Numerical Analysis Lecture Notes

Peter J. Olver

9. Numerical Solution of Algebraic Systems

In this part, we discuss basic iterative methods for solving systems of algebraic equations. By far the most common is a vector-valued version of Newton's Method, which will form our primary object of study.

9.1. Vector–Valued Iteration.

Extending the preceding analysis to vector-valued iterative systems is not especially difficult. We will build on our experience with linear iterative systems from Chapter 7.

We begin by fixing a norm $\| \cdot \|$ on $\mathbb{R}^n$. Since we will also be computing the associated matrix norm $\| A \|$, as defined in Theorem 7.13, it may be more convenient for computations to adopt either the 1 or the ∞ norms rather than the standard Euclidean norm.

We begin by defining the vector-valued counterpart of the basic linear convergence condition (2.21).

Definition 9.1. A function $\mathbf{g}\colon \mathbb{R}^n \to \mathbb{R}^n$ is a *contraction* at a point $\mathbf{u}^\star \in \mathbb{R}^n$ if there exists a constant $0 \le \sigma < 1$ such that

$$\| \mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{u}^\star) \| \ \le \ \sigma \| \mathbf{u} - \mathbf{u}^\star \| \tag{9.1}$$

for all $\mathbf{u}$ sufficiently close to $\mathbf{u}^\star$, i.e., $\| \mathbf{u} - \mathbf{u}^\star \| < \delta$ for some fixed $\delta > 0$.

Remark: The notion of a contraction depends on the underlying choice of matrix norm. Indeed, the linear function $\mathbf{g}(\mathbf{u}) = A\mathbf{u}$ if and only if $\| A \| < 1$, which implies that A is a convergent matrix. While every convergent matrix satisfies $\| A \| < 1$ in *some* matrix norm, and hence defines a contraction relative to that norm, it may very well have $\| A \| > 1$ in a particular norm, violating the contaction condition; see (7.31) for an explicit example.

Theorem 9.2. *If* $\mathbf{u}^\star = \mathbf{g}(\mathbf{u}^\star)$ *is a fixed point for the discrete dynamical system* (2.1) *and* $\mathbf{g}$ *is a contraction at* $\mathbf{u}^\star$, *then* $\mathbf{u}^\star$ *is an asymptotically stable fixed point.*

Proof: The proof is a copy of the last part of the proof of Theorem 2.6. We write

$$\| \mathbf{u}^{(k+1)} - \mathbf{u}^\star \| = \| \mathbf{g}(\mathbf{u}^{(k)}) - \mathbf{g}(\mathbf{u}^\star) \| \le \sigma \| \mathbf{u}^{(k)} - \mathbf{u}^\star \|,$$

using the assumed estimate (9.1). Iterating this basic inequality immediately demonstrates that

$$\| \mathbf{u}^{(k)} - \mathbf{u}^\star \| \;\leq\; \sigma^k \, \| \mathbf{u}^{(0)} - \mathbf{u}^\star \| \qquad \text{for} \qquad k = 0, 1, 2, 3, \ldots \;. \tag{9.2}$$

Since $\sigma < 1$, the right hand side tends to 0 as $k \to \infty$, and hence $\mathbf{u}^{(k)} \to \mathbf{u}^\star$. *Q.E.D.*

In most interesting situations, the function $\mathbf{g}$ is differentiable, and so can be approximated by its first order Taylor polynomial

$$\mathbf{g}(\mathbf{u}) \approx \mathbf{g}(\mathbf{u}^\star) + \mathbf{g}'(\mathbf{u}^\star)\,(\mathbf{u} - \mathbf{u}^\star) = \mathbf{u}^\star + \mathbf{g}'(\mathbf{u}^\star)\,(\mathbf{u} - \mathbf{u}^\star). \tag{9.3}$$

Here

$$\mathbf{g}'(\mathbf{u}) = \begin{pmatrix} \dfrac{\partial g_1}{\partial u_1} & \dfrac{\partial g_1}{\partial u_2} & \cdots & \dfrac{\partial g_1}{\partial u_n} \\ \dfrac{\partial g_2}{\partial u_1} & \dfrac{\partial g_2}{\partial u_2} & \cdots & \dfrac{\partial g_2}{\partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial g_n}{\partial u_1} & \dfrac{\partial g_n}{\partial u_2} & \cdots & \dfrac{\partial g_n}{\partial u_n} \end{pmatrix}, \tag{9.4}$$

denotes the $n \times n$ *Jacobian matrix* of the vector-valued function $\mathbf{g}$, whose entries are the partial derivatives of its individual components. Since $\mathbf{u}^\star$ is fixed, the the right hand side of (9.3) is an affine function of $\mathbf{u}$. Moreover, $\mathbf{u}^\star$ remains a fixed point of the affine approximation. Proposition 7.25 tells us that iteration of the affine function will converge to the fixed point if and only if its coefficient matrix, namely $\mathbf{g}'(\mathbf{u}^\star)$, is a convergent matrix, meaning that its spectral radius $\rho(\mathbf{g}'(\mathbf{u}^\star)) < 1$. This observation motivates the following theorem and corollary.

