DRAFT V1.2

From

Math, Numerics, & Programming

(for Mechanical Engineers)

Masayuki Yano
James Douglass Penn
George Konidaris
Anthony T Patera

September 2012

Contents

Unit VII

Nonlinear Equations

Chapter 29

Newton Iteration

29.1 Introduction

The demonstration robot arm of Figure 29.1 is represented schematically in Figure 29.2. Note that although the robot of Figure 29.1 has three degrees-of-freedom ("shoulder," "elbow," and "waist"), we will be dealing with only two degrees-of-freedom — "shoulder" and "elbow" — in this assignment.

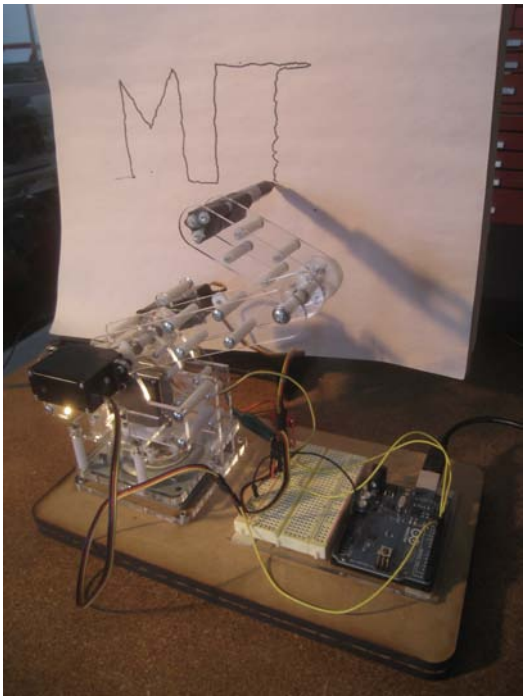


Figure 29.1: Demonstration robot arm. (Robot and photograph courtesy of James Penn.)

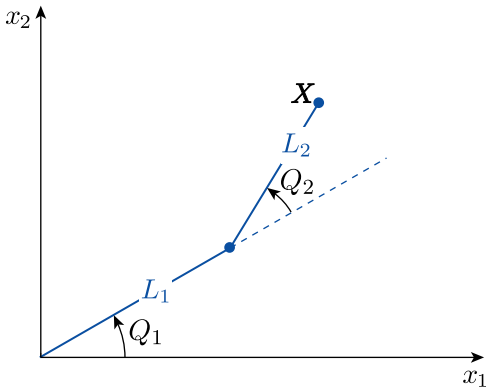


Figure 29.2: Schematic of robot arm.

The forward kinematics of the robot arm determine the coordinates of the end effector $\boldsymbol{X} =$

$[X_1, X_2]^{\mathrm{T}}$ for given joint angles $\boldsymbol{Q} = [Q_1, Q_2]^{\mathrm{T}}$ as

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}(\boldsymbol{Q}) = \begin{bmatrix} L_1\cos(Q_1) + L_2\cos(Q_1+Q_2) \\ L_1\sin(Q_1) + L_2\sin(Q_1+Q_2) \end{bmatrix}, \tag{29.1}$$

where L_1 and L_2 are the lengths of the first and second arm links, respectively. For our robot, $L_1 = 4$ inches and $L_2 = 3.025$ inches.

The inverse kinematics of the robot arm — the joint angles $\boldsymbol{Q}$ needed to realize a particular end effector position $\boldsymbol{X}$ — are not so straightforward and, for many more complex robot arms, a closed-form solution does not exist. In this assignment, we will solve the inverse kinematic problem for a two degree-of-freedom, planar robot arm by solving numerically for Q_1 and Q_2 from the set of nonlinear Equations (29.1) .

