Summary so far

- Root finding versus Fixed Point finding:

\[ f(x) = 0 \iff x = g(x) \]

- We denote the solution by \( \alpha \).

- The corresponding fixed point iteration is

\[ x_0 \text{ given, } x_{n+1} = g(x_n) \quad (1) \]

- The iteration (1) converges \textit{linearly} (or of first order) if

\[ 0 < |g'(\alpha)| < 1. \]

- It converges of order \( p > 1 \) if

\[ g(\alpha) = \alpha, \quad g'(\alpha) = g''(\alpha) = g^{(p-1)}(\alpha) = 0, \quad g^{(p)}(\alpha) \neq 0. \]

- Convergence will occur if we start sufficiently close to \( \alpha \).

- Stop if

\[ \frac{1}{1 - L} |x_n - x_{n+1}| < \epsilon \]

where

\[ L = \begin{cases} |x_{n+1} - x_n| & \text{if } p = 1 \\ 0 & \text{if } p > 1. \end{cases} \]
• omit arguments for sake of simpler notation.

• For Newton’s Method:

\[
g = x - \frac{f}{f'}
\]
\[
g' = 1 - \frac{f'^{2} - f f''}{f'^{2}} = \frac{f f''}{f'^{2}}
\]
\[
g'' = \frac{(f' f'' + f f''') f'^{2} - 2 f' f'' f f''}{f'^{4}}
\]
\[
= \frac{f'' f'^{2} + f f'''' f' - 2 f f''}{f'^{3}}
\]
and
\[
g''(\alpha) = \frac{f''}{f'} \hspace{1cm} (\alpha)
\]

• We get the following cases:

1. \( f'(\alpha) \neq 0, f''(\alpha) \neq 0 \). This is the standard case. NM converges quadratically.

2. \( f'(\alpha) \neq 0, f''(\alpha) = 0 \). NM converges of order at least 3. This is unlikely but it has a compelling geometric interpretation: NM is based on a first order Taylor Series. If \( f''(\alpha) = 0 \) then the quadratic term in the Taylor series is zero, and the linear Taylor approximation actually equals the quadratic Taylor approximation.

3. \( f'(\alpha) = 0 \). This is a special case of having a multiple root.
• $\alpha$ is a root of multiplicity $p$ if

$$f(\alpha) = f'(\alpha) = \ldots = f^{(p-1)}(\alpha) = 0, \quad f^{(p)}(\alpha) \neq 0.$$ 

• Numerically, having a multiple root is similar to having several roots that are close together compared with the distance from the starting point.
Example: \( f(x) = x^2 \)

Figure 1. \( f(x) = x^2 \).

It seems to work.
Suppose
\[ f'(\alpha) = 0 \quad \text{and} \quad f''(\alpha) \neq 0 \]

What happens to NM?

We get
\[
\begin{align*}
g(x) &= x - \frac{ff'}{f'} \\
g'(x) &= \frac{ff''}{f'^2} \quad \longrightarrow \quad 0 \quad 0
\end{align*}
\]

We need to use the Rule of L’Hopital:
\[
\frac{ff''}{f'^2} \longrightarrow \frac{ff''' + f'f''}{2f'f''} \\
\text{(again)} \longrightarrow \frac{ff^{iv} + f'f''' + f''^2 + f'f'''}{2(f''^2 + f'f'''})
\]

Evaluating at \( \alpha \) gives
\[
g'(\alpha) = \frac{f''^2(\alpha)}{2f''^2(\alpha)} = \frac{1}{2}.
\]

NM converges linearly with \( g'(\alpha) = \frac{1}{2} \).
• Check with \( f(x) = x^2 \):

\[ g(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2}{2x} = x - \frac{x}{2} = \frac{x}{2} \]

Exercise: The modified Newton’s Method

\[ x_{n+1} = x_n - 2 \frac{f(x_n)}{f'(x_n)} \]

converges of order 2.
• What about roots of multiplicity greater than 2?
• We could use the Rule of L’Hôpital again.
• This gives rise to a mess (try it!)
• Here is a better idea:
• Write

\[ f(x) = (x - \alpha)^p h(x) \]

where

\[ h(\alpha) \neq 0 \]

• Of course,

\[ h(x) = \frac{f(x)}{(x - \alpha)^p} \]

• NM turns into

\[
g(x) = x - \frac{f(x)}{f'(x)} \\
= x - \frac{(x - \alpha)^p h(x)}{(x - \alpha)^p h'(x) + p(x - \alpha)^{p-1} h(x)} \\
= x - \frac{(x - \alpha) h(x)}{(x - \alpha) h'(x) + ph(x)}
\]

where

\[
g'(x) = 1 - \frac{(h + (x - \alpha)h')(h + \alpha h' + ph) - ((x - \alpha)h \times \text{denominator}')}{{((x - \alpha)h' + ph)^2}}
\]
\[ g(x) = x - p \frac{f(x)}{f'(x)} \]

and

\[ g'(\alpha) = 1 - \frac{1}{p} = \frac{p - 1}{p} \]

- The iteration converges linearly, but \( g'(\alpha) \) goes to 1 as \( p \) goes to infinity.

