2^2}(\vec{p})\,\dfrac{(x_2-p_2)^2}{2} \end{bmatrix}
$$

$+ \ldots$ Error term

If $\vec{F} = \nabla f$, then $\nabla f(\vec{p}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\vec{p}) \\ \frac{\partial f_2}{\partial x_2}(\vec{p}) \end{pmatrix}$

$$
\nabla f(\vec{x}) = \nabla f(\vec{p}) + \underbrace{\begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2}(\vec{p}) & \frac{\partial^2 f_1}{\partial x_1 \partial x_2}(\vec{p}) \\ \frac{\partial^2 f_2}{\partial x_2 \partial x_1}(\vec{p}) & \frac{\partial^2 f_2}{\partial x_2^2}(\vec{p}) \end{bmatrix}}_{\text{"Hessian Matrix"} \;\equiv H(\vec{p})} \begin{bmatrix} x_1 - p_1 \\ x_2 - p_2 \end{bmatrix} + \text{Error term.}
$$

or

$$
\boxed{Df(\vec{x}) = Df(\vec{p}) + D^2 f(\vec{p})\,(\vec{x}-\vec{p}) + \text{Error term } O(\|\vec{x}-\vec{p}\|^2)} \quad (*)
$$

cont. of p. derivs is required.

$$
\Rightarrow \quad \|Df(\vec{x}) - Df(\vec{p}) - D^2 f(\vec{p})\,(\vec{x}-\vec{p})\| \le M_1 \|\vec{x}-\vec{p}\|^2 .
$$

$$
= \begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & \frac{\partial^2 f_1}{\partial x_1 \partial x_2} & \frac{\partial^2 f_1}{\partial x_2^2} \\ \frac{\partial^2 f_2}{\partial x_1^2} & \frac{\partial^2 f_2}{\partial x_1 \partial x_2} & \frac{\partial^2 f_2}{\partial x_2^2} \end{bmatrix} \begin{bmatrix} \frac{(x_1-p_1)^2}{2} \\ (x_1-p_1)(x_2-p_2) \\ \frac{(x_2-p_2)^2}{2} \end{bmatrix}
$$

$$
= \frac{1}{2} \sum_{i,j=1}^{2} \frac{\partial^2 f}{\partial x_i \partial x_j} (x_i - p_i)(x_j - p_j)
$$

do search in this expression

From $(*)$

$\nabla f(\vec{x}) = 0$

$\Rightarrow$

$\vec{x} = \vec{p} - H^{-1}(\vec{p})\,\nabla f(\vec{p})$

Funct. Iteration

$\vec{p}_n = \vec{p}_{n-1} - H^{-1}(\vec{p}_{n-1})\,\nabla f(\vec{p}_{n-1})$

or $\boxed{\vec{p}_n = \vec{p}_{n-1} + \vec{y}_{n-1}}$

where $H(\vec{p}_{n-1})\,\vec{y}_{n-1} = -\nabla f(\vec{p}_{n-1})$

Nonlinear Systems – Newton Method

Exercise #5 (Sect 10.1)

Find the roots of

$$\begin{cases} f_1(x_1,x_2) = x_1^2 - 10x_1 + x_2^2 + 8 = 0 & (1) \\ f_2(x_1,x_2) = x_1 x_2^2 + x_1 - 10x_2 + 8 = 0 & (2) \end{cases}$$

(1,1) is a root

Idea is use Newton for $\boxed{\vec{F}(x_1,x_2) = (f_1(x_1,x_2), f_2(x_1,x_2)) = 0.}$

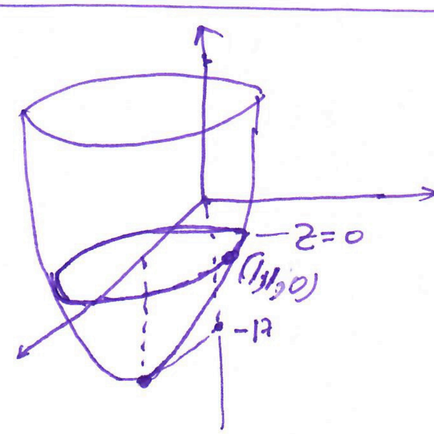
1-D $\quad f(x) = 0 \longrightarrow g(x) = x - \dfrac{f(x)}{f'(x)}$

2-D $\quad \vec{F}(\vec{x}) = 0 \xrightarrow{\text{Ask students}} \vec{G}(\vec{x}) = \underbrace{\vec{x} - \left(J(\vec{x})\right)^{-1} \vec{F}(\vec{x})}_{\text{Gives quadratic convergence!}}$

