Intro to Computational Math, Spring 2026
Section 7, March 10 (not due or graded).

TL;DR: The first section motivates the root finding problem and explains conditioning. If you feel particularly motivated already, feel free to skip. The second section gives some more information on Taylor series. If this is a strong suit, feel free to skip as well.

The main aspects of this section focus on two methods for finding roots: Fixed point and Newton's. Fixed point is motivated by noting that

$$f(r) = 0 \Longleftrightarrow g(r) = r, \text{ with } g(x) := x - f(x).$$

Therefore, we may transform out function and look for fixed points. The method for finding fixed points is simple. Start with an initial guess x_0 and take the next iterative to be the function evaluate at the current one. This method can be shown to exhibit linear convergence, where

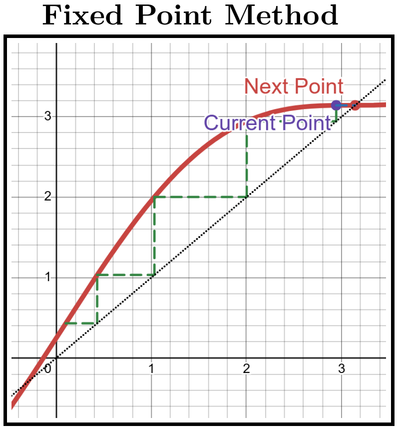
$$\varepsilon_k = x_k - r, \quad |\varepsilon_{k+1}| \approx g'(r)|\varepsilon_k|.$$

In order for convergence to hold, we expect that $g'(r) < 1$.

As far as Newton's method goes, we solve the root finding problem by taking linear approximations at a point, and finding the root of our approximation. This method is typically has better convergence, in that more cases have convergence and the rate is faster (by an order of magnitude). Under certain conditions, we may expect

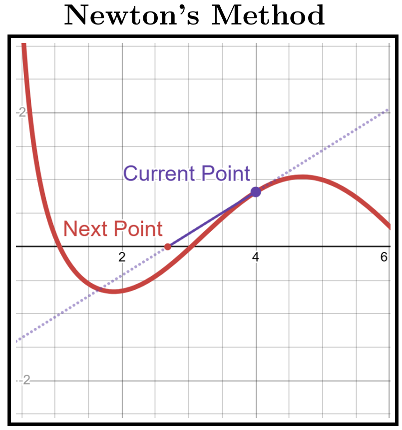
$$|\varepsilon_{k+1}| \approx \frac{1}{2}\frac{f''(r)}{f'(r)}|\varepsilon_k|^2.$$

Fixed Point Method



$x_{k+1} = g(x_k)$

Newton's Method



$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$

Click on the titles of the graphs to pull up an interactive Desmos where you can apply each method on a generic function!

For the generalization of Newton's method to the multivariate setting i.e. $f : \mathbb{R}^n \to \mathbb{R}^m$, we want

$$x_{k+1} \text{ "solves" } \quad 0 = f(x_k) + J_f(x_k)(x - x_k)$$

in the sense of least squares/utilizing normal equations, meaning $x_{k+1} = x_k + \text{argmin}\,\|Ax - b\|_2^2$ with $A = J_f(x_k)$ and $b = -f(x_k)$

1 The root finding problem:

Often in both applied and theoretical mathematics, we may benefit from finding the roots of a function. Everything from representing a polynomial in its factored form, to solving to eigenvalue problem, to finding stationary points in the framework of optimization benefit from root finding.

Definition 1.1

*Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous function. Then $r \in \mathbb{R}^d$ is a **root** of the function f if*

$$f(r) = 0.$$

If $f : \mathbb{R} \to \mathbb{R}$ is k times differentiable, then r is a root of multiplicity $m \leq k$ if

$$f(r) = 0,\ f'(r) = 0,\ f''(r) = 0,\ \ldots,\ f^{(m-1)}(r) = 0,\ f^{(m)}(r) \neq 0$$

*If $m = 1$, we call r a **simple root***

Typically, finding these roots is difficult. At the very least, if f is nonlinear, then there rarely exists a closed form solution for r. When there are multiple roots or when roots are not simple (have multiplicity greater than 1), then even more challenges arise. Like all computational math, computers further have the challenge of conditioning.

Proposition 1.2 (Conditioning of root finding)

*Let $f : \mathbb{R} \to \mathbb{R}$ be differentiable at a root r. The **condition number** of the root finding problem is*

$$\kappa_r = |f'(r)|^{-1}$$

If $f'(r) = 0$ (i.e. r is not a simple root), then $\kappa_r = \infty$.