Theorem 9.3. *Let $\mathbf{u}^\star$ be a fixed point for the discrete dynamical system $\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)})$. If the Jacobian matrix norm $\| \mathbf{g}'(\mathbf{u}^\star) \| < 1$, then $\mathbf{g}$ is a contraction at $\mathbf{u}^\star$, and hence the fixed point $\mathbf{u}^\star$ is asymptotically stable.*

Proof: The first order Taylor expansion of $\mathbf{g}(\mathbf{u})$ at the fixed point $\mathbf{u}^\star$ takes the form

$$\mathbf{g}(\mathbf{u}) = \mathbf{g}(\mathbf{u}^\star) + \mathbf{g}'(\mathbf{u}^\star)\,(\mathbf{u} - \mathbf{u}^\star) + R(\mathbf{u} - \mathbf{u}^\star), \tag{9.5}$$

where the remainder term satisfies

$$\lim_{\mathbf{u} \to \mathbf{u}^\star} \frac{R(\mathbf{u} - \mathbf{u}^\star)}{\| \mathbf{u} - \mathbf{u}^\star \|} \;=\; 0.$$

Let $\varepsilon > 0$ be such that

$$\sigma = \| \mathbf{g}'(\mathbf{u}^\star) \| + \varepsilon < 1.$$

Choose $0 < \delta < 1$ such that $\| R(\mathbf{u} - \mathbf{u}^\star) \| \leq \varepsilon \| \mathbf{u} - \mathbf{u}^\star \|$ whenever $\| \mathbf{u} - \mathbf{u}^\star \| \leq \delta$. For such $\mathbf{u}$, we have, by the Triangle Inequality,

$$\begin{aligned} \| \mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{u}^\star) \| \;&\leq\; \| \mathbf{g}'(\mathbf{u}^\star)\,(\mathbf{u} - \mathbf{u}^\star) \| + \| R(\mathbf{u} - \mathbf{u}^\star) \| \\ &\leq\; \bigl(\| \mathbf{g}'(\mathbf{u}^\star) \| + \varepsilon \bigr) \, \| \mathbf{u} - \mathbf{u}^\star \| = \sigma \, \| \mathbf{u} - \mathbf{u}^\star \|, \end{aligned}$$

which establishes the contraction inequality (9.1). *Q.E.D.*

Corollary 9.4. *If the Jacobian matrix $\mathbf{g}'(\mathbf{u}^\star)$ is a convergent matrix, meaning that its spectral radius satisfies $\rho\big(\mathbf{g}'(\mathbf{u}^\star)\big) < 1$, then $\mathbf{u}^\star$ is an asymptotically stable fixed point.*

Proof: Corollary 7.23 assures us that $\|\, \mathbf{g}'(\mathbf{u}^\star) \,\| < 1$ in some matrix norm. Using this norm, the result immediately follows from the theorem. *Q.E.D.*

Theorem 9.3 tells us that initial values $\mathbf{u}^{(0)}$ that are sufficiently near a stable fixed point $\mathbf{u}^\star$ are guaranteed to converge to it. In the linear case, closeness of the initial data to the fixed point was not, in fact, an issue; all stable fixed points are, in fact, globally stable. For nonlinear iteration, it is of critical importance, and one does not typically expect iteration starting with far away initial data to converge to the desired fixed point. An interesting (and difficult) problem is to determine the so-called *basin of attraction* of a stable fixed point, defined as the set of all initial data that ends up converging to it. As in the elementary logistic map (2.22), initial values that lie outside a basin of attraction can lead to divergent iterates, periodic orbits, or even exhibit chaotic behavior. The full range of possible phenomena is a topic of contemporary research in dynamical systems theory, [**46**], and in numerical analysis, [**1**].

Example 9.5. Consider the function

$$\mathbf{g}(u,v) = \begin{pmatrix} -\frac{1}{4}u^3 + \frac{9}{8}u + \frac{1}{4}v^3 \\ \frac{3}{4}v - \frac{1}{2}uv \end{pmatrix}.$$