- What if we don’t know \( p \) but \( p < \infty \)?

- A root of multiplicity \( p \) of \( f \) is a root of multiplicity 1 of

\[ \phi = \frac{f}{f'} \]

- Run NM of \( \Phi \).

\[ \text{Ex.:} \quad f(x) = e^{-1/x^2} \]
Inverse Interpolation

• Here is a nifty trick by which you can build iterations of arbitrarily high order of convergence.

• Suppose $F$ is the inverse of $f$.

\[ F(f(x)) = x. \]

• Then

\[ \alpha = F(0) \]

• Idea: expand $F$ into a Taylor series about $f(x_n) = y_n$.

• Evaluate the truncated Taylor series at $y = 0$.

\[ F(y) = F(y_n) + F'(y_n)(y - y_n) + \frac{1}{2} F''(y_n)(y - y_n)^2 + \ldots \]

\[ F(0) = F(y_n) - F'(y_n)y_n + \frac{1}{2} F''(y_n)y_n^2 + \ldots \]

• Of course, $F'(y_n) = x_n$.

• What’s $F'(y)$?

We have

\[ F(f(x)) = \quad \implies F'(f(x))f'(x) = 1 \]

\[ \implies F'(f(x)) = \frac{1}{f'(x)} \]
Evaluating the Taylor Series for $F(y)$ and truncating after the linear term gives

$$x_{n+1} = F(0) = x_n - \frac{f(x_n)}{f'(x_n)},$$

i.e., Newton’s Method again.

- Exercise: Work out the second derivative of $F$, truncate after the quadratic term in the Taylor Series, and obtain a third order method.

$$F'\left(f(x)\right) = \frac{1}{f'(x)}$$

$$F''(f(x))f'(x) = \frac{-f''(x)}{f'(x)^2}$$

$$F''(f(x)) = \frac{-f''(x)}{f'(x)^3}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{1}{2} \frac{f''(x_n)}{f'(x_n)^3} f''(x_n)$$
Newton’s Method for Systems

• Suppose $F$ is function from $\mathbb{R}^n$ to $\mathbb{R}^n$:

$$F : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

and we want to solve

$$F(x) = 0 \quad (2)$$

Here

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

and

$$F(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_n(x) \end{bmatrix}$$

• We are going to construct a sequence of vectors

$$x^{(0)}, x^{(1)}, x^{(2)}, \ldots$$

that hopefully converges to the solution of (2). (We use superscripts in parentheses because subscripts would refer to components of a vector, and plain superscripts would denote exponents.)
• As in the case of one variable, Newton’s Method for (2) can be obtained by linearizing $F$ and solving the linear problem.

$$F(x) = F(x^{(0)}) + \nabla F(x^{(0)}) (x - x^{(0)}) + \text{HOT}$$

where HOT means “higher order terms” (which we ignore) and

$$\nabla F = \left[ \frac{\partial F}{\partial x_j} \right]_{i,j=1,2,...n}$$

is the **Jacobian** of $F$.

• Solving

$$F(x^{(0)}) + \nabla F(x^{(0)}) (x - x^{(0)}) = 0$$

for $x$ gives

$$x = x^{(1)} = x^{(0)} - \left( \nabla F(x^{(0)}) \right)^{-1} F(x^{(0)})$$

• This gives rise to Newton’s Method:

$$x^{(0)} \text{ given } \quad x^{(k+1)} = x^{(k)} - \left( \nabla F(x^{(k)}) \right)^{-1} F(x^{(k)})$$

where $k = 0, 1, 2, \ldots$

• Note that in the case that $n = 1$ this reduces to the ordinary scalar Newton’s method.
• As a practical matter, we never invert a matrix (more on that later in the semester).

• Instead we can implement Newton’s Method as follows:

\[ x^{(0)} \text{ given} \]

• For \( k = 0, 1, 2, \ldots \) do:
  1. Compute \( F(x^{(k)}) \) and \( A = \nabla F(x^{(k)}) \)
  2. Solve \( As = -F(x^{(k)}) \)
     (The vector \( s \) is called the Newton Step).
  3. Let \( x^{(k+1)} = x^{(k)} + s \)
  4. Repeat until \( \|s\| < 0.01 \)
     (We stop when the length of the Newton step is less than 1 cm.)