Solving Scalar Nonlinear Equations
Atkinson Chapter 2, Stoer & Bulirsch Chapter 5

1 With linear systems we started with existence, uniqueness (both were review), and sensitivity to perturbations. For scalar nonlinear equations solutions might not exist, or might not be unique; there's no general theory. Naturally there's no general theory for sensitivity either.

Notice that any scalar equation can be written as

$$f(x) = 0$$

in which case we're 'rootfinding' or as

$$g(x) = x.$$

We will start by considering equations in the form $g(x) = x$. We can (seemingly arbitrarily) define an iteration

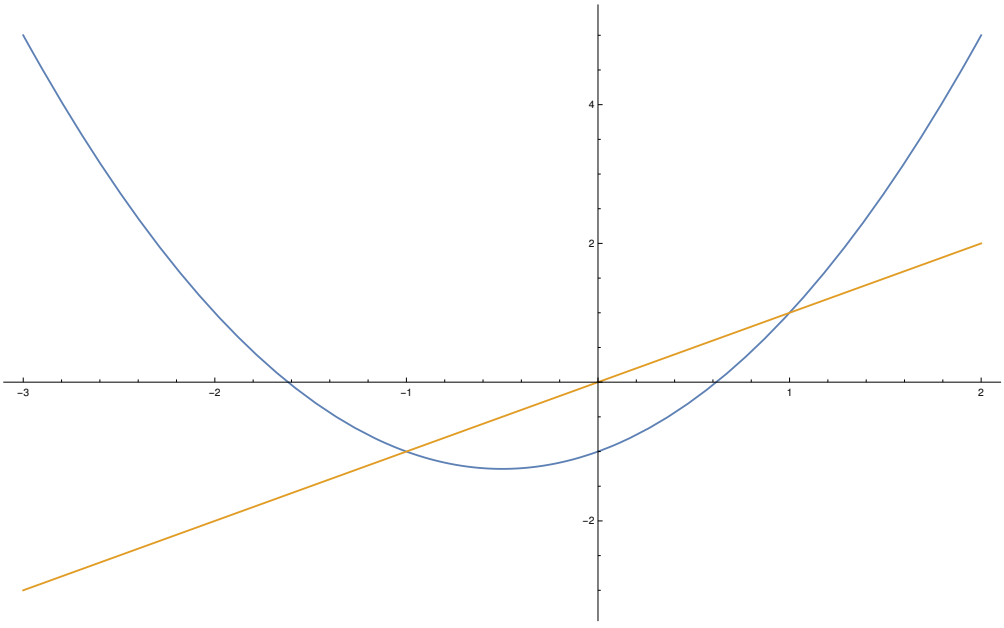
$$x_{k+1} = g(x_k).$$

If it converges then it converges to $x = g(x)$. The value of x that solves $x = g(x)$ is called a 'fixed point' and the iteration above is called a 'fixed point iteration.'

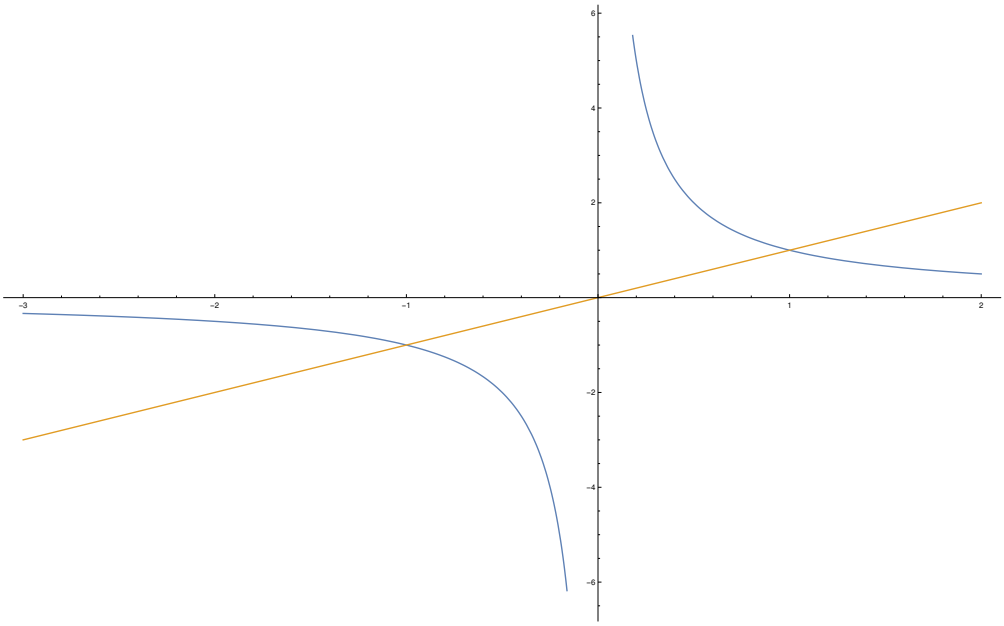
E.g. solve $0 = x^2 - c$ with $c > 0$; roots are $\alpha_\pm = \pm\sqrt{c}$. Let $g(x) = x^2 - c + x$ so that $g(x) = x$ when $0 = x^2 - c$. Note that there are lots of ways to choose a function g here.

$$x_{k+1} = g(x_k) = f(x_k) + x_k = x_k + x_k^2 - c.$$

Draw the path of x_k. Appears to converge to $-\sqrt{c}$ for some range of x_0, but looks like it won't ever converge to $\sqrt{c}$.