Proof. Recall that the condition number is the change of the output relative to the change in input with respect to some small perturbation. We need to be careful what we are perturbing here. Recall the input here is a *function* and the output is a *root*. That is, suppose we perturb the function by some small $\epsilon > 0$, so

$$f(x) \mapsto \tilde{f}(x) = x + \epsilon.$$

Assuming a root of $\tilde{f}$ exists, call it $\tilde{r} = r + \delta$. By definition $\tilde{f}(\tilde{r}) = 0$. Furthermore, since the function f is assumed to be differentiable,

$$\begin{aligned} 0 = \tilde{f}(\tilde{r}) &= \tilde{f}(r + \delta) \\ &= f(r + \delta) + \varepsilon \\ &= f(r) + \delta f'(r) + \varepsilon + o(\varepsilon^2) \\ &= \delta f'(r) + \varepsilon + o(\varepsilon^2) \end{aligned} \tag{1.1}$$

where the first equality considers that $\tilde{r}$ is a root of $\tilde{f}$, the second equality considers the perturbation on r, the next equality considers the perturbation on f, the penultimate equality applies Taylor's theorem, and the last equality considers that r is a root of f.

If we recall the absolute condition number is

$$\kappa_r = \lim_{\varepsilon \to 0} \frac{|f - \tilde{f}|}{|r - \tilde{r}|} = \lim_{\varepsilon \to 0} \left| \frac{\delta}{\varepsilon} \right|$$

where we note that δ is implicitly defined by ε. In fact, we can use the formula (1.1) to deduce that

$$\kappa_r = \lim_{\varepsilon \to 0} \left| \frac{\varepsilon - o(\varepsilon^2)}{f'(r)\varepsilon} \right| = \left| \frac{1}{f'(r)} \right|$$

where we note that by definition of little-o notation $\lim_{\varepsilon \to 0} o(\varepsilon^2)/\varepsilon = 0$. □

Note that this result is similar to something we have seen before.

Example 1.1

Suppose $f(x) = (x-1)(x-a)$ for some $a \in \mathbb{R}$. We have already seen from chapter 1 that the condition number (on the coefficients of the quadratic) for finding roots via the quadratic formula was closely tied to $|r_1 - r_2|$. Considering a general problem of root finding, we note that 1 is always a root, and $|f'(1)| = |1-a|$. Therefore, the conditioning number is

$$\kappa_r = \frac{1}{|1-a|}$$

which blows up at $a \to 1$, or as $f(x) \to (x-1)^2$. This is precisely when f goes from having only simple roots to having repeated ones.

Finally, we give examples of applications that may require some form of root finding.

- **Null states of harmonic oscillators:** If f is periodic with period m i.e. $f(x) = f(x+m)$ then finding a single root gives rise to infinity many
- **Grounding of planes, ships, projectiles, etc:** knowing a trajectory and finding when its position or height is zero.
- **Eigenvalue problems:** Anything from solving

  $$Av = \lambda v \iff (A - \lambda I)v = 0$$

  to something more complex like solving Schrödinger's equation

  $$i\hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2 \psi}{\partial x^2} + V\psi$$

  requires finding a particular vector or function that when operated on, evaluates to zero.
- **Optimization** that (for suitably structures functions)

  $$x_\star \text{is a minimizer} \iff \nabla f(x_\star) = 0$$

  so we look for roots of the gradient or derivative, but the idea is the same.