There are four (real) fixed points; stability is determined by the size of the eigenvalues of the Jacobian matrix

$$\mathbf{g}'(u,v) = \begin{pmatrix} \frac{9}{8} - \frac{3}{4}u^2 & -\frac{1}{2}v \\ \frac{3}{4}v^2 & \frac{3}{4} - \frac{1}{2}u \end{pmatrix}$$

at each of the fixed points. The results are summarized in the following table:

fixed point	$\mathbf{u}_1^\star = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\mathbf{u}_2^\star = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$	$\mathbf{u}_3^\star = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$	$\mathbf{u}_4^\star = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$
Jacobian matrix	$\begin{pmatrix} \frac{9}{8} & 0 \\ 0 & \frac{3}{4} \end{pmatrix}$	$\begin{pmatrix} \frac{3}{4} & 0 \\ 0 & \frac{3}{4} - \frac{1}{2\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{3}{4} & 0 \\ 0 & \frac{3}{4} + \frac{1}{2\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{15}{16} & -\frac{1}{4} \\ \frac{3}{16} & 1 \end{pmatrix}$
eigenvalues	$1.125,\ .75$	$.75,\ .396447$	$1.10355,\ .75$	$.96875 \pm .214239\,\mathrm{i}$
spectral radius	1.125	$.75$	1.10355	$.992157$

Thus, $\mathbf{u}_2^\star$ and $\mathbf{u}_4^\star$ are stable fixed points, whereas $\mathbf{u}_1^\star$ and $\mathbf{u}_3^\star$ are both unstable. Indeed, starting with $\mathbf{u}^{(0)} = (\,.5, .5\,)^T$, it takes 24 iterates to converge to $\mathbf{u}_2^\star$ with 4 significant decimal digits, whereas starting with $\mathbf{u}^{(0)} = (\,-.7, .7\,)^T$, it takes 1049 iterates to converge to within 4 digits of $\mathbf{u}_4^\star$; the slower convergence rate is predicted by the larger Jacobian spectral radius. The two basins of attraction are plotted in Figure 9.1. The stable fixed

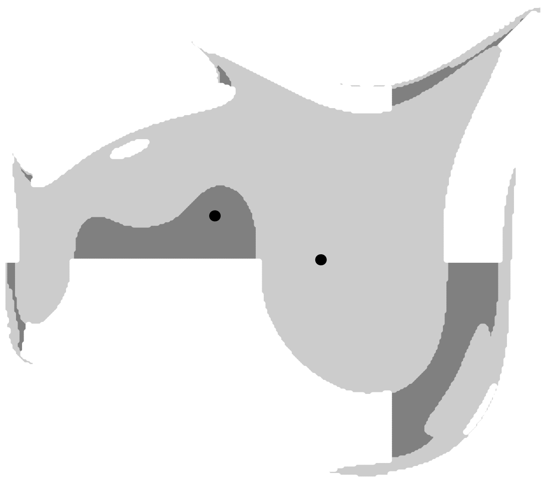

Figure 9.1. Basins of Attraction.

points are indicated by black dots. The light gray region contains $\mathbf{u}_2^\star$ and indicates all the points that converge to it; the darker gray indicates points converging, more slowly, to $\mathbf{u}_4^\star$. All other initial points, except $\mathbf{u}_1^\star$ and $\mathbf{u}_3^\star$, have rapidly unbounded iterates: $\| \mathbf{u}^{(k)} \| \to \infty$.

The smaller the spectral radius or matrix norm of the Jacobian matrix at the fixed point, the faster the nearby iterates will converge to it. As in the scalar case, quadratic convergence will occur when the Jacobian matrix $\mathbf{g}'(\mathbf{u}^\star) = \mathrm{O}$ is the zero matrix[†], i.e., *all* first order partial derivatives of the components of $\mathbf{g}$ vanish at the fixed point. The quadratic convergence estimate

$$\| \mathbf{u}^{(k+1)} - \mathbf{u}^\star \| \ \leq \ \tau \, \| \mathbf{u}^{(k)} - \mathbf{u}^\star \|^2 \tag{9.6}$$

is a consequence of the second order Taylor expansion at the fixed point. Details of the proof are left as an exercise.

Of course, in practice we don't know the norm or spectral radius of the Jacobian matrix $\mathbf{g}'(\mathbf{u}^\star)$ because we don't know where the fixed point is. This apparent difficulty can be easily circumvented by requiring that $\| \mathbf{g}'(\mathbf{u}) \| < 1$ for all $\mathbf{u}$ — or, at least, for all $\mathbf{u}$ in a domain Ω containing the fixed point. In fact, this hypothesis can be used to prove the exitence and uniqueness of asymptotically stable fixed points. Rather than work with the Jacobian matrix, let us return to the contraction condition (9.1), but now imposed uniformly on an entire domain.

Definition 9.6. A function $\mathbf{g}\colon \mathbb{R}^n \to \mathbb{R}^n$ is called a *contraction mapping* on a domain $\Omega \subset \mathbb{R}^n$ if

(a) it maps Ω to itself, so $\mathbf{g}(\mathbf{u}) \in \Omega$ whenever $\mathbf{u} \in \Omega$, and

(b) there exists a *constant* $0 \leq \sigma < 1$ such that

$$\| \mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{v}) \| \ \leq \ \sigma \, \| \mathbf{u} - \mathbf{v} \| \qquad \text{for all} \qquad \mathbf{u}, \mathbf{v} \in \Omega. \tag{9.7}$$

[†] Having zero spectral radius is not sufficient for quadratic convergence; see Exercise 9.1.11.