Given a trajectory of data vectors $\boldsymbol{X}_{(i)}$, $1 \leq i \leq p$ — a sequence of p desired end effector positions — the corresponding joint angles satisfy

$$\boldsymbol{F}\big(\boldsymbol{Q}_{(i)}, \boldsymbol{X}_{(i)}\big) = 0, \quad 1 \leq i \leq p\ , \tag{29.2}$$

where

$$\boldsymbol{F}(\boldsymbol{q}, \boldsymbol{X}) = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} L_1\cos(q_1) + L_2\cos(q_1+q_2) - X_1 \\ L_1\sin(q_1) + L_2\sin(q_1+q_2) - X_2 \end{bmatrix}.$$

For the robot "home" position, $\boldsymbol{X}^{\text{home}} \approx [-0.7154, 6.9635]^{\mathrm{T}}$, the joint angles are known: $\boldsymbol{Q}^{\text{home}} = [1.6, 0.17]^{\mathrm{T}}$ (in radians). We shall assume that $\boldsymbol{X}_{(1)} = \boldsymbol{X}^{\text{home}}$ and hence $\boldsymbol{Q}_{(1)} = \boldsymbol{Q}^{\text{home}}$ in all cases; it will remain to find $\boldsymbol{Q}_{(2)}, \ldots, \boldsymbol{Q}_{(p)}$.

Based on the design of our robot, we impose the following physical constraints on Q_1 and Q_2:

$$\sin(Q_1) \geq 0\,; \qquad \sin(Q_2) \geq 0\ . \tag{29.3}$$

Note that a mathematically valid solution of Equation (29.2) might not satisfy the constraints of Equation (29.3) and, therefore, will need to be checked for physical consistency.

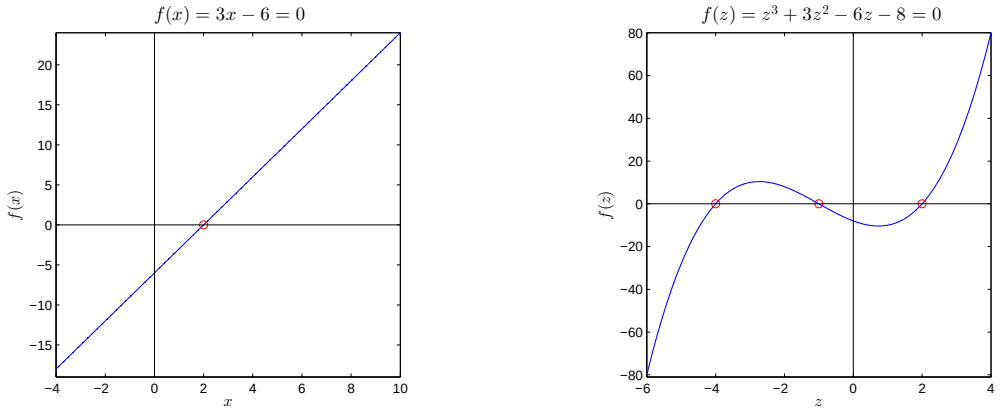
Previously we considered solving equations of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ for solution $\boldsymbol{X}$, given appropriately sized matrix and vector $\boldsymbol{A}$ and $\boldsymbol{b}$. In the univariate case with scalar (1×1) A and b, we could visualize the solution of these *linear* systems of equations as finding the zero crossing (root) of the line $f(x) = Ax - b$, as shown in Figure 29.3(a).

Now we will consider the solution of *nonlinear* systems of equations $\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{0}$ for root $\boldsymbol{Z}$, where terms such as powers of $\boldsymbol{z}$, transcendental functions of $\boldsymbol{z}$, discontinuous functions of $\boldsymbol{z}$, or any other such nonlinearities preclude the linear model. In the univariate case, we can visualize the solution of the nonlinear system as finding the roots of a nonlinear function, as shown in Figure 29.3(b) for a cubic, $f(z)$. Our robot example of Figures 29.1 and 29.2 represent a bivariate example of a nonlinear system (in which $\boldsymbol{F}$ plays the role of $\boldsymbol{f}$, and $\boldsymbol{Q}$ plays the role of $\boldsymbol{Z}$ — the root we wish to find).

In general, a linear system of equations may have no solution, one solution (a unique solution), or an infinite family of solutions. In contrast, a nonlinear problem may have no solution, one solution, two solutions, three solutions (as in Figure 29.3(b)), *any* number of solutions, or an infinite family of solutions. We will also need to decide which solutions (of a nonlinear problem) are of interest or relevant given the context and also stability considerations.

The nonlinear problem is typically solved as a sequence of linear problems — hence builds directly on linear algebra. The sequence of linear problems can be generated in a variety of fashions; our focus here is Newton's method. Many other approaches are also possible — for example, least squares/optimization — for solution of nonlinear problems.

The fundamental approach of Newton's method is simple:



(a) linear function

(b) nonlinear function

Figure 29.3: Solutions of univariate linear and nonlinear equations.