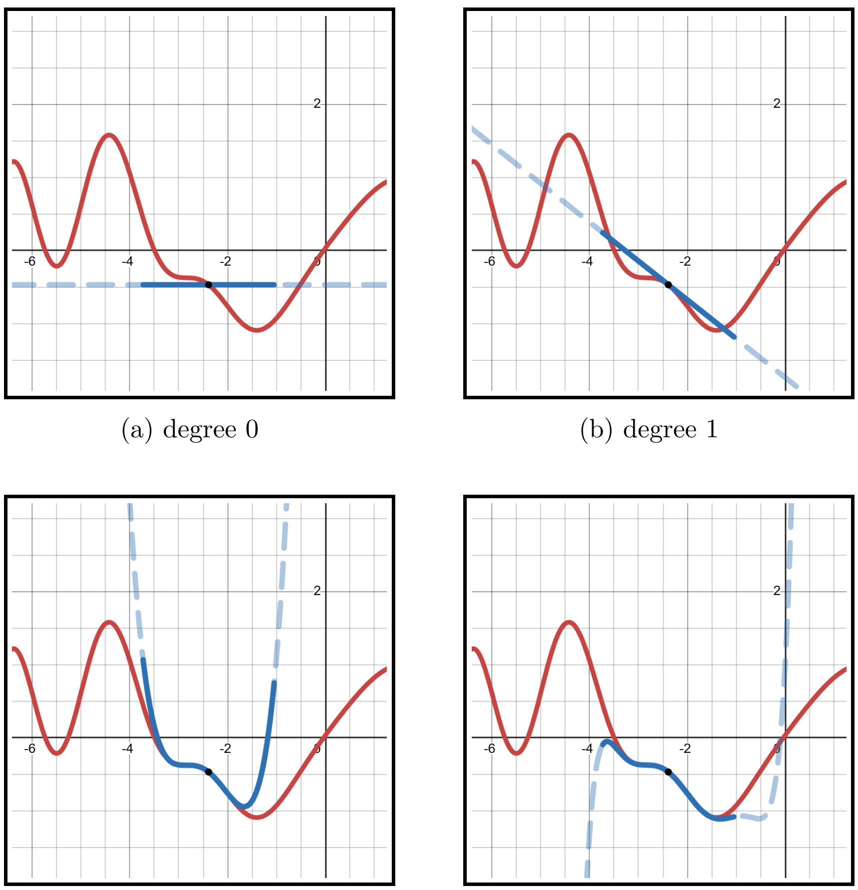
2 Taylor's Theorem

One tool we already used for deriving the condition number of the root finding problem is Taylor's Theorem. This tool will be used again and again throughout this course and many others. It is the most (and for some, the only) useful result to come from Calc II. There are higher dimensional generalizations that utilizes generalize versions of first, second, and so on derivatives. We will just focus on the one-dimensional case as it is easiest to visualize.

The story goes like this. Suppose you have a function $f(x)$ and you want to approximate it with a k-degree polynomial at a point a. Notice that a k-degree polynomial admits k derivatives before it becomes constant. We wish for our approximate $P_k(x)$ to match $f(x)$ at all these derivatives.

Definition 2.1 (Taylor Polynomial)
*Let $f(x) : \mathbb{R} \to \mathbb{R}$ be at least k-times differentiable. We say $P_k(x)$ is a **k-degree Taylor polynomial approximation at** a if*

$$f(a) = P_k(a),\ f'(a) = P_k'(a),\ \ldots,\ f^{(k)}(a) = P_k^{(k)}(a)$$



(a) degree 0
(b) degree 1
(c) degree 4
(d) degree 7

Figure 2: k-degree Taylor Polynomials

For a 0-degree Taylor approximation, we simply take the constant function $P_0(x) = f(a)$. It is well known that the 1-degree approximation, the "linear approximation," is $P_1(x) = f(a) + f'(a)(x - a)$. This is simply a rewrite of the classic "point-slope" form. We can derive an expression for any Taylor approximation constructively.

Theorem 2.2

The k-degree Taylor polynomial approximation at a for the function $f(x)$ is

$$P_k(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + ... + \frac{f^{(k)}(a)}{k!}(x-a)^k = \sum_{n=0}^{k} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

Proof. We can show this completely constructive. We know that $P_k(x)$ is a k-degree polynomial, so we aim to just find the coefficients. Furthermore, we can "center" the polynomial at any point, and the only adjustment would be to the constant addition at the end. That is, without loss of generality, we have

$$P_k(x) = c_0 + c_1(x-a) + c_2(x-a)^2 + ... + c_k(x-a)^k$$

where if a above were replaced with any other number, only c_0 would be changed. We can then solve for each c_j by considering the requirements

$$P_k^{(m)}(a) = f^{(m)}(a), \quad m = 0, 1, ..., k$$

For $m = 0$, $P_k(a) = c_0$ as all the terms with $(x-a)^m$ go away. This forces $c_0 = f(a)$. The process is similar for any other m as well. It is not hard to show that

$$P_k^{(m)}(a) = m!c_m = f^{(m)}(a)$$

by the requirements of the approximation. The powers of $(x-a)^n$ with $n < m$ go away by taking the m^{th} derivative. If $n > m$, then terms of $(x-a)^{n-m}$ remain, and plugging in $x = a$ vanishes them. □

We can note from the visual above that a higher degree polynomial will approximate a function better and better. We may ask ourselves two questions. How good is this approximation? Where do k-degree polynomials converge as $k \to \infty$? Note that the first question can be answered as an immediate corollary of the theorem.