For this same nonlinear equation we could use $g(x) = c/x$; the solutions $g(x) = x$ are $\pm\sqrt{c}$. Draw the path of x_k. Appears to converge to both roots if you start close enough.



2 (Atkinson §2.5) How can we *rigorously* analyze the behavior of the iteration? We will find that (i) if the initial condition is 'close enough' to a solution, and (ii) if the function g is 'nice enough' near the solution, then the iteration will converge.

Close Enough First we find an interval with the property that the iteration remains within the interval for all time, i.e. an interval $[a, b]$ that $g(x)$ maps into itself: $a \le g(x) \le b$ for all $a \le x \le b$. E.g. for $g(x) = c/x$ the interval $[c, 1]$ maps to itself (or $[1, c]$ if $c > 1$). This guarantees that the iteration will remain in this interval for all k (if started in the interval). Of course the interval isn't useful to us unless it contains a solution, but we have the following result:

Lemma: (Atkinson 2.4) If $g(x)$ is continuous on $[a, b]$ and $a \le g(x) \le b$ for $x \in [a, b]$ then there is at least one solution to $g(x) = x$ in $[a, b]$.

The proof considers that $g(x) - x$ is continuous on $[a, b]$. By assumption $a \le g(a) \Rightarrow g(a) - a \ge 0$ and $g(b) \le b \Rightarrow g(b) - b \le 0$. By the IVT there must be at least one zero of $g(x) - x$ in the interval.

Nice Enough We've already assumed g is continuous on the interval in question. For convergence we need something a little stronger: Lipschitz continuous with constant less than 1:

$$\exists\, 0 < \lambda < 1 \;:\; |g(x) - g(y)| < \lambda |x - y| \quad \forall x, y \in [a, b]$$

(Lipschitz continuity is $|g(x) - g(y)| \le K|x - y|$. The condition above is slightly stronger than Lipschitz continuity.)

Lemma: (Atkinson 2.5) If g satisfies the Lipschitz condition above and $a \le g(x) \le b$ for $x \in [a, b]$ then (i) $x = g(x)$ has a unique solution α in $[a, b]$ and (ii) the iteration will converge to α for any initial condition in the interval.[1]

[1] For those who care, this is a specific example of the more general Banach fixed-point theorem. We don't prove the fully general theorem here, we just do a very specific 1D version of it.

Proof:

(i) Suppose there are two solutions $\alpha = g(\alpha)$, $\beta = g(\beta)$. Plug $x = \alpha$, $y = \beta$ into the assumption and simplify:

$$|g(\alpha) - g(\beta)| = |\alpha - \beta| < \lambda|\alpha - \beta|$$

if $\alpha \neq \beta$ then this implies $\lambda > 1$, a contradiction. (Reductio ad absurdum)

(ii) Plug in $x = \alpha$ and $y = x_k$:

$$|g(\alpha) - g(x_k)| = |\alpha - x_{k+1}| < \lambda|\alpha - x_k|.$$

By induction, $|e_k| < \lambda^k|e_0|$, so it must converge.

In summary, if g has certain properties ('nice enough') and x_0 starts 'close enough' (how close is close enough depends on the function g) then the iteration will converge to the fixed point.

3 The above lemmas give sufficient conditions for the existence of a solution and for convergence to that solution. But the conditions are not always easy to check; in particular it can be hard to find an interval that maps to itself. It turns out that we can assume something a little stronger and easier to check.

> **Theorem** (Atkinson 2.7) If $g(x)$ is continuously differentiable in a neighborhood of the fixed point α and $|g'(\alpha)| < 1$ then there is some $\epsilon > 0$ such that the iteration will converge for all $x_0 \in [\alpha - \epsilon, \alpha + \epsilon]$.

Proof: If $|g'(\alpha)| < 1$ then by continuity of g' there is some $\epsilon > 0$ such that $|g'| \leq \lambda < 1$ for all $x \in [\alpha - \epsilon, \alpha + \epsilon]$. Now suppose that we have an initial guess in this interval. Then

$$\alpha - x_1 = g(\alpha) - g(x_0) = g'(\xi_0)(\alpha - x_0)$$

which is basically Taylor's theorem, aka the mean value theorem. The number ξ_0 is between α and x_0. Taking absolute values and using $|g'| \leq \lambda$ we have

$$|\alpha - x_1| \leq \lambda|\alpha - x_0| \leq \epsilon$$

where the last inequality uses $|\alpha - x_0| < \epsilon$. So we know that if we start within the interval then we'll stay in the interval for all k. Furthermore, repeated application of the same argument as above implies

$$|\alpha - x_k| \leq \lambda^k|\alpha - x_0|.$$

The fact that $\lambda < 1$ implies that $\lim |\alpha - x_k| = 0$, i.e. the iteration converges.

If we don't know whether a solution exists at all, then we have to fall back on lemma 2.4: find an interval that maps to itself and where g is continuous. Usually you know the solution exists and you just want to know whether the iteration will converge for 'close-enough' initial conditions. Theorem 2.7 answers this question.