Follow Nw_7

$f_1(x_1,x_2) = (x_1-5)^2 + x_2^2 - 17$

All roots located on ellipse obtained by intersecting $z=0$ with surface



10.2 Nonlinear Systems Newton Method.

Back to #5 Sect. 10.1

$$\begin{cases} \underbrace{x_1^2 - 10x_1 + x_2^2 + 8}_{=f_1(x_1,x_2)} = 0 \\ \underbrace{x_1x_2^2 + x_1 - 10x_2 + 8}_{=f_2(x_1,x_2)} = 0 \end{cases}$$

(1,1) is a root.

$$\vec{F}(\vec{x}) = \vec{F}(x_1,x_2) = (f_1(x_1,x_2),\ f_2(x_1,x_2)) = \vec{0}$$

Newt. method

$$\vec{G}(\vec{x}) = \vec{x} - J^{-1}(\vec{x})\,\vec{F}(\vec{x})$$

In our example,

$$J(\vec{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}) & \frac{\partial f_1}{\partial x_2}(\vec{x}) \\ \frac{\partial f_2}{\partial x_1}(\vec{x}) & \frac{\partial f_2}{\partial x_2}(\vec{x}) \end{bmatrix} = \begin{bmatrix} 2x_1-10 & 2x_2 \\ x_2^2+1 & 2x_1x_2-10 \end{bmatrix}$$

Functional Iteration:

Initial Guess: $\vec{x}^0 = (x_1^0, x_2^0) \overset{\text{for example}}{=} (1/2, 1/2)$

1st step: $J(\vec{x}^0)\,\vec{y}^{(0)} = -\vec{F}(\vec{x}^0)$

$$\begin{bmatrix} 2x_1^0-10 & 2x_2^0 \\ (x_2^0)^2+1 & 2x_1^0x_2^0-10 \end{bmatrix}\begin{bmatrix} y_1^0 \\ y_2^0 \end{bmatrix} = -\begin{bmatrix} (x_1^0)^2-10x_1^0+(x_2^0)^2+8 \\ x_1^0(x_2^0)^2+x_1^0-10x_2^0+8 \end{bmatrix}$$

Solve this linear system to obtain $\vec{y}^{(0)} = \begin{pmatrix} y_1^0 \\ y_2^0 \end{pmatrix}$.

2nd step: $\vec{X}^{(1)} = \vec{X}^{(0)} + \vec{y}^{(0)}$

An updated approx. $\vec{x}^{(1)}$ to the root $\vec{X}^*$ has been obtained.

3rd step: check tolerance satisfaction.

$$\| \vec{X}^{(1)} - \vec{X}^{(0)} \| = \left\| \begin{pmatrix} x_i^1 - x_i^0 \\ y_i^1 - y_i^0 \end{pmatrix} \right\| \leq TOL. \quad \Big| \quad \text{or simply } \|\vec{y}^{(0)}\| \leq TOL$$

If yes. stop.

else. $\vec{X}^0 = \vec{X}^{(1)}$.

repeat steps 1-3.

Math 410 - Introduction to Numerical Methods
Introduction to Optimization - Fall 2009

Prof. Vianey Villamizar

1 Multi-dimensional Newton's Method as Minimizer

Suppose that we are to minimize a multi-variable function $f : \mathbb{R}^n \to \mathbb{R}$. Here we will use the following notation $\mathbf{x} = (x_1, x_2, ..., x_n)$. As in the one-dimensional case, we will use the derivatives of this function to construct Newton's method and approximate a local minimizer $\mathbf{x}^*$ with an arbitrary degree of accuracy. As before, we assume that we can evaluate $f(\mathbf{x})$, all of its partial derivatives $\frac{\partial f}{\partial x_i}$, and all of its second order partial derivatives $\frac{\partial^2 f}{\partial x_i \partial x_j}$ in the vicinity of the minimizer $\mathbf{x}^*$.

In order to construct Newton's method we must introduce some notation about the derivatives of a multi-variable function. If $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, then the function ∇f defined by

$$\nabla f(\mathbf{x}) = Df(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \tag{1.1}$$

is called the *gradient* of f. The gradient of f is vector in $\mathbb{R}^n$, and we can regard it as an $n \times 1$ matrix. Now, if Df is differentiable, we say that f is twice differentiable, and we write the derivative of Df as

$$D^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \frac{\partial^2 f}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \tag{1.2}$$

The matrix $D^2 f(\mathbf{x})$ is called the *Hessian* matrix of f at $\mathbf{x}$. If f is twice continuously differentiable at $\mathbf{x}$ then the partial derivatives can be interchanged, that is, $\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$ for all $i, j = 1, ..., n$. In that case, then the Hessian matrix $D^2 f$ is symmetric.