Corollary 2.3

Let $f : \mathbb{R} \to \mathbb{R}$ be a function that is $k+1$ times differentiable on an interval I. For any $a \in I$ it holds that

$$f(x) = f(a) + f'(a)(x-a) + ... \frac{f^{(k)}(a)}{k!}(x-a)^k + R_k(x)$$

where

$$|R_k(x)| \leq \frac{\max_{z \in I} |f^{(k+1)}(z)|}{(k+1)!} |(x-a)|^{k+1}$$

The residual between the function and approximation is then bounded by two factors. First, the distance away from the center and second, the size of the subsequent derivative nearby. If the derivatives start to blow up, it may be that no approximation gets close to convergence. If we consider the extreme case at $k \to \infty$ we no longer have a polynomial, but we may have that we get convergence on some interval.

Definition 2.4

The power series of a function on an interval I, centered at $a \in I$, is defined by a sequence $\{c_0, c_1, ...\}$ such that

$$f(x) = \sum_{n=0}^{\infty} c_n (x-a)^n, \quad \forall x \in I.$$

For a given $a \in \mathbb{R}$, we define the radius of convergence to be the largest distance from a in which the above series converges.

Finding the radius of convergence is then relatively simple if we consider all the insight above. We should expect each $c_n = f^{(n)}(a)$ and to converge, we certainly need c_n to converge as a sequence. In fact, by considering $(x-a) = r$, it must hold that

$$\sum_{n=0}^{\infty} c_n (x-a)^n \text{ converges } \iff \sum_{n=0}^{\infty} c_n r^n \text{ converges.}$$

Therefore, we can do any series convergence test, like the ratio test, to determine r. That is, we need for the ration of the terms in the series to converge. If $a_n = c_n r^n$ then we need

$$\text{The series converges} \iff \lim_{n\to\infty} \left| \frac{a_{n+1}}{a_n} \right| < 1 \iff \lim_{n\to\infty} \left| \frac{c_{n+1} r}{c_n} \right| < 1$$

Solving for r we get the following lemma.

Lemma 2.5

Suppose a function admits a power series $f(x) = \sum_{n=0}^{\infty} c_n (x-a)^n$ then the radius of convergence is

$$r = \lim_{n\to\infty} \left| \frac{c_n}{c_{n+1}} \right|$$

3 Fixed Points Iteration:

One method to find a root is to use fixed point iteration.

Definition 3.1

For a function $g : \mathbb{R} \to \mathbb{R}$, a value $p \in \mathbb{R}$ is called a ***fixed point of f*** *if*

$$g(p) = p$$

Root finding is very closely related to finding a fixed point. If f is a function we wish to find a root of, then $g(x) = f(x) - x$ has the property that

$$f(r) = 0 \iff g(r) = r$$

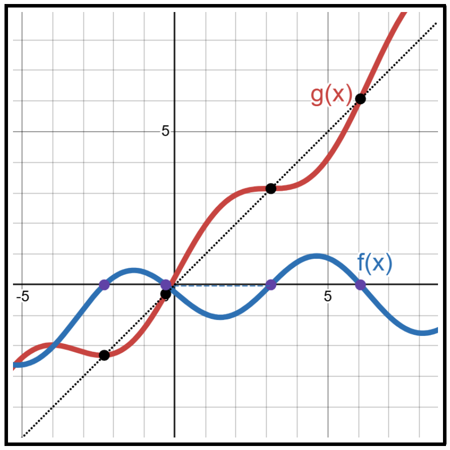


Figure 3: Roots of $f(x)$ are fixed points of $g(x)$

With the goal of finding a fixed point in mind, we can define the fixed point iteration. Like many iterative algorithms, we start with an *initial guess* and move from there.

Algorithm 3.1 (Fixed Point Iteration)

Given a continuous function g and initial guess x_0, define

$$x_{k+1} = g(x_k), \quad k = 0, 1, 2, \ldots$$

If the sequence x_k converges, then it convergence to a fixed point of g.

The idea behind a fixed point iteration is to take the next "x value" to be the current "y-value." This looks exactly like taking the current point on the graph, projecting it only the $y = x$ line, and then dropping it back on the graph and repeating. We get this "staircase" pattern emerging. Note that we can directly conclude convergence by looking at the error term-by-term. Suppose $\varepsilon_k = x_k - p$ where $p = g(p)$. That is, p is a fixed point. Then, by considering a Taylor approximation

$$\varepsilon_{k+1} + p = x_{k+1} = g(x_k) = g(\varepsilon_k + p) = g(p) + g'(p)\varepsilon_k + o(\varepsilon_k^2)$$

Since $g(p) = 0$ we get that for small enough $\varepsilon_{k+1} \approx g'(p)\varepsilon_k$

Remark 1

If p is a fixed point of g. Let x_0 be an initial point for a fixed point iteration method. If the initial error is small enough and $g'(p) < 1$, then the method converges.