It's worth noting that the condition in Theorem 2.7 is not necessary-and-sufficient. In particular, if $|g'(\alpha)| = 1$ you might still be able to get convergence.

4 Rate of Convergence. Supposing again that g is continuously-differentiable and satisfies the conditions for convergence with fixed point $x = \alpha$. Taylor's formula says

$$\alpha - x_{k+1} = g(\alpha) - g(x_k) = g'(\xi_k)(\alpha - x_k)$$

for some ξ_k between α and x_k. Take absolute values, divide, and take the limit:

$$\lim_{k\to\infty} \left| \frac{\alpha - x_{k+1}}{\alpha - x_k} \right| = \lim_{k\to\infty} |g'(\xi_k)| = |g'(\alpha)|.$$

So for continuously-differentiable g, the rate of convergence is 'linear' unless $g'(\alpha) = 0$.

If $g'(\alpha) = 0$ then the iteration will converge (since $0 < 1$), but how fast? In the above analysis just use a higher-order Taylor expansion.

> **Theorem** (Atkinson 2.8) Assume that g is p times continuously differentiable at the fixed point α, and that $g'(\alpha) = \ldots = g^{(p-1)}(\alpha) = 0$ but $g^{(p)}(\alpha) \neq 0$. Then the iteration converges and there is some $\lambda \neq 0$ such that
>
> $$\lim_{k\to\infty} \frac{|\text{error}_{k+1}|}{|\text{error}_k|^p} = \lambda.$$

Comment 'Linear convergence' means that the error goes to zero exponentially fast with increasing k:

$$|\text{error}_k| \leq \lambda^k |\text{error}_0| \text{ for some } 0 < \lambda < 1.$$

'Quadratic convergence' is $p = 2$ in Theorem 2.8; it would look like

$$|\text{error}_{k+1}| \leq \lambda |\text{error}_k|^2$$

which is faster than exponential as a function of k.

5 Convergence rates & computational cost. Suppose that you have a linearly converging method; then for large k the error is

$$|e_k| \approx \lambda |e_{k-1}|$$

for some $0 < \lambda < 1$. Suppose that at step $k - 1 = 999$ the error is $|e_{k-1}| \approx 10^{-5}$; then at step $k = 1000$ the error is $|e_k| \approx \lambda \times 10^{-5}$. How many steps to increase your accuracy by one digit to $\approx 10^{-6}$? It takes $k \approx \log(10)/\log(\lambda)$ steps. For a linearly converging method in the asymptotic regime, the number of correct digits increases linearly with the number of steps.

Suppose that you have a quadratically converging method; then for large k the error is

$$|e_k| \approx \lambda |e_{k-1}|^2$$

for some $0 < \lambda$. Suppose that at step $k - 1 = 999$ the error is $|e_{k-1}| \approx 10^{-5}$; then at step $k = 1000$ the error is $|e_k| \approx \lambda \times 10^{-10}$. I.e. you approximately double the number of digits at each step. (If $\lambda = 10$ then the number of digits at step k is twice what it was at step $k-1$, minus one digit.) For a quadratically converging method in the asymptotic regime, the number of correct digits increases exponentially (e.g. as 2^k) with the number of steps.

Clearly higher-order methods converge a lot faster, but you should always consider the cost of a single step. If a single step of the higher-order method is ten times as expensive than the linear method, then the linear method might still be a better choice. There is always some number K such that the quadratic method is more accurate than the linear method for $k > K$, but there's also always a finite precision (e.g. 16 digits in double precision). The real question is how much work it takes to get to the desired precision, and sometimes a linear method could be better than a quadratic method by this metric.

6 Aitken extrapolation. (Atkinson §2.6) Assume we have a linearly-converging fixed-point iteration with continuously-differentiable g

$$\lim_{k\to\infty} x_k = \alpha, \;\; 0 < |g'(\alpha)| < 1.$$

If we can get an estimate of the error, then we could add the error to the current iterate to produce a better guess. We don't have access to the error, but we have access to $x_k - x_{k-1}$. So, consider

$$\alpha - x_k = \alpha - x_{k-1} - (x_k - x_{k-1}).$$

The error formula says

$$\alpha - x_k = g'(\xi_{k-1})(\alpha - x_{k-1}).$$

If we're close to the root then $g'(\xi_{k-1}) \approx g'(\alpha)$, so

$$\alpha - x_k \approx g'(\alpha)(\alpha - x_{k-1}).$$

Plug this in above:

$$\alpha - x_k \approx \frac{\alpha - x_k}{g'(\alpha)} - (x_k - x_{k-1}).$$

Solve for the error

$$\alpha - x_k \approx \frac{g'(\alpha)}{1 - g'(\alpha)}(x_k - x_{k-1}).$$

If you have an estimate for $g'(\alpha)$, then you can get an improved guess for the solution by adding your estimate of the error

$$\hat{x}_k = x_k + \frac{g'(\alpha)}{1 - g'(\alpha)}(x_k - x_{k-1}).$$

We can estimate $|g'(\alpha)|$ as follows

$$\lambda_k = \frac{x_k - x_{k-1}}{x_{k-1} - x_{k-2}}, k \geq 2$$

Re-arrange

$$\lambda_k = \frac{\alpha - x_{k-1} - (\alpha - x_k)}{(\alpha - x_{k-2}) - (\alpha - x_{k-1})}$$