- Start with an initial guess or approximation for the root of a nonlinear system.
- Linearize the system around that initial approximation and solve the resulting *linear* system to find a better approximation for the root.
- Continue linearizing and solving until satisfied with the accuracy of the approximation.

This approach is identical for both univariate (one equation in one variable) and multivariate (n equations in n variables) systems. We will first consider the univariate case and then extend our analysis to the multivariate case.

29.2 Univariate Newton

29.2.1 The Method

Given a univariate function $f(z)$, we wish to find a *real* zero/root Z of f such that $f(Z) = 0$. Note that z is any real value (for which the function is defined), whereas Z is a particular value of z at which $f(z = Z)$ is *zero*; in other words, Z is a *root* of $f(z)$.

We first start with an initial approximation (guess) $\hat{z}^0$ for the zero Z. We next approximate the function $f(z)$ with its first-order Taylor series expansion around $\hat{z}^0$, which is the line tangent to $f(z)$ at $z = \hat{z}^0$

$$f^0_{\text{linear}}(z) \equiv f'(\hat{z}^0)(z - \hat{z}^0) + f(\hat{z}^0) \ . \tag{29.4}$$

We find the zero $\hat{z}^1$ of the linearized system $f^0_{\text{linear}}(z)$ by

$$f^0_{\text{linear}}(\hat{z}^1) \equiv f'(\hat{z}^0)(\hat{z}^1 - \hat{z}^0) + f(\hat{z}^0) = 0 \ , \tag{29.5}$$

which yields

$$\hat{z}^1 = \hat{z}^0 - \frac{f(\hat{z}^0)}{f'(\hat{z}^0)} \ . \tag{29.6}$$

We then repeat the procedure with $\hat{z}^1$ to find $\hat{z}^2$ and so on, finding successively better approximations to the zero of the original system $f(z)$ from our linear approximations $f^k_{\text{linear}}(z)$ until we reach $\hat{z}^N$ such that $|f(\hat{z}^N)|$ is within some desired tolerance of zero. (To be more rigorous, we must relate $|f(\hat{z}^N)|$ to $|Z - \hat{z}^N|$.)

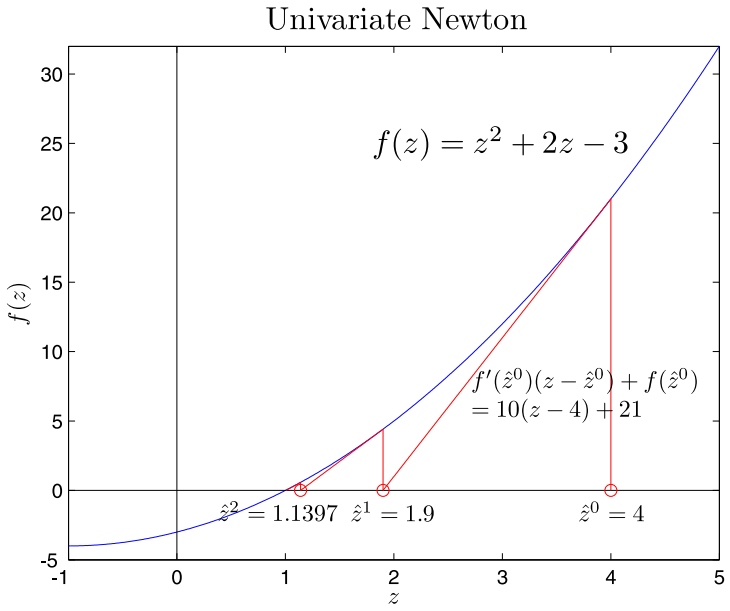


Figure 29.4: Graphical illustration of the Newton root finding method.

29.2.2 An Example

The first few steps of the following example have been illustrated in Figure 29.4.