Furthermore, we say the convergence is ***linear*** as we have

$$\lim_{k\to\infty} \left| \frac{\varepsilon_{k+1}}{\varepsilon_k} \right| = \sigma < 1.$$

Remark 2

Note that linear convergence looks like exponential decay when plotted. That is $|\varepsilon_k| \approx C\sigma^k$. We typically call this linear convergence for two reasons. First, ***the next iterate is a linear***

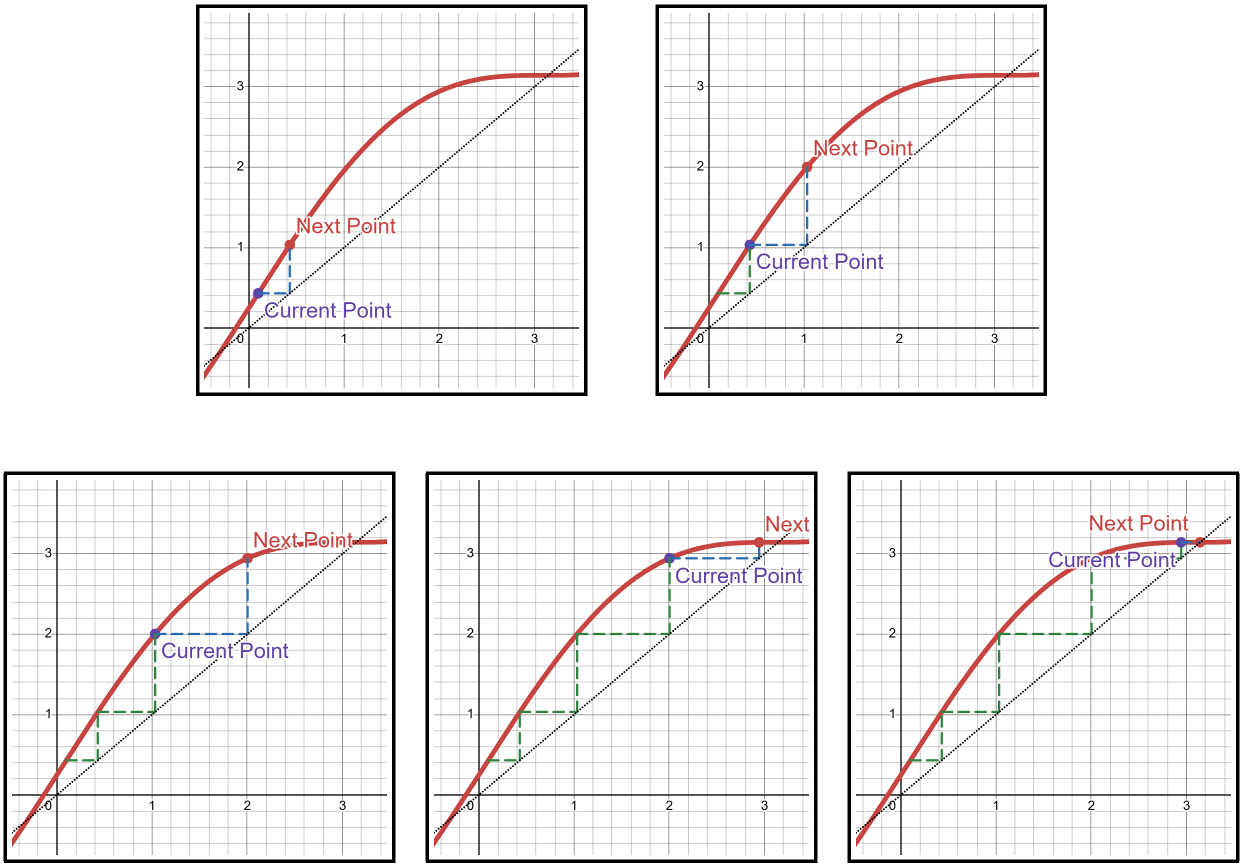


Figure 4: 5 steps of Fixed Point Iteration, $x_0 = 0.1$

function of the current one. *Second, when plotted on a log plot, the actually now looks like a line because renaming*

$$y = \log(\varepsilon_k) = \log(a\sigma^k) = \log(C) + k\log(\sigma) = \alpha k + \beta$$

Suppose we want to see this in action. We can explicitly compute the rate of convergence by looking at the residuals and see if it matches our theorem.