Use

$$\alpha - x_k = g'(\xi_k)(\alpha - x_{k-1}), \;\; \alpha - x_{k-2} = \frac{\alpha - x_{k-1}}{g'(\xi_{k-2})}$$

to get an expression involving only $\alpha - x_{k-1}$

$$\lambda_k = \frac{\alpha - x_{k-1} - g'(\xi_{k-1})(\alpha - x_{k-1})}{(\alpha - x_{k-1})/g'(\xi_{k-2}) - (\alpha - x_{k-1})}$$

Now divide out $\alpha - x_{k-1}$

$$\lambda_k = \frac{1 - g'(\xi_{k-1})}{1/g'(\xi_{k-2}) - 1}$$

Now take the limit and recall our assumption that the sequence is converging

$$\lim_{k\to\infty} \lambda_k = \frac{1 - g'(\alpha)}{1/g'(\alpha) - 1} = g'(\alpha).$$

This implies

$$\lambda_k \approx g'(\alpha) \text{ for large } k.$$

7 Aitken extrapolation. Review: If you have an estimate for $g'(\alpha)$, then you can get an improved guess for the solution by adding your estimate of the error

$$\hat{x}_k = x_k + \frac{g'(\alpha)}{1 - g'(\alpha)}(x_k - x_{k-1}).$$

We can estimate $g'(\alpha)$ as follows

$$\lambda_k = \frac{x_k - x_{k-1}}{x_{k-1} - x_{k-2}}, k \geq 2$$

We showed above that

$$\lambda_k \approx g'(\alpha) \text{ for large } k.$$

Aitken's method first computes $\{x_k\}$ using the fixed-point iteration, then computes

$$\hat{x}_k = x_k + \frac{\lambda_k}{1 - \lambda_k}(x_x - x_{k-1}).$$

This is not too useful since you might as well just compute the last iteration, not the whole sequence.

Steffensen instead suggested the following iteration:

(0) Run your fixed-point method until it looks like it's starting to converge. Then set the three most recent estimates to x_2, x_1, and x_0. This is the initial condition for Aitken extrapolation.

(1) Set $x_3 = x_2 + \frac{\lambda_2}{1-\lambda_2}(x_2 - x_1)$

(2) Set replace x_0 with x_3, then compute new x_1, x_2 using the fixed point iteration.

(3) Return to (1) (until convergence)

This doesn't look like a fixed-point iteration any more, but it can be written that way. First simplify a bit:

$$\hat{x}_k = x_k - \frac{(x_k - x_{k-1})^2}{(x_k - x_{k-1}) - (x_{k-1} - x_{k-2})}. \tag{2.6.8}$$

We want to write $\hat{x}_k = x_{k+1}$. Use $x_{k-1} = g(x_{k-2})$ and $x_k = g(x_{k-1}) = g(g(x_{k-2}))$ to write the RHS above as a function of x_{k-2}:

$$\hat{x}_k = g(g(x_{k-2})) - \frac{(g(g(x_{k-2})) - g(x_{k-2}))^2}{(g(g(x_{k-2})) - g(x_{k-2})) - (g(x_{k-2}) - x_{k-2})} = \frac{xg(g(x)) - g(x)^2}{g(g(x)) - 2g(x) + x} = G(x_{k-2}).$$

So this new method is equivalent to a fixed-point iteration of the form

$$x_{j+1} = G(x_j).$$

It can be shown that if the original iteration converges with order $p > 1$ then the updated method converges with order $2p - 1$. If the original method is linear ($p = 1$), then the updated method is at least of order 2. (Stoer & Bulirsh, Theorem 5.10.13.)

8 Return to $f(x) = 0$. Suppose you know that $f(x)$ is continuous on $[a, b]$ and $f(a)f(b) < 0$. Then there must be a root somewhere in (a, b). The 'bisection' method computes $f((a + b)/2)$. If it is the same sign as $f(a)$ then there must be a root in the right half of the interval, and vice versa. Bisection iterates this process to convergence. The size of the region containing the root decreases by a factor of 2 at each step.

9 $f(x) = 0$. Newton's method produces a local linear approximation to f at x_k, then finds the root of this local linear approximation and sets that equal to x_{k+1}. The local linear approximation is

$$f(x) \approx L(x) = f(x_k) + f'(x_k)(x - x_k).$$

Setting $L(x) = 0$ and letting the solution be x_{k+1} yields

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

This is a fixed-point iteration with

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

First ask whether fixed points of g are actually roots of f:

$$g(x) = x \Rightarrow \frac{f(x)}{f'(x)} = 0 \Rightarrow f(x) = 0.$$

OK, now assume that g is continuously-differentiable at the fixed point. *This assumes that f is twice-continuously-differentiable.* Then

$$g'(x) = 1 - 1 + \frac{f(x)f''(x)}{(f'(x))^2}.$$

If $f' \neq 0$ at the fixed point then $g' = 0$ and convergence is faster than linear. To find out how fast, take another derivative (now assuming $f^{(3)}$ is continuous!):

$$g''(x) = \frac{(f'(x))^2 f''(x) + f(x)f'(x)f^{(3)}(x) - 2f(x)(f''(x))^2}{(f'(x))^3} \rightarrow \frac{f''(x)}{f'(x)}.$$

Assuming that $f'' \neq 0$ at the fixed point, convergence will be quadratic. If $f'' = 0$ then we might have faster than quadratic convergence.