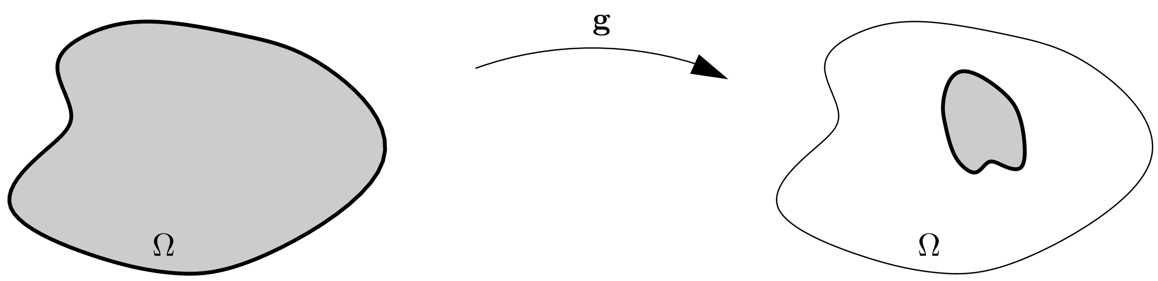

Figure 9.2. A Contraction Mapping.

In other words, applying a contraction mapping reduces the mutual distance between points. In view of Theorem 9.3, we can detect contraction mappings by lloking at the norm of their Jacobian matrix.

Lemma 9.7. *If* $\mathbf{g}\colon \Omega \to \Omega$ *and* $\| \mathbf{g}'(\mathbf{u}) \| < 1$ *for all* $\mathbf{u} \in \Omega$*, then* $\mathbf{g}$ *is a contraction mapping.*

So, as its name indicates, a contraction mapping effectively shrinks the size of its domain; see Figure 9.2. As the iterations proceed, the successive image domains become smaller and smaller. If the original domain is closed and bounded, then it is forced to shrink down to a single point, which is the unique fixed point of the iterative system, leading to the *Contraction Mapping Theorem*.

Theorem 9.8. *If* $\mathbf{g}$ *is a contraction mapping on a closed bounded domain* $\Omega \subset \mathbb{R}^n$*, then* $\mathbf{g}$ *admits a unique fixed point* $\mathbf{u}^\star \in \Omega$*. Moreover, starting with any initial point* $\mathbf{u}^{(0)} \in \Omega$*, the iterates* $\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)})$ *necessarily converge to the fixed point:* $\mathbf{u}^{(k)} \to \mathbf{u}^\star$*.*

9.2. Solution of Algebraic Systems.

There is no direct universal solution method for nonlinear systems comparable to Gaussian elimination. Numerical solution techniques rely almost exclusively on iterative algorithms. This section presents the principal methods for numerically approximating the solution(s) to a nonlinear system. We shall only discuss general purpose algorithms; specialized methods for solving particular classes of equations, e.g., polynomial equations, can be found in numerical analysis texts, e.g., [**5**, **7**, **47**]. Of course, the most important specialized methods — those designed for solving linear systems — will continue to play a critical role, even in the nonlinear regime.

We concentrate on the "regular" case when the system contains the same number of equations as unknowns:

$$f_1(u_1, \ldots, u_n) = 0, \qquad \ldots \qquad f_n(u_1, \ldots, u_n) = 0. \tag{9.8}$$

We will rewrite the system in vector form

$$\mathbf{f}(\mathbf{u}) = \mathbf{0}, \tag{9.9}$$

where $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued function of n variables. In practice, we do not necessarily require that $\mathbf{f}$ be defined on all of $\mathbb{R}^n$, although this does simplify the exposition.

We shall only consider solutions that are separated from any others. More formally:

Definition 9.9. A solution $\mathbf{u}^\star$ to a system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ is called *isolated* if there exists $\delta > 0$ such that $\mathbf{f}(\mathbf{u}) \neq \mathbf{0}$ for all $\mathbf{u}$ satisfying $0 < \| \mathbf{u} - \mathbf{u}^\star \| < \delta$.

Example 9.10. Consider the planar equation

$$x^2 + y^2 = (x^2 + y^2)^2.$$

Rewriting the equation in polar coordinates as

$$r = r^2 \qquad \text{or} \qquad r(r-1) = 0,$$

we immediately see that the solutions consist of the origin $x = y = 0$ and all points on the unit circle $r^2 = x^2 + y^2 = 1$. Only the origin is an isolated solution, since every solution lying on the circle has plenty of other points on the circle that lie arbitrarily close to it.