We wish to use the Newton method to find a solution to the equation

$$z^2 + 2z = 3 \ . \tag{29.7}$$

We begin by converting the problem to a root-finding problem

$$f(Z) = Z^2 + 2Z - 3 = 0 \ . \tag{29.8}$$

We next note that the derivative of f is given by

$$f'(z) = 2z + 2 \ . \tag{29.9}$$

We start with initial guess $\hat{z}^0 = 4$. We then linearize around $\hat{z}^0$ to get

$$f^0_{\text{linear}}(z) \equiv f'(\hat{z}^0)(z - \hat{z}^0) + f(\hat{z}^0) = 10(z - 4) + 21 \ . \tag{29.10}$$

We solve the linearized system

$$f^0_{\text{linear}}(\hat{z}^1) \equiv 10(z - 4) + 21 = 0 \tag{29.11}$$

to find the next approximation for the root of $f(z)$,

$$\hat{z}^1 = 1.9 \ . \tag{29.12}$$

We repeat the procedure to find

$$f^1_{\text{linear}}(z) \equiv f'(\hat{z}^1)(z - \hat{z}^1) + f(\hat{z}^1) = 5.8(z - 1.9) + 4.41 \tag{29.13}$$

$$f^1_{\text{linear}}(\hat{z}^2) = 0 \tag{29.14}$$

$$\hat{z}^2 = 1.1397 \ ; \tag{29.15}$$

$$f^2_{\text{linear}}(z) \equiv f'(\hat{z}^2)(z - \hat{z}^2) + f(\hat{z}^2) = 4.2793(z - 1.1397) + 0.5781 \tag{29.16}$$

$$f^2_{\text{linear}}(\hat{z}^3) = 0 \tag{29.17}$$

$$\hat{z}^3 = 1.0046 \ . \tag{29.18}$$

Note the rapid convergence to the actual root $Z = 1$. Within three iterations the error has been reduced to 0.15% of its original value.

29.2.3 The Algorithm

The algorithm for the univariate Newton's method is a simple **while** loop. If we wish to store the intermediate approximations for the zero, the algorithm is shown in Algorithm 1. If we don't need to save the intermediate approximations, the algorithm is shown in Algorithm 2.

If, for some reason, we cannot compute the derivative $f'(\hat{z}^k)$ directly, we can substitute the finite difference approximation of the derivative (see Chapter 3) for some arbitrary (small) given Δz, which, for the backward difference is given by

$$f'(\hat{z}^k) \approx \frac{f(\hat{z}^k) - f(\hat{z}^k - \Delta z)}{\Delta z} \ . \tag{29.19}$$

In practice, for Δz sufficiently small (but not too small — round-off), Newton with approximate derivative will behave similarly to Newton with exact derivative.

Algorithm 1 Newton algorithm with storage of intermediate approximations

$k \leftarrow 0$
while $\left|f(\hat{z}^k)\right| > \text{tol}$ **do**
  $\hat{z}^{k+1} \leftarrow \hat{z}^k - \frac{f(\hat{z}^k)}{f'(\hat{z}^k)}$
  $k \leftarrow k + 1$
end while
$Z \leftarrow \hat{z}^k$

Algorithm 2 Newton algorithm without storage

$\hat{z} \leftarrow \hat{z}^0$
while $\left|f(\hat{z})\right| > \text{tol}$ **do**
  $\delta\hat{z} \leftarrow \frac{-f(\hat{z})}{f'(\hat{z})}$
  $\hat{z} \leftarrow \hat{z} + \delta\hat{z}$
end while
$Z \leftarrow \hat{z}$

There also exists a method, based on Newton iteration, that directly incorporates a finite difference approximation of the derivative by using the function values at the two previous iterations

(thus requiring two initial guesses) to construct

$$f'(\hat{z}^k) \approx \frac{f(\hat{z}^k) - f(\hat{z}^{k-1})}{\hat{z}^k - \hat{z}^{k-1}} \ . \tag{29.20}$$

This is called the *secant* method because the linear approximation of the function is no longer a line tangent to the function, but a secant line. This method works well with one variable (with a modest reduction in convergence rate); the generalization of the secant method to the multivariate case (and quasi-Newton methods) is more advanced.

Another root-finding method that works well (although slowly) in the univariate case is the bisection (or "binary chop") method. The bisection method finds the root within a given interval by dividing the interval in half on each iteration and keeping only the half whose function evaluations at its endpoints are of opposite sign — and which, therefore, must contain the root. This method is very simple and robust — it works even for non-smooth functions — but, because it fails to exploit any information regarding the derivative of the function, it is slow. It also cannot be generalized to the multivariate case.

