1 Approximate methods and assumptions

The final topic covered in this class is iterative methods for optimization. These are meant to help us find approximate solutions to problems in cases where finding exact solutions would be too hard.

There are two things we’ve taken for granted before which might actually be too hard to do exactly:

1. Evaluating derivatives of a function $f$ (e.g., $\nabla f$ or $Hf$) at a given point.
2. Solving an equation or system of equations.

Before, we’ve assumed (1) and (2) are both easy. The first few methods we cover assume that (1) is still easy, but (2) might be hard. For instance, this happens when dealing with polynomial functions: taking derivatives in that case is straightforward, but solving a high-degree polynomial equation even in one variable is impossible to do exactly.

2 The classical Newton’s method

Eventually we’ll get to optimization problems. But we’ll begin with Newton’s method in its basic form: an algorithm for approximately finding zeroes of a function $f : \mathbb{R} \to \mathbb{R}$. This is an iterative algorithm: starting with an initial guess $x_0$, it makes a better guess $x_1$, then uses it to make an even better guess $x_2$, and so on. We hope that eventually these approach a solution.

From a point $x_k$, Newton’s method does the following:

1. Compute $f(x_k)$ and $f'(x_k)$.
2. Approximate $f(x)$ by the linear function $f(x_k) + (x - x_k)f'(x_k)$.
3. Let $x_{k+1}$ be the point where the linear approximation is zero:

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

If we start with a good enough $x_0$, and compute $x_1, x_2, x_3, \ldots$, then with some luck, as $k \to \infty$, $x_k \to x^*$, which satisfies $f(x^*) = 0$.

---

1 This document comes from the Math 484 course webpage: https://faculty.math.illinois.edu/~mlavrov/courses/484-fall-2018.html
2.1 Examples

When Newton’s method works, it usually works very well. For example, if we try to solve $4x^3 + x - 1 = 0$ with an initial guess of $x_0 = 1$, we’ll get:

$$
\begin{align*}
x_0 &= 1.00000 00000 \\
x_1 &= 0.69230 76923 \\
x_2 &= 0.54129 30628 \\
x_3 &= 0.50239 01750 \\
x_4 &= 0.50000 85354 \\
x_5 &= 0.50000 00001
\end{align*}
$$

and it’s easy to see that we are getting closer and closer to $x^* = 0.5$ which really is a solution of the equation.

This example is an ideal case, where the convergence is quadratic: once $x_k$ is sufficiently close to $x^*$, $|x^* - x_{k+1}| \leq C|x^* - x_k|^2$. This means that the number of correct digits roughly doubles at each step. You should think of this type of convergence as “fast” convergence.

Occasionally, we get worse behavior. For example, if we try to solve $4x^3 - 3x + 1 = 0$ with an initial guess of $x_0 = 0$, we get

$$
\begin{align*}
x_0 &= 1.00000 00000 \\
x_1 &= 0.33333 33333 \\
x_2 &= 0.42222 22222 \\
\vdots \\
x_{10} &= 0.49971 21712 \\
\vdots \\
x_{20} &= 0.49999 97190
\end{align*}
$$

which would maybe have looked impressive if you hadn’t seen the previous example first. In this example, we get linear convergence: even when $x_k$ is close to the limit $x^*$, we only have $|x^* - x_{k+1}| \leq C|x^* - x_k|$ for some $C < 1$. You should think of this type of convergence as “slow convergence”.

You might ask: can’t we also encounter convergence rates where $|x^* - x_{k+1}| \leq C|x^* - x_k|^{\alpha}$ for some $\alpha$ other than 1 or 2? There are methods that do this—for example, there are variants of Newton’s method which can sometimes achieve cubic convergence. However, all values of $\alpha > 1$ are very similar to each other, and very different from $\alpha = 1$. When $\alpha = 1$, then depending on $C$, we get a fixed number of new correct digits at each step. When $\alpha > 1$, the number of correct digits grows exponentially. The base of the exponent varies with $\alpha$, but exponential growth is very quick no matter what the base of the exponent is.