Typically, solutions to a system of n equations in n unknowns are isolated, although this is not always true. For example, if A is a singular $n \times n$ matrix, then the solutions to the homogeneous linear system $A\mathbf{u} = \mathbf{0}$ form a nontrivial subspace, and so are not isolated. Nonlinear systems with non-isolated solutions can similarly be viewed as exhibiting some form of degeneracy. In general, the numerical computation of non-isolated solutions, e.g., solving the implicit equations for a curve or surface, is a much more difficult problem, and we will not attempt to discuss these issues in this introductory presentation. (However, our continuation approach to the Kepler equation in Example 2.20 indicates how one might proceed in such situations.)

In the case of a single scalar equation, the simple roots, meaning those for which $f'(u^\star) \neq 0$, are the easiest to compute. In higher dimensions, the role of the derivative of the function is played by the Jacobian matrix (9.4), and this motivates the following definition.

Definition 9.11. A solution $\mathbf{u}^\star$ to a system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ is called *nonsingular* if the associated Jacobian matrix is nonsingular there: $\det \mathbf{f}'(\mathbf{u}^\star) \neq 0$.

Note that the Jacobian matrix is square if and only if the system has the same number of equations as unknowns, which is thus one of the requirements for a solution to be nonsingular in our sense. Moreover, the Inverse Function Theorem from multivariable calculus, [**2**, **36**], implies that a nonsingular solution is necessarily isolated.

Theorem 9.12. *Every nonsingular solution* $\mathbf{u}^\star$ *to a system* $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ *is isolated.*

Being the multivariate counterparts of simple roots also means that nonsingular solutions of systems are the most amenable to practical computation. Computing non-isolated solutions, as well as isolated solutions with a singular Jacobian matrix, is a considerable challenge, and practical algorithms remain much less well developed. For this reason, we focus exclusively on numerical solution techniques for nonsingular solutions.

Now, let us turn to numerical solution techniques. The first remark is that, unlike the scalar case, proving existence of a solution to a system of equations is often a challenging issue. There is no counterpart to the Intermediate Value Lemma 2.12 for vector-valued functions. It is not hard to find vector-valued functions whose entries take on both positive and negative values, but admit no solutions. This precludes any simple analog of the Bisection Method for nonlinear systems in more than one unknown.

On the other hand, Newton's Method can be straightforwardly adapted to compute nonsingular solutions to systems of equations, and is *the* most widely used method for this purpose. The derivation proceeds in very similar manner to the scalar case. First, we replace the system (9.9) by a fixed point system

$$\mathbf{u} = \mathbf{g}(\mathbf{u}) \tag{9.10}$$

having the same solutions. By direct analogy with (2.33), any (reasonable) fixed point method will take the form

$$\mathbf{g}(\mathbf{u}) = \mathbf{u} - L(\mathbf{u})\,\mathbf{f}(\mathbf{u}), \tag{9.11}$$

where $L(\mathbf{u})$ is an $n \times n$ matrix-valued function. Clearly, if $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ then $\mathbf{g}(\mathbf{u}) = \mathbf{u}$; conversely, if $\mathbf{g}(\mathbf{u}) = \mathbf{u}$, then $L(\mathbf{u})\,\mathbf{f}(\mathbf{u}) = \mathbf{0}$. If we further require that the matrix $L(\mathbf{u})$ be nonsingular, i.e., $\det L(\mathbf{u}) \neq 0$, then every fixed point of the iterator (9.11) will be a solution to the system (9.9) and vice versa.

According to Theorem 9.3, the speed of convergence (if any) of the iterative method

$$\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)}) \tag{9.12}$$

is governed by the matrix norm (or, more precisely, the spectral radius) of the Jacobian matrix $\mathbf{g}'(\mathbf{u}^\star)$ at the fixed point. In particular, if

$$\mathbf{g}'(\mathbf{u}^\star) = \mathrm{O} \tag{9.13}$$

is the zero matrix, then the method converges quadratically fast. Let's figure out how this can be arranged. Computing the derivative using the matrix version of the Leibniz rule for the derivative of a matrix product, we find

$$\mathbf{g}'(\mathbf{u}^\star) = \mathrm{I} - L(\mathbf{u}^\star)\,\mathbf{f}'(\mathbf{u}^\star), \tag{9.14}$$

where I is the $n \times n$ identity matrix. (Fortunately, all the terms that involve derivatives of the entries of $L(\mathbf{u})$ go away since $\mathbf{f}(\mathbf{u}^\star) = \mathbf{0}$ by assumption.) Therefore, the quadratic convergence criterion (9.13) holds if and only if

$$L(\mathbf{u}^\star)\,\mathbf{f}'(\mathbf{u}^\star) = \mathrm{I}, \qquad \text{and hence} \qquad L(\mathbf{u}^\star) = \mathbf{f}'(\mathbf{u}^\star)^{-1} \tag{9.15}$$

should be the inverse of the Jacobian matrix of $\mathbf{f}$ at the solution, which, fortuitously, was already assumed to be nonsingular.