10 Suppose that $f' = 0$ at the fixed point but $f'' \neq 0$. Then to evaluate g' at the fixed point we use could use L'Hopital's rule and find

$$g'(\alpha) = \frac{1}{2}.$$

So for a 'double' root, Newton's method still converges, but linearly rather than quadratically. If $f' = f'' = 0$ but $f''' \neq 0$ at the fixed point you can keep using L'Hopital's rule, but there's an easier way.

A root of order p of a smooth function f is a number α such that

$$f(x) = (x - \alpha)^p h(x), \;\; h(\alpha) \neq 0.$$

At a root of order p,

$$f^{(k)}(\alpha) = 0, \; k = 1, \ldots, p - 1.$$

Plug this into the expression for $g'(x)$ and simplify:

$$g'(x) = \frac{f(x)f''(x)}{(f'(x))^2} = h(x)\frac{p(1-p)h(x) + (x-\alpha)(2ph'(x) - (x-\alpha)h''(x))}{(ph(x) + (x-\alpha)h'(x))^2}.$$

If we evaluate at $x = \alpha$ we find

$$g'(\alpha) = 1 - \frac{1}{p}.$$

So for any finite-order multiple root ($p > 1$), and provided that f is sufficiently differentiable, Newton's method will converge linearly.

Interestingly, if you know the order of the root beforehand and modify Newton's method to

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

you recover quadratic convergence.

Or, define

$$F(x) = \frac{f(x)}{f'(x)}.$$

F has a simple root at α, so Newton's method applied to F will converge quadratically. This requires extra function evaluations, including the second derivative of f.

12 Secant method (Atkinson §2.3). If you don't have access to $f'(x)$ (e.g. because the way you evaluate $f(x)$ is to give x to some black-box software package), you can't use Newton's method. But you can use a "Quasi-Newton" method that approximates the derivative f using a difference:

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = m_k.$$

Instead of solving for the root of the linearization $L(x)$, you solve for the root of the approximate linearization (the 'secant' line)

$$S(x) = f(x_k) + m_k(x - x_k).$$

Solve for the root of $S(x)$ and set this equal to the next iterate

$$x_{k+1} = x_k - \frac{f_k}{m_k}.$$

The error analysis is somewhat involved and the analysis itself is not terribly educational (see Atkinson). We simply present the result: If the initial condition is 'close enough' to a simple root of a 'nice enough' function f, then the method converges faster than linearly but slower than quadratically.

13 We now specialize to a specific kind of scalar nonlinear equation: Finding polynomial roots. The methods described in Atkinson §2.9 are not widely used, and the widely-used method is not described in Atkinson §2.9. The standard method nowadays (e.g. in Matlab's polynomial root finder) is to find the eigenvalues of a 'companion' matrix. Recall that the eigenvalues of a matrix $\mathbf{A}$ are the roots of the characteristic polynomial $\det[\mathbf{A} - \lambda\mathbf{I}]$.

If we start with a polynomial

$$p(x) = a_0 + a_1 x + \ldots + a_n x^n$$

we can construct a 'companion' matrix whose characteristic polynomial is the proportional to the polynomial p. Then we can find the eigenvalues of the companion matrix using methods discussed in APPM 5610 (Numerics II) and in APPM 5620: Numerical Linear Algebra.

First assume wlog that $a_n \neq 0$ and notice that the roots of p are the same as the roots of

$$p(x)/a_n = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1} + x^n.$$

Now construct the matrix

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & \\ & 0 & 1 & 0 & \cdots \\ & & & & \\ & \cdots & \cdots & 0 & 1 \\ -c_0 & -c_1 & \cdots & -c_{n-2} & -c_{n-1} \end{bmatrix}.$$

Notice that if α is a root, then the vector

$$(1, \alpha, \alpha^2, \cdots, \alpha^{n-1})$$

is an eigenvector with eigenvalue α. To verify, plug the vector and value into the formula $\mathbf{A}\boldsymbol{x} = \lambda\boldsymbol{x}$. The first $n-2$ equations all have the form $\alpha^k = \alpha^k$, while the last equation has the form

$$-c_0 - c_1\alpha - \cdots - c_{n-1}\alpha^{n-1} = \alpha^n.$$

You can alternatively expand the determinant of the companion matrix along the bottom row and find that the characteristic polynomial is proportional to the original polynomial, so they have the same roots.

Of course you can also apply Newton's method if you prefer. The above analysis works if you have coefficients for the polynomial with respect to the monomial basis. If you have coordinates of the polynomial in an orthogonal basis like Chebyshev or Legendre, the above idea can be applied but using a 'colleague' or 'comrade' matrix. We might come back to that when we learn about orthogonal polynomials.