Example 3.1

Let $g(x) = \sin(x) + \frac{5}{4}x - \frac{\pi}{4}$ *with* $x_0 = 2$. *It is not hard to see that* $g(\pi) = \pi$ *and* $g'(\pi) = 0.25 < 1$, *so* π *is a fixed point. Let's hope that 2 is close enough to* π *for convergence. Doing the explicit calculations gives*

$$\begin{aligned} x_1 = g(x_0) &= \sin(2) + \frac{5}{4}(2) - \frac{\pi}{4} \approx 2.6239 \\ x_2 = g(x_1) &= \sin(x_1) + \frac{5}{4}(x_1) - \frac{\pi}{4} \approx 2.9894 \\ x_3 = g(x_2) &= \sin(x_2) + \frac{5}{4}(x_2) - \frac{\pi}{4} \approx 3.1029 \\ &\vdots \end{aligned}$$

We can then look at all the residuals to get the sequence

$$(1.1416, 0.5177, 0.1522, 0.03864, 0.009671, 0.002418, 0.000605, 0.000151, 0.0000377...)$$

If we took all the ratios of $|\varepsilon_{k+1}|/|\varepsilon_k|$, *we would see this series approaches* $0.25 = g'(\pi)$ *rather quickly as expected.*

4 Newtons Method:

While fixed point iteration has its uses, as far as root finding goes, there is a much more common and powerful method. Root finding has linear convergence, but requires a (rather tight) bound on the first derivative, It would be nice if a less restrictive method exists and convergence faster. Newton's method will satisfy both. The idea is simple: **take a linear approximation, solve for that root, repeat.**

Algorithm 4.1 (Newton's Method)

Given a differentiable $f : \mathbb{R} \to \mathbb{R}$ *and initial value* $x_0 \in \mathbb{R}$, *define*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, ...$$

Recall that applying Taylor's theorem, we have at any point $x_k \in \mathbb{R}$

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k)$$

where the approximation is better the closer x is to x_k. We can then easily solve for when the right hand side is 0. We set this value to be x_{k+1}. That is,

$$f(x_k) + f'(x_k)(x - x_k) = 0 \iff x = x_k - \frac{f(x_k)}{f'(x_k)}$$

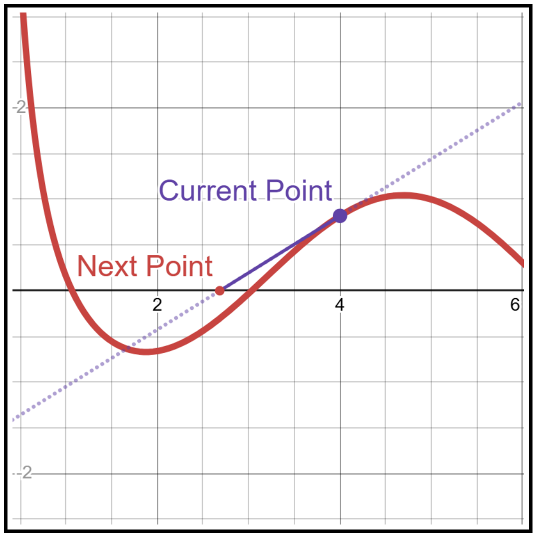


Figure 5: Newton Step

Note that both the convergence and its rate are very nontrivial at this point. In fact, it can be rather hard to find initial point that are guaranteed to converge for a generic function. Furthermore, it can be rather tricky to even characterize when convergence is guaranteed. Typically, we can guarantee convergence to a root r if $f'(r) \neq 0$, but that is unfortunately not always the case. Some exact criteria are shown below.