As in the scalar case, we don't know the solution $\mathbf{u}^\star$, but we can arrange that condition (9.15) holds by setting

$$L(\mathbf{u}) = \mathbf{f}'(\mathbf{u})^{-1}$$

everywhere — or at least everywhere that $\mathbf{f}$ has a nonsingular Jacobian matrix. The resulting fixed point system

$$\mathbf{u} = \mathbf{g}(\mathbf{u}) = \mathbf{u} - \mathbf{f}'(\mathbf{u})^{-1}\,\mathbf{f}(\mathbf{u}), \tag{9.16}$$

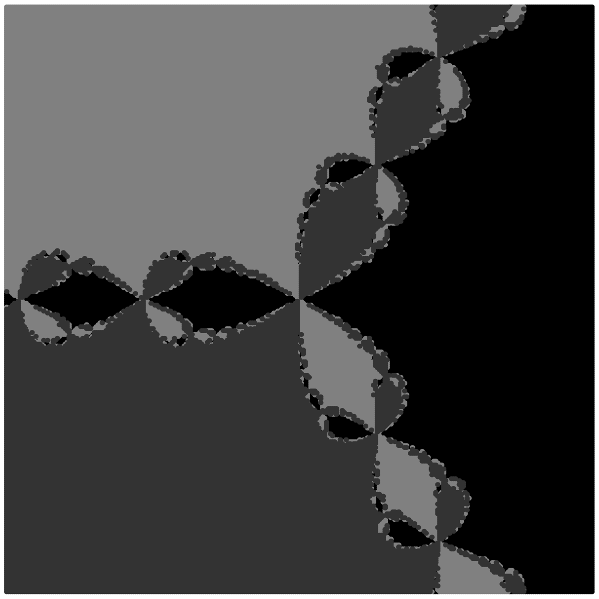

Figure 9.3. Computing the Cube Roots of Unity by Newton's Method.

leads to the quadratically convergent *Newton iteration scheme*

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mathbf{f}'(\mathbf{u}^{(k)})^{-1}\,\mathbf{f}(\mathbf{u}^{(k)}). \tag{9.17}$$

All it requires is that we guess an initial value $\mathbf{u}^{(0)}$ that is sufficiently close to the desired solution $\mathbf{u}^\star$. We are then guaranteed that the iterates $\mathbf{u}^{(k)}$ converge quadratically fast to $\mathbf{u}^\star$.

Theorem 9.13. *Let $\mathbf{u}^\star$ be a nonsingular solution to the system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$. Then, provided $\mathbf{u}^{(0)}$ is sufficiently close to $\mathbf{u}^\star$, the Newton iteration scheme* (9.17) *converges at a quadratic rate to the solution: $\mathbf{u}^{(k)} \to \mathbf{u}^\star$.*

Example 9.14. Consider the pair of simultaneous cubic equations

$$f_1(u,v) = u^3 - 3\,u\,v^2 - 1 = 0, \qquad\qquad f_2(u,v) = 3\,u^2\,v - v^3 = 0. \tag{9.18}$$

It is not difficult to prove that there are precisely three solutions:

$$\mathbf{u}_1^\star = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \mathbf{u}_2^\star = \begin{pmatrix} -.5 \\ .866025\ldots \end{pmatrix}, \qquad \mathbf{u}_3^\star = \begin{pmatrix} -.5 \\ -.866025\ldots \end{pmatrix}. \tag{9.19}$$

The Newton scheme relies on the Jacobian matrix

$$\mathbf{f}'(\mathbf{u}) = \begin{pmatrix} 3\,u^2 - 3\,v^2 & -\,6\,u\,v \\ 6\,u\,v & 3\,u^2 - 3\,v^2 \end{pmatrix}.$$

Since $\det \mathbf{f}'(\mathbf{u}) = 9(u^2 + v^2)$ is non-zero except at the origin, all three solutions are non-singular, and hence, for a sufficiently close initial value, Newton's Method will converge to the nearby solution. We explicitly compute the inverse Jacobian matrix:

$$\mathbf{f}'(\mathbf{u})^{-1} = \frac{1}{9(u^2+v^2)} \begin{pmatrix} 3\,u^2 - 3\,v^2 & 6\,u\,v \\ -\,6\,u\,v & 3\,u^2 - 3\,v^2 \end{pmatrix}.$$

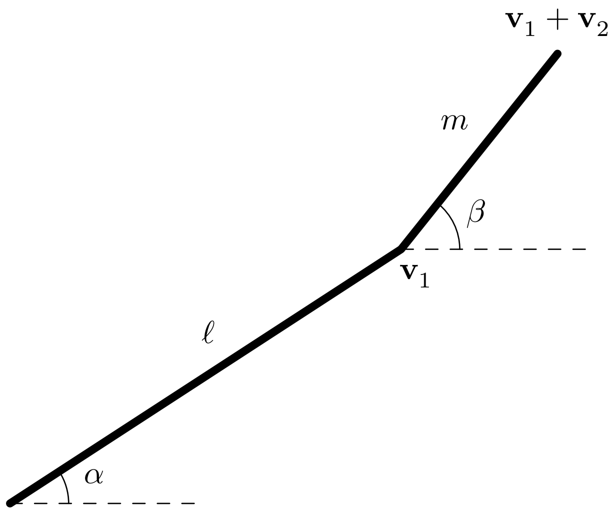

Figure 9.4. Robot Arm.