14 Although we won't really cover methods for finding polynomial roots, we can cover *sensitivity* of the solution to perturbations in the coefficients, just like for linear systems.

Lemma (S&B Theorem 6.9.8, section 5.8) Let ξ be a simple root of the polynomial $p(x)$ and let g be a polynomial of degree less than or equal to that of p. For sufficiently small values of ϵ, there is an analytic function $\xi(\epsilon)$ that is a simple root of

$$p_\epsilon(x) = p(x) + \epsilon g(x)$$

with $\xi(0) = \xi$. (No proof; requires analysis of eigenvalues of perturbed matrices; see e.g. S&B §5.8.)

The fact that ξ is an analytic function means that it has a convergent Taylor expansion, so for small-enough ϵ

$$\xi(\epsilon) \approx \xi(0) + \xi'(0)\epsilon$$

The derivative will tell us the sensitivity. Take the derivative of $p_\epsilon(\xi(\epsilon)) = 0$ wrt ϵ, then set $\epsilon = 0$:

$$\xi'(0)p'(\xi(0)) + g(\xi(0)) = 0 \;\Rightarrow\; \xi'(0) = -\frac{g(\xi(0))}{p'(\xi(0))}.$$

Example (Wilkinson, 1959): $p(x) = \Pi_{i=1}^{20}(x - i)$. Roots are all simple and well-separated. Look at the sensitivity of the last root $x = 20$.

$$p'(x) = \sum_{j=1}^{20} \Pi_{i=1, i\neq j}^{20}(x - i).$$

Evaluate at $x = 20$; every term that has a factor of $x - 20$ will be zero, and there is only one term without that factor, so

$$p'(20) = 19!$$

That's great! The sensitivity is $g(20)/p'(20) = g(20)/19!$ which should be small. Suppose you want to perturb the constant coefficient which is $a_0 = 20!$, and you set $a_0 \leftarrow (1 + \epsilon)a_0$, i.e. $g(x) = a_0$. So the root of the perturbed polynomial, for small ϵ, is

$$\xi(\epsilon) \approx 20 - \epsilon\frac{20!}{19!} = 20(1 - \epsilon).$$

Perhaps not as small as you thought. What if you perturb the second-to-last coefficient $a_{19} \leftarrow a_{19}(1 + \epsilon)$

$$a_{19} = -(1 + 2 + \ldots + 20) = -210, \text{ i.e. } g(x) = a_{19}x^{19}.$$

So the root of the perturbed polynomial, for small ϵ, is

$$\xi(\epsilon) \approx 20 + \epsilon\frac{210 \times 20^{19}}{19!}.$$

The sensitivity is $\approx 0.9 \times 10^{10}$.

What about the root $x = 1$? $p'(1) = -19!$. If you perturb $a_0 \leftarrow a_0(1 + \epsilon)$ you find

$$\xi(\epsilon) \approx 1 + 20\epsilon$$

If you perturb a_{19} you find

$$\xi(\epsilon) \approx 1 - \epsilon\frac{210}{19!}.$$

This behavior is typical: roots are more sensitive to changes in coefficients of higher order, and roots far from the origin are more sensitive to perturbations.

Solving Nonlinear Systems of Equations
Atkinson §2.10, Stoer & Bulirsch Chapter 5

1 Fixed-point iteration!

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k).$$

Theory is very similar to 1D. One can consider the case of contractive maps (Lipschitz constant less than 1), but we will jump straight to the continuously-differentiable case, which often occurs in practice. In 1D we saw that if $g'(\alpha) < 1$ (continuously-differentiable) then there is an interval that maps to itself, and that the iteration will converge to the fixed point for any 'close enough' initial condition. We used a 1D Mean Value Theorem $g(\alpha) - g(x) = g'(\xi)(\alpha - x)$ for some ξ between α and x. We need a generalization of this to multiple dimensions.

Consider a scalar function of multiple variables $g_i(\boldsymbol{x})$. Along a line connecting two points $\boldsymbol{\alpha}$ and $\boldsymbol{x}$, this is just a univariate function so the 1D MVT applies:

$$g_i(\boldsymbol{\alpha}) - g_i(\boldsymbol{x}) = (\nabla g_i)|_{\boldsymbol{\xi}}^T(\boldsymbol{\alpha} - \boldsymbol{x})$$

where $\boldsymbol{\xi}$ is a point along the line connecting $\boldsymbol{\alpha}$ and $\boldsymbol{x}$. If we have a vector of functions $\boldsymbol{g} = \{g_i\}_{i=1}^n$, we can just apply this repeatedly

$$\boldsymbol{g}(\boldsymbol{\alpha}) - \boldsymbol{g}(\boldsymbol{x}) = \tilde{\mathbf{G}}(\boldsymbol{\alpha} - \boldsymbol{x}).$$

The catch is that $\tilde{\mathbf{G}}$ is not the Jacobian matrix of $\boldsymbol{g}$ evaluated at some point. Instead, the i^{th} row of $\tilde{\mathbf{G}}$ is the i^{th} row of $\mathbf{G}$ (the Jacobian) evaluated at some point $\boldsymbol{\xi}_i$.