Theorem 4.1 (Convergence of Newton's Method)
Let $f : \mathbb{R} \to \mathbb{R}$ is a function with $x_0 \in [a, b]$ as an initial point and $r \in [a, b]$ is a root: $f(r) = 0$. Suppose the following hold

1. $f(x_1)f(x_2) < 0$ *for all* $x_1 \in [a, r)$, $x_2 \in (r, b]$. *That is, the function takes opposite signs on opposite sides of the root*
2. $f'(r) \neq 0$
3. f'' *does not change signs on* $[a, b]$
4. $f(x)f''(x) \geq 0$

then $x_k \to r$ *where* $x_{k+1} = x_k - f(x_k)/f'(x_k)$

First note is that when this convergence is guaranteed, it happens very quickly. In fact, we say newtons method converges quadratically because we can show that the residual $\varepsilon_k = x_k - r$ behaves such that

$$|\varepsilon_{k+1}| \leq C|\varepsilon_k|^2, \quad C > 0$$

This is called quadratic convergence because the subsequent error is related quadratically to the current one. The proof is rather complex, but it truly relies on nothing other than Taylor's theorem and some algebra. The implication of quadratic converge is that the iterates nearly double their precision each step. With linear convergence, we say near one digit was gained each step. Note that this is just a heuristic, and really the coefficients (i.e. C above, and $g'(p)$ for fixed point) have a lot of effect on exactly how many digits are gained.

Example 4.1

Let $f(x) = (x-3)(x-1)$. Let $x_0 = 4$. Note that $f'(x) = 2x - 4$. We can then explicitly compute the first few iterates of Newton's method.

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 4 - \frac{3}{4} = 3.25$$
$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 3.25 - \frac{9}{40} = 3.025$$
$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 3.025 - \frac{81}{3280} = 3.00030487805$$
$$x_4 = x_3 - \frac{f(x_3)}{f'(x_3)} = \ldots = 3.00000004646$$

We very quickly converge, nearly doubling the number of digits each time. Furthermore,

$$\varepsilon_{k+1} \approx \frac{1}{2}\frac{f''(r)}{f'(r)}\varepsilon_k^2 \iff \frac{\varepsilon_{k+1}}{\varepsilon_k^2} \approx \frac{1}{2}\frac{f''(r)}{f'(r)}$$

which we see by direct calculation.

Note, there are many generalizations to Newton's method, one of them using Normal equations to solve the linear approximation in the higher-dimensional setting. Really, one "solves" the same problem of taking a linear approximation and looking for a point that gives the closest evaluation to zero.

Definition 4.2

Given $f : \mathbb{R}^n \to \mathbb{R}^m$ and initial value $x_0 \in \mathbb{R}^n$ define

$$x_{k+1} = x_k + \operatorname*{argmin}_x \|A_k x - b_k\|_2^2, \quad k = 0, 1, \ldots$$

with $A_k = J_f(x_k)$ and $b_k = -f(x_k)$.

Rearranging the above definition gives

$$0 \approx f(x_k) + J_f(x_k)(x_{k+1} - x_k) \approx f(x_{k+1}).$$

Example 4.2

Suppose we want to find a solution to $\begin{cases} x^2 + y = 1 \\ 10x + y^3 = -7 \\ x - y = 5 \end{cases}$ *with initial point* $x_0 = (0, 0)$*. We first rewrite the system of equations as a single multivariate function to find a zero*

$$f(x,y) = \begin{bmatrix} x^2 + y - 1 \\ 10x + y^3 + 7 \\ x - y - 5 \end{bmatrix}, \quad J_f(x,y) = \begin{bmatrix} 2x & 1 \\ 10 & 3y^2 \\ 1 & -1 \end{bmatrix}$$

where $f(x_k, y_k) = 0 \iff (x_k, y_k)$ *solves the system. Then by taking the linear approximation of* $f(x, y)$ *centered at* (x_k, y_k) *we get*

$$f(x,y) \approx f(x_k, y_k) + J_f(x_k, y_k)(x - x_k).$$

"Solving" this for a root reduces to finding an x *such that this is as close to zero as possible. In other words, finding*

$$\underbrace{J_f(x_k, y_k)}_{A_k}(x - x_k) \approx \underbrace{-f(x_k, y_k)}_{b_k}.$$

With A_k *and* b_k *taking the roles above, we use the normal equations to solve for* $x - x_k = (A_k^T A_k)^{-1}(A_k^T b_k)$ *(this is directly from the optimality conditions used in least squares, see previous notes). Therefore, we get that*

$$A_k^T A_k = J_f(x,y)^T J_f(x,y) = \begin{bmatrix} 4x^2 + 101 & 2x + 30y^2 - 1 \\ 2x + 30y^2 - 1 & 9y^4 + 2 \end{bmatrix}$$

Call the $\det_k := \det(A_k^T A_k) = (4x^2 + 101)(9y^4 + 2) - (2x + 30y^2 - 1)^2$*. We can see that the expressions get messy quickly, but for completion here is the full result for* $(A_k^T A_k)^{-1} A_k$

$$\frac{1}{\det_k} \begin{bmatrix} 18xy^4 + 2x - 30y^2 + 1 & 20 - 6xy^2 + 3y^2 & 9y^4 + 2x + 30y^2 + 1 \\ 101 + 2x - 60xy^2 & 12x^2y^2 - 20x + 3y^2 + 10 & 4x^2 + 2x + 30y^2 - 100 \end{bmatrix}$$