Hence, in this particular example, the Newton iterator (9.16) is

$$\mathbf{g}(\mathbf{u}) = \begin{pmatrix} u \\ v \end{pmatrix} - \frac{1}{9(u^2+v^2)} \begin{pmatrix} 3u^2 - 3v^2 & 6uv \\ -6uv & 3u^2 - 3v^2 \end{pmatrix} \begin{pmatrix} u^3 - 3uv^2 - 1 \\ 3u^2 v - v^3 \end{pmatrix}.$$

A complete diagram of the three basins of attraction, consisting of points whose Newton iterates converge to each of the three roots, has a remarkably complicated, fractal-like structure, as illustrated in Figure 9.3. In this plot, the x and y coordinates run from -1.5 to 1.5. The points in the black region all converge to $\mathbf{u}_1^\star$; those in the light gray region all converge to $\mathbf{u}_2^\star$; while those in the dark gray region all converge to $\mathbf{u}_3^\star$. The closer one is to the root, the sooner the iterates converge. On the interfaces between the basins of attraction are points for which the Newton iterates fail to converge, but exhibit a random, chaotic behavior. However, round-off errors will cause such iterates to fall into one of the basins, making it extremely difficult to observe such behavio over the long run.

Remark: The alert reader may notice that in this example, we are in fact merely computing the cube roots of unity, i.e., equations (9.18) are the real and imaginary parts of the complex equation $z^3 = 1$ when $z = u + \mathrm{i}\,v$.

Example 9.15. A robot arm consists of two rigid rods that are joined end-to-end to a fixed point in the plane, which we take as the origin $\mathbf{0}$. The arms are free to rotate, and the problem is to configure them so that the robot's hand ends up at the prescribed position $\mathbf{a} = (a, b)^T$. The first rod has length ℓ and makes an angle α with the horizontal, so its end is at position $\mathbf{v}_1 = (\ell\cos\alpha, \ell\sin\alpha)^T$. The second rod has length m and makes an angle β with the horizontal, and so is represented by the vector $\mathbf{v}_2 = (m\cos\beta, m\sin\beta)^T$. The hand at the end of the second arm is at position $\mathbf{v}_1 + \mathbf{v}_2$, and the problem is to find values for the angles α, β so that $\mathbf{v}_1 + \mathbf{v}_2 = \mathbf{a}$; see Figure 9.4. To this end, we need to solve the system of equations

$$\ell\cos\alpha + m\cos\beta = a, \qquad \ell\sin\alpha + m\sin\beta = b, \tag{9.20}$$

for the angles α, β.

To find the solution, we shall apply Newton's Method. First, we compute the Jacobian matrix of the system with respect to α, β, which is

$$\mathbf{f}'(\alpha,\beta) = \begin{pmatrix} -\,\ell\sin\alpha & -\,m\sin\beta \\ \ell\cos\alpha & m\cos\beta \end{pmatrix},$$

with inverse

$$\mathbf{f}'(\alpha,\beta)^{-1} = \frac{1}{\ell\, m\sin(\beta-\alpha)} \begin{pmatrix} -\,\ell\sin\alpha & m\sin\beta \\ -\,\ell\cos\alpha & m\cos\beta \end{pmatrix}.$$

As a result, the Newton iteration equation (9.17) has the explicit form

$$\begin{pmatrix} \alpha^{(k+1)} \\ \beta^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha^{(k)} \\ \beta^{(k)} \end{pmatrix} - \\ - \frac{1}{\ell\, m\sin(\beta^{(k)}-\alpha^{(k)})} \begin{pmatrix} -\,\ell\cos\alpha^{(k)} & m\sin\beta^{(k)} \\ -\,\ell\cos\alpha^{(k)} & m\sin\beta^{(k)} \end{pmatrix} \begin{pmatrix} \ell\cos\alpha^{(k)} + m\cos\beta^{(k)} - a \\ \ell\sin\alpha^{(k)} + m\sin\beta^{(k)} - b \end{pmatrix}.$$

when running the iteration, one must be careful to avoid points at which $\alpha^{(k)} - \beta^{(k)} = 0$ or π, i.e., where the robot arm has straightened out.