As before, we are going to construct a sequence of points $\mathbf{x}_k$ that will hopefully converge to the actual minimizer $\mathbf{x}^*$. At the k^{th} iteration, we define a closed neighborhood by $B_k := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}^*\| \leq \|\mathbf{x}_k - \mathbf{x}^*\|\}$. In order to prove the convergence and determine its rate, we make the following assumptions. First, $f \in C^3(\mathbb{R}^n)$, which implies (by the extreme value theorem and the definition of a third derivative) that for a closed ball B_k there exists a positive constant M_1 such that,

$$\|Df(\mathbf{x}) - Df(\mathbf{x}_k) - D^2 f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)\| \leq M_1 \|\mathbf{x} - \mathbf{x}_k\|^2 \quad \text{for all} \quad \mathbf{x} \in B_k. \tag{1.3}$$

The second assumption is that the concavity of the function f remains bounded away from zero within each B_k. In the context of multi-variable functions, this requirement implies that $(D^2 f(\mathbf{x}))^{-1}$ exists for all $\mathbf{x} \in B_k$, and that there is a positive constant M_2 such that

$$\|(D^2 f(x))^{-1}\| \leq M_2 \quad \text{for all} \quad \mathbf{x} \in B_k. \tag{1.4}$$

Now, assuming that Df can be expanded in its multi-dimensional Taylor series about the current point $\mathbf{x}_k$, we obtain,

$$Df(\mathbf{x}) = Df(\mathbf{x}_k) + D^2 f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) + \mathcal{O}(\|\mathbf{x} - \mathbf{x}_k\|^2).$$

Since $\mathbf{x}^*$ is a local minimum, then $Df(\mathbf{x}^*) = \mathbf{0}$. By evaluating the above equation at $\mathbf{x} = \mathbf{x}^*$ we obtain,

$$\mathbf{0} = Df(\mathbf{x}^*) = Df(\mathbf{x}_k) + D^2 f(\mathbf{x}_k)(\mathbf{x}^* - \mathbf{x}_k) + \mathcal{O}(\|\mathbf{x}^* - \mathbf{x}_k\|^2).$$

Rearranging some terms we also find that

$$\mathbf{x}^* = \mathbf{x}_k - (D^2 f(\mathbf{x}_k))^{-1} Df(\mathbf{x}_k) + \mathcal{O}(\|\mathbf{x}^* - \mathbf{x}_k\|^2).$$

At this point we neglect the term $\mathcal{O}(\|\mathbf{x}^* - \mathbf{x}_k\|^2)$, and instead of obtaining the minimizer $\mathbf{x}^*$ at the left-hand side, we obtain the Newton's algorithm to calculate a new approximation to the minimizer, i.e.,

$$\textbf{\textit{Newton's Algorithm:}} \qquad \mathbf{x}_{k+1} = \mathbf{x}_k - (D^2 f(\mathbf{x}_k))^{-1} Df(\mathbf{x}_k). \tag{1.5}$$

Now we proceed to prove the convergence of this method and estimate the rate at which it converges in the context of multi-dimensional functions. We present these results in the form of a theorem.

Theorem 1.1. *Suppose that $f \in C^3(\mathbb{R}^n)$, and $\boldsymbol{x}^*$ is a point such that $Df(\boldsymbol{x}^*) = 0$ and that conditions (1.3)-(1.4) are satisfied in the vicinity of $\boldsymbol{x}^*$. Then, for all x_0 sufficiently close to x^*, Newton's method (1.5) is well defined for all k, and converges to x^* with order of convergence at least* 2.

Proof. We define the error at the step k by $\mathbf{e}_k := \mathbf{x}^* - \mathbf{x}_k$. From (1.3) evaluated at $\mathbf{x} = \mathbf{x}^*$ then

$$\|D^2 f(\mathbf{x})\mathbf{e}_k + Df(\mathbf{x}_k)\| \leq M_1 \|\mathbf{e}_k\|^2.$$

On the other hand, subtracting $\mathbf{x}^*$ from both sides of Newton's algorithm (1.5) and using the properties of the norm, we obtain

$$\|\mathbf{e}_{k+1}\| = \|\mathbf{e}_k + (D^2 f(\mathbf{x}_k))^{-1} Df(\mathbf{x}_k)\| \leq \|(D^2 f(\mathbf{x}_k))^{-1}\| \|D^2 f(\mathbf{x})\mathbf{e}_k + Df(\mathbf{x}_k)\|$$