Regardless of the visual complexity for us, computers (i.e. Desmos or any other program) have no problem handling these expressions. Noting that

$$b_k = -f(x_k, y_k) = \begin{bmatrix} -x_k^2 + y_k + 1 \\ -10x_k - y_k^3 - 7 \\ -x_k + y_k + 5 \end{bmatrix}$$

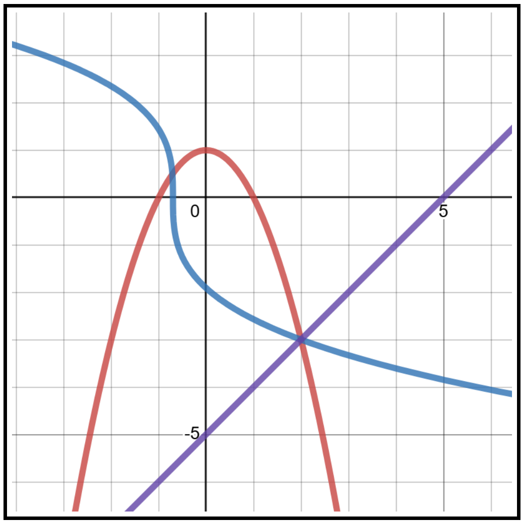
we can fully write out the iterates. With $(x_0, y_0) = (0, 0)$ *then*

$$(A_0^T A_0)^{-1} A_0 = \frac{1}{201} \begin{bmatrix} 1 & 20 & 1 \\ 101 & 10 & -100 \end{bmatrix}, \quad b_0 = \begin{bmatrix} 1 \\ -7 \\ 5 \end{bmatrix}$$

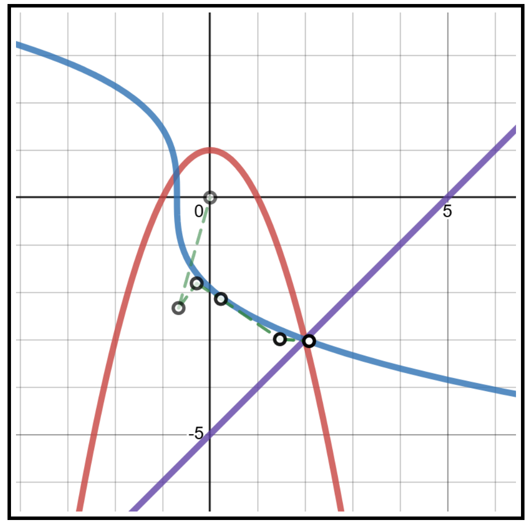
Therefore,

$$x_1 = x_0 + (A_0^T A_0)^{-1} A_0 b_0 = \begin{bmatrix} \frac{-134}{201} \\ \frac{-469}{201} \end{bmatrix} = \begin{bmatrix} -2/3 \\ -7/3 \end{bmatrix}$$

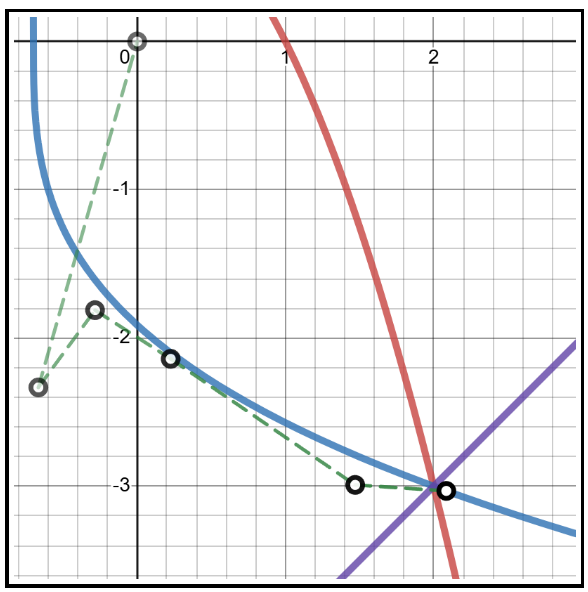
Continuing onward gives the following plot. Click here for an interactive graph.



Intersection of 3 Graphs



Iterations of Newtons



Zoomed in Iterates N = 5