(Brief review) $\boldsymbol{g}(\boldsymbol{x})$ is a vector of functions $\{g_i(\boldsymbol{x})\}_{i=1}^n$. The gradient of a scalar function is a column vector $\nabla g_i = (\partial_{x_1} g_i, \partial_{x_2} g_i, \ldots, \partial_{x_n} g_i)^T$. The Jacobian of $\boldsymbol{g}$ is a matrix whose rows are the transposes of the gradients

$$\mathbf{G} = \begin{pmatrix} \nabla g_1^T \\ \nabla g_2^T \\ \vdots \\ \nabla g_n^T \end{pmatrix}.$$

Theorem If $\boldsymbol{g}$ is continuously-differentiable, and the ∞-norm of the Jacobian matrix $\mathbf{G}$ at the fixed point is strictly less than 1, then there is an open ball (we need convexity) containing the fixed point such that all iterations started within the ball converge to the fixed point.

(It's convenient to consider the $\|\cdot\|_\infty$ norm, which is the max row sum, since each row of $\tilde{\mathbf{G}}$ is a row of the Jacobian.)

Proof: By assumption $\|\mathbf{G}\|_\infty < 1$ at the fixed point; since $\mathbf{G}$ is continuous, there must be some non-empty closed ball B centered on the fixed point where

$$\max_{\boldsymbol{x} \in B} \|\mathbf{G}\|_\infty \leq \lambda < 1.$$

Since we're using the ∞ norm (max row sum), and since each row of $\tilde{\mathbf{G}}$ is a row of $\mathbf{G}$, we have

$$\max_{\boldsymbol{x} \in B} \|\tilde{\mathbf{G}}\|_\infty \leq \max_{\boldsymbol{x} \in B} \|\mathbf{G}\|_\infty$$

For any point $\boldsymbol{x}$ in this ball we have

$$\|\boldsymbol{\alpha} - \boldsymbol{x}_{k+1}\|_\infty = \|\boldsymbol{g}(\boldsymbol{\alpha}) - \boldsymbol{g}(\boldsymbol{x}_k)\|_\infty = \|\tilde{\mathbf{G}}(\alpha - \boldsymbol{x}_k)\|_\infty \leq \|\tilde{\mathbf{G}}\|_\infty \|\alpha - \boldsymbol{x}_k\|_\infty \leq \lambda \|\boldsymbol{\alpha} - \boldsymbol{x}_k\|_\infty, \;\; 0 < \lambda < 1.$$

This implies convergence.

2 Newton's method for $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$. For a nonlinear system just replace division by inversion

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \mathbf{F}_k^{-1} \boldsymbol{f}_k$$

where $\mathbf{F}_k$ is the jacobian matrix of $\boldsymbol{f}$ at $\boldsymbol{x}_k$. Naturally you do this in a two-step process

$$\text{First solve } \mathbf{F}_k\boldsymbol{\delta}_k = -\boldsymbol{f}_k, \quad \text{then set } \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\delta}_k.$$

First we will show that this converges, using the fixed-point theory.

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\delta}(\boldsymbol{x})$$

$$\mathbf{G}(\boldsymbol{x}) = \mathbf{I} + \boldsymbol{\Delta}(\boldsymbol{x})$$

where $\boldsymbol{\Delta}(\boldsymbol{x})$ is the Jacobian matrix of $\boldsymbol{\delta}$. We find this by implicit differentiation

$$\mathbf{F}(x)\boldsymbol{\delta}(x) = -\boldsymbol{f}(x)$$

$$\mathcal{H}\boldsymbol{\delta} + \mathbf{F}\boldsymbol{\Delta} = -\mathbf{F}$$

where $\mathcal{H}$ is the Hessian (It's a third-order tensor). Now plug in $\boldsymbol{\delta} = -\mathbf{F}^{-1}\boldsymbol{f}$ and solve for $\boldsymbol{\Delta}$

$$\boldsymbol{\Delta} = \mathbf{F}^{-1}\left(-\mathbf{F} + \mathcal{H}\mathbf{F}^{-1}\boldsymbol{f}\right)$$

Our fixed-point theory is based on the Jacobian of the iteration function evaluated at the fixed point, so we want to evaluate $\boldsymbol{\Delta}$ at the solution, i.e. where $\boldsymbol{f} = \mathbf{0}$, so

$$\boldsymbol{\Delta}|_{\text{the solution}} = -\mathbf{I}$$

which implies

$$\mathbf{G} = \mathbf{0} \text{ at the solution.}$$

This proves convergence (if the function is 'nice enough' and the initial condition is 'close enough'), and suggests faster-than-linear convergence. The proof assumes that $\mathbf{F}$ is invertible in a neighborhood of the solution, which is just like assuming that the root is 'simple' in the 1D case.

3 Proof of quadratic convergence for Newton's method for systems. Closely follows S&B §5.3. First note that in the 1D case we have the standard result

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

that follows from the Taylor expansion of f about x. If f'' is nice enough then we'll have

$$|f(x) - f(y) - f'(x)(x-y)| \le \frac{c}{2}|x-y|^2$$

for all x and y in some set, and we can use this to establish quadratic convergence.