Going back to Newton’s method: it’s also possible for it to never converge at all. For example, suppose we take $f(x) = |x|^{1/3}$, for which the only root is $x^* = 0$. Starting with $x_0 = 1$, we get
$x_1 = -2, x_2 = 4, x_3 = -8, x_4 = 16$, and in general $x_k = (-2)^k$. We are getting further and further away from the desired value of $x^*$ as we go.

There are other ways in which we can fail to converge; for example, we might get into a loop where, after a few iterations, we’re back where we started. We might also end up at a point $x_k$ where $f'(x_k) = 0$, in which case Newton’s method doesn’t give us a way to proceed at all.

We get an intuitive explanation for what’s going on in these three examples once we graph all three functions. From left to right, the graphs are:

In the first case, we got fast convergence because, near the root, we have a straightforward linear approximation to the function and everything is nice.

In the second case, at the root $x^*$, we have not just $f(x^*) = 0$ but also $f'(x^*) = 0$. Points where $f'(x)$ is close to 0 are normally very bad for Newton’s method, because then the linear approximation is very nearly horizontal. So we have to fight this problem all along the way, and get slow convergence as a result.

In the third case, at the root $x^*$, the derivative $f'(x^*)$ is undefined. It’s not surprising that when the root is as poorly behaved as this one, Newton’s method has trouble converging to it.

Assuming we’re not in the third case where $f'(x^*)$ is undefined, Newton’s method is guaranteed to converge, provided we have a sufficiently good initial guess. We won’t prove this, but intuitively speaking, the local behavior of any nice function near the root is going to be more or less the same as the local behavior of one of our first two examples, so it’s going to converge because they do.

But if we start out very far from $x^*$, it takes some luck to get close enough to $x^*$ for Newton’s method to start actually being helpful.

### 2.2 Some proofs

The previous section was very handwavy; this is because it’s hard to prove guarantees, in general, that Newton’s method works.

What we can prove without too much trouble is that, assuming $f'(x)$ is continuous, if Newton’s method converges to some point $x^*$, then we will have $f(x^*) = 0$ at that point.

The argument for this is an argument that can be applied without modification to many iterative methods, so it’s very useful to know.
Suppose that the sequence of points \(x_0, x_1, x_2, x_3, \ldots\) we get from Newton’s method satisfies \(x_k \to x^*\) as \(k \to \infty\). Because \(f\) and \(f’\) are continuous, we also have:

\[
\lim_{k \to \infty} f(x_k) + (x_{k+1} - x_k)f'(x_k) = f(x^*) + (x^* - x^*)f'(x^*) = f(x^*).
\]

(We just apply the rule that \(\lim_{k \to \infty} h(x_k) = h(\lim_{k \to \infty} x_k)\) several times, which works when \(h\) is continuous.)

On the other hand, we choose \(x_{k+1}\) at each step to satisfy \(f(x_k) + (x_{k+1} - x_k)f'(x_k) = 0\). So we’re taking a limit of 0 as \(k \to \infty\), which guarantees that we get 0 at the end. Since one way of taking the limit gives 0 and another gives \(f(x^*)\), we must have \(f(x^*) = 0\).

If \(f'(x^*) \neq 0\), then we can simplify this argument a bit. Define \(h(x) = x - \frac{f(x)}{f'(x)}\): this is a continuous function of \(x\) away from points where \(f'(x) = 0\), because \(f\) and \(f’\) are continuous. Then

\[
\lim_{k \to \infty} x_k = x^* \implies \lim_{k \to \infty} h(x_k) = h(x^*)
\]

by the same argument, but \(h(x_k)\) is just \(x_{k+1}\), and the limit of \(x_{k+1}\) is the same as the limit of \(x_k\). So \(x^*\) must satisfy \(h(x^*) = x^*\), from which a little bit of algebraic manipulation gives \(f(x^*) = 0\).

This is the general rule you should keep in mind: whenever you are using an iterative method which sets \(x_{k+1} = h(x_k)\), then if it converges, it will always converge to a fixed point of \(h\).