Now, combining the previous inequalities and condition (1.4), we obtain the desired result, i.e.,

$$\|\mathbf{e}_{k+1}\| \leq \alpha \|\mathbf{e}_k\|^2 \quad \text{for all} \quad k = 0, 1, 2, ... \qquad \text{where} \quad \alpha = M_1 M_2. \tag{1.6}$$

If there is convergence, then the above inequality implies that the order of convergence is at least 2. However, convergence is not guaranteed in general. Suppose that $\alpha\|\mathbf{e}_0\| < 1$, then multiply the inequality (1.6) by α and use it recursively to obtain,

$$\alpha\|\mathbf{e}_k\| \leq (\alpha\|\mathbf{e}_{k-1}\|)^2 \leq (\alpha\|\mathbf{e}_{k-2}\|)^4 \leq ... \leq (\alpha\|\mathbf{e}_0\|)^{2^k} \to 0 \quad \text{as} \quad k \to \infty.$$

Therefore convergence is guaranteed if the starting point $\mathbf{x}_0$ is near the actual minimizer $\mathbf{x}^*$; near enough so that $\alpha|\mathbf{x}^* - \mathbf{x}_0| < 1$. □

Newton Method

$$\boxed{G(\vec{x}) = \vec{x} - [J(\vec{x})]^{-1}\vec{F}(\vec{x})} \quad (5.1)$$

$$\boxed{\vec{x}^{(k)} = \vec{G}(\vec{x}^{(k-1)}) = \vec{x}^{(k-1)} - [J(\vec{x}^{k-1})]^{-1}\vec{F}(\vec{x}^{k-1})} \quad (5.2)$$

(5.2) is computed in two steps.

1st we find $\vec{y}^{k-1}$ such that

I $\boxed{J(\vec{x}^{k-1})\,\vec{y}^{(k-1)} = -\vec{F}(\vec{x}^{k-1})}$

and then

II $\boxed{\vec{x}^{(k)} = \vec{x}^{(k-1)} + \vec{y}^{(k-1)}}$

STOP CRITERIA: $\|\vec{y}^{(k-1)}\| = \|\vec{x}^{(k)} - \vec{x}^{(k-1)}\| \le TOL$

Burden-Faires

Thm 10.7 (book pp. 612)

$\vec{G}: D \subset \mathbb{R}^n \to \mathbb{R}^n$

$\vec{G}(\vec{x}) = (g_1(\vec{x}), .., g_n(\vec{x}))$

$$\boxed{\|\vec{x}^k - \vec{p}\|_\infty \le \frac{n^2 M}{2}\|\vec{x}^{(k-1)} - \vec{p}\|_\infty}$$

1. $G(\vec{p}) = \vec{p}$
2. $\frac{\partial g_i}{\partial x_j}$ conts on $N_\delta = \{\vec{x} : \|\vec{x} - \vec{p}\| < \delta\}$. for all i,j.
3. $\frac{\partial^2 g_i}{\partial x_j \partial x_k}$ is cont and $|\partial^2 g_i(\vec{x})/(\partial x_j \partial x_k)| \le M$, for $\vec{x} \in N_\delta$, for all i,j,k.
4. $\frac{\partial g_i}{\partial x_k}(\vec{p}) = 0$, for $i = 1,2,..,n$, $k = 1,2,..,n$

$\Rightarrow \vec{x}^{(k)} = \vec{G}(\vec{x}^{k-1}) \longrightarrow \vec{p}$, for $\vec{x}_0$ such that $\|\vec{x}_0 - p\| < \hat{\delta} < \delta$.

Quadratically

Omit this. Work on Book's exercise #5

Example: $\vec{F}(\vec{x}) = \vec{0}$

where $\vec{F}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}))$

and

$$f_1(x_1, x_2, x_3) = 3x_1 - \cos(x_2 x_3) - \frac{1}{2}$$

$$f_2(x_1, x_2, x_3) = x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06$$

$$f_3(x_1, x_2, x_3) = e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3}$$

Jacobian matrix

$$J(\vec{x}) = \begin{bmatrix} 3 & x_3 \sin(x_2 x_3) & x_2 \sin(x_2 x_3) \\ 2x_1 & -162(x_2 + 0.1) & \cos x_3 \\ -x_2 e^{-x_1 x_2} & -x_1 e^{-x_1 x_2} & 20 \end{bmatrix}$$

Initial estimate $\vec{x}^{(0)} = \begin{pmatrix} 0.1 \\ 0.1 \\ -0.1 \end{pmatrix}$

$\|\vec{x}^{(4)} - \vec{x}^{(3)}\| = 1.24 \times 10^{-5}$

and $\vec{x}^{(4)} = \vec{x}^{(5)}$ for the computer used in book.