There is an n-dimensional version of the Taylor expansion, but it's unwieldy and we don't need its full detail. Instead we just need an expression for

$$\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y}) - \mathbf{F}(\boldsymbol{x}-\boldsymbol{y})\|$$

and we want it to be bounded above by a constant times $\|\boldsymbol{x}-\boldsymbol{y}\|^2$. An appropriate expression is given by the following

> (S&B Lemma 5.3.1) If $\mathbf{F}$ (the Jacobian of $\boldsymbol{f}$) exists for all $\boldsymbol{x}$ in a convex region $C_0 \subset \mathbb{R}^n$ and if a constant γ exists with
>
> $$\|\mathbf{F}(\boldsymbol{x}) - \mathbf{F}(\boldsymbol{y})\| \le \gamma\|\boldsymbol{x}-\boldsymbol{y}\|$$
>
> for all $\boldsymbol{x}, \boldsymbol{y} \in C_0$, then for all $\boldsymbol{x}, \boldsymbol{y} \in C_0$ we have
>
> $$\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y}) - [\mathbf{F}(\boldsymbol{y})](\boldsymbol{x}-\boldsymbol{y})\| \le \frac{\gamma}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2.$$

Proof: Consider $\varphi(t) = f(y + t(x - y))$ for $t \in [0, 1]$, which has derivative

$$\varphi'(t) = [\mathbf{F}(y + t(x - y))](x - y).$$

The quantity we care about can be written as

$$\varphi(1) - \varphi(0) - \varphi'(0) = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y}) - [\mathbf{F}(\boldsymbol{x})](\boldsymbol{x} - \boldsymbol{y}).$$

We can also write this as

$$\varphi(1) - \varphi(0) - \varphi'(0) = \int_0^1 (\varphi'(t) - \varphi'(0))\mathrm{d}t.$$

We can write the integrand as follows

$$\varphi'(t) - \varphi'(0) = [\mathbf{F}(y + t(\boldsymbol{x} - y)) - \mathbf{F}(\boldsymbol{y})](x - y).$$

Now note that by the assumptions on the function $\boldsymbol{f}$ we have

$$\|\varphi'(t) - \varphi'(0)\| = \|[\mathbf{F}(y + t(\boldsymbol{x} - y)) - \mathbf{F}(\boldsymbol{y})](x - y)\| \leq \gamma t \|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

We can use this to provide a bound on the expression of interest:

$$\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y}) - [\mathbf{F}(\boldsymbol{x})](\boldsymbol{x} - \boldsymbol{y})\| \leq \gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2 \int_0^1 t\mathrm{d}t = \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

□

We're now in a position to prove quadratic convergence. We've already proven convergence for a class of functions, so let's skip the convergence proof in S&B Theorem 5.3.2 and forge ahead on our own. First note that

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \mathbf{F}_k^{-1}\boldsymbol{f}_k \Rightarrow \mathbf{F}_k(\alpha - \boldsymbol{x}_{k+1}) = \mathbf{F}_k(\alpha - \boldsymbol{x}_k) + \boldsymbol{f}_k = \mathbf{F}_k(\alpha - \boldsymbol{x}_k) + \boldsymbol{f}_k - \boldsymbol{f}(\alpha).$$

So we can write

$$\begin{aligned}\|\alpha - \boldsymbol{x}_{k+1}\| &= \| - \mathbf{F}_k^{-1}\left(\boldsymbol{f}(\alpha) - \boldsymbol{f}_k - \mathbf{F}_k(\alpha - \boldsymbol{x}_k)\right)\| \\ &\leq \|\mathbf{F}_k^{-1}\|\|\boldsymbol{f}(\alpha) - \boldsymbol{f}_k - \mathbf{F}_k(\alpha - \boldsymbol{x}_k)\|\end{aligned}$$

We're now in a position to use Lemma 5.3.1 to get

$$\|\alpha - \boldsymbol{x}_{k+1}\| \leq \frac{\gamma}{2}\|\mathbf{F}_k^{-1}\|\|\boldsymbol{\alpha} - \boldsymbol{x}_k\|^2.$$

In summary, as long as $\boldsymbol{F}$ is Lipschitz continuous in the sense of 5.3.1 and $\|\mathbf{F}^{-1}\|$ is not infinity at the root (i.e. it's a simple root), then we have quadratic convergence.

4 Unconstrained optimization (Atkinson §2.12?). This is a large topic and we only touch on it very briefly. Consider trying to minimize the function $\psi(\boldsymbol{x}) \geq 0$ over $\boldsymbol{x} \in \mathbb{R}^n$. If the function ψ is continuously differentiable then minima must occur at critical points defined by $\nabla\psi = \mathbf{0}$. If we're trying to find a value of $\boldsymbol{x}$ where $\nabla\psi = \mathbf{0}$ then we're solving a nonlinear system of $n \times n$ equations of the form $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$ where $\boldsymbol{f}(\boldsymbol{x}) = \nabla\psi$. We can apply a fixed-point iteration or, if the Hessian matrix of ψ is available we can use Newton's method.

The following is basically just a quick review of topics from multivariate calculus. If we can find a critical point it might not be a local minimum; it could be, e.g., a saddle point or a maximum. If ψ is twice differentiable, then you can check if your critical point is a minimum by evaluating the Hessian (the matrix of second partial derivatives) at the critical point. The critical point is a minimum when the Hessian is positive definite or semi-definite.