As an example, let us assume that the rods have lengths $\ell = 2$, $m = 1$, and the desired location of the hand is at $\mathbf{a} = (\,1, 1\,)^T$. We start with an initial guess of $\alpha^{(0)} = 0$, $\beta^{(0)} = \frac{1}{2}\pi$, so the first rod lies along the x–axis and the second is perpendicular. The first few Newton iterates are given in the accompanying table. The first column is the iterate number k; the second and third columns indicate the angles $\alpha^{(k)}$, $\beta^{(k)}$ of the rods. The fourth and fifth give the position $(x^{(k)}, y^{(k)})^T$ of the joint or elbow, while the final two indicate the position $(z^{(k)}, w^{(k)})^T$ of the robot's hand.

Observe that the robot has rapidly converged to one of the two possible configurations. (Can you figure out what the second equilibrium is?) In general, convergence depends on the choice of initial configuration, and the Newton iterates do not always settle down to a fixed point. For instance, if $\|\,\mathbf{a}\,\| > \ell + m$, there is no possible solution, since the arms are too short for the hand to reach to desired location; thus, no choice of initial conditions will lead to a convergent scheme and the robot arm flaps around in a chaotic manner.

k	$\alpha^{(k)}$	$\beta^{(k)}$	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$	$w^{(k)}$
0	.0000	1.5708	2.0000	.0000	2.0000	1.0000
1	.0000	2.5708	2.0000	.0000	1.1585	.5403
2	.3533	2.8642	1.8765	.6920	.9147	.9658
3	.2917	2.7084	1.9155	.5751	1.0079	.9948
4	.2987	2.7176	1.9114	.5886	1.0000	1.0000
5	.2987	2.7176	1.9114	.5886	1.0000	1.0000

Now that we have gained a little experience with Newton's Method for systems of equations, some supplementary remarks are in order. As we learned, except perhaps in very low-dimensional situations, one should not directly invert a matrix, but rather use Gaussian elimination, or, in favorable situations, a linear iterative scheme, e.g., Jacobi, Gauss–Seidel or even SOR. So a better strategy is to leave the Newton system (9.17) in unsolved, implicit form

$$\mathbf{f}'(\mathbf{u}^{(k)})\,\mathbf{v}^{(k)} = -\,\mathbf{f}(\mathbf{u}^{(k)}), \qquad \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{v}^{(k)}. \tag{9.21}$$

Given the iterate $\mathbf{u}^{(k)}$, we compute the Jacobian matrix $\mathbf{f}'(\mathbf{u}^{(k)})$ and the right hand side $-\,\mathbf{f}(\mathbf{u}^{(k)})$, and then use our preferred linear systems solver to find $\mathbf{v}^{(k)}$. Adding $\mathbf{u}^{(k)}$ to the result immediately yields the updated approximation $\mathbf{u}^{(k+1)}$ to the solution.

The main bottleneck in the implementation of the Newton scheme, particularly for large systems, is solving the linear system in (9.21). The coefficient matrix $\mathbf{f}'(\mathbf{u}^{(k)})$ must be recomputed at each step of the iteration, and hence knowing the solution to the k^{th} linear system does not appear to help us solve the subsequent system. Pereforming a complete Gaussian elimination at every step will tend to slow down the algorithm, particularly in high dimensional situations involving many equations in many unknowns.

One simple dodge for speeding up the computation is to note that, once we start converging, $\mathbf{u}^{(k)}$ will be very close to $\mathbf{u}^{(k-1)}$ and so we will probably not go far wrong by using $\mathbf{f}'(\mathbf{u}^{(k-1)})$ in place of the updated Jacobian matrix $\mathbf{f}'(\mathbf{u}^{(k)})$. Since we have already solved the linear system with coefficient matrix $\mathbf{f}'(\mathbf{u}^{(k-1)})$, we know its $L\,U$ factorization, and hence can use Forward and Back Substitution to quickly solve the modified system

$$\mathbf{f}'(\mathbf{u}^{(k-1)})\,\mathbf{v}^{(k+1)} = -\,\mathbf{f}(\mathbf{u}^{(k)}), \qquad \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{v}^{(k)}. \tag{9.22}$$

If $\mathbf{u}^{(k+1)}$ is still close to $\mathbf{u}^{(k-1)}$, we can continue to use $\mathbf{f}'(\mathbf{u}^{(k-1)})$ as the coefficient matrix when proceeding on to the next iterate $\mathbf{u}^{(k+2)}$. We proceed in this manner until there has been a notable change in the iterates, at which stage we can revert to solving the correct, unmodified linear system (9.21) by Gaussian Elimination. This strategy may dramatically reduce the total amount of computation required to approximate the solution to a prescribed accuracy. The down side is that this *quasi-Newton scheme* is only linearly convergent, and so does not home in on the root as fast as the unmodified implementation. The user needs to balance the trade-off between speed of convergence versus amount of time needed to solve the linear system at each step in the process. See [**47**] for further discussion.