29.2.4 Convergence Rate

When Newton works, it works extremely fast. More precisely, if we denote the error in Newton's approximation for the root at the k^{th} iteration as

$$\epsilon^k = \hat{z}^k - Z, \tag{29.21}$$

then if

(*i*) $f(z)$ is smooth (e.g., the second derivative exists),

(*ii*) $f'(Z) \neq 0$ (i.e., the derivative at the root is nonzero), and

(*iii*) ϵ^0 (the error of our initial guess) is sufficiently small,

we can show that we achieve *quadratic* (i.e., $\epsilon^{k+1} \sim (\epsilon^k)^2$) convergence:

$$\epsilon^{k+1} \sim (\epsilon^k)^2 \left(\frac{1}{2} \frac{f''(Z)}{f'(Z)} \right) \ . \tag{29.22}$$

Each iteration *doubles* the number of correct digits; this is extremely fast convergence. For our previous example, the sequence of approximations obtained is shown in Table 29.1. Note that the doubling of correct digits applies only once we have at least one correct digit. For the secant method, the convergence rate is slightly slower:

$$\epsilon^{k+1} \sim (\epsilon^k)^\gamma \ , \tag{29.23}$$

where $\gamma \approx 1.618$.

Proof. A sketch of the proof for Newton's convergence rate is shown below:

iteration	approximation	number of correct digits
0	4	0
1	1.9	1
2	1.1...	1
3	1.004...	3
4	1.000005...	6
5	1.000000000006...	12

Table 29.1: Convergence of Newton for the example problem.

$$\hat{z}^{k+1} = \hat{z}^k - \frac{f(\hat{z}^k)}{f'(\hat{z}^k)} \tag{29.24}$$

$$\cancel{Z} + \epsilon^{k+1} = \cancel{Z} + \epsilon^k - \frac{f(Z+\epsilon^k)}{f'(Z+\epsilon^k)} \tag{29.25}$$

$$\epsilon^{k+1} = \epsilon^k - \frac{\cancel{f(Z)} + \epsilon^k f'(Z) + \frac{1}{2}(\epsilon^k)^2 f''(Z) + \cdots}{f'(Z) + \epsilon^k f''(Z) + \cdots} \tag{29.26}$$

$$\epsilon^{k+1} = \epsilon^k - \epsilon^k \frac{f'(Z) + \frac{1}{2}\epsilon^k f''(Z) + \cdots}{f'(Z)(1 + \epsilon^k \frac{f''(Z)}{f'(Z)} + \cdots)} \tag{29.27}$$

$$\epsilon^{k+1} = \epsilon^k - \epsilon^k \frac{\cancel{f'(Z)}(1 + \frac{1}{2}\epsilon^k \frac{f''(Z)}{f'(Z)} + \cdots}{\cancel{f'(Z)}(1 + \epsilon^k \frac{f''(Z)}{f'(Z)} + \cdots)} \; ; \tag{29.28}$$

since $\frac{1}{1+\rho} \sim 1 - \rho + \cdots$ for small ρ,

$$\epsilon^{k+1} = \epsilon^k - \epsilon^k \left(1 + \frac{1}{2}\epsilon^k \frac{f''(Z}{f'(Z)}\right)\left(1 - \epsilon^k \frac{f''(Z)}{f'(Z)}\right) + \cdots \tag{29.29}$$

$$\epsilon^{k+1} = \cancel{\epsilon^k} - \cancel{\epsilon^k} + \frac{1}{2}(\epsilon^k)^2 \frac{f''(Z)}{f'(Z)} + \cdots \tag{29.30}$$

$$\epsilon^{k+1} = \frac{1}{2}\frac{f''(Z)}{f'(Z)}(\epsilon^k)^2 + \cdots \; . \tag{29.31}$$

We thus confirm the quadratic convergence. □

Note that, if $f'(Z) = 0$, we must stop at equation (29.26) to obtain the *linear* (i.e., $\epsilon^{k+1} \sim \epsilon^k$) convergence rate

$$\epsilon^{k+1} = \epsilon^k - \frac{\frac{1}{2}(\epsilon^k)^2 f''(Z) + \cdots}{\epsilon^k f''(Z)} \tag{29.32}$$

$$\epsilon^{k+1} = \frac{1}{2}\epsilon^k + \cdots \; . \tag{29.33}$$

In this case, we gain only a *constant* number of correct digits after each iteration. The bisection method also displays linear convergence.

29.2.5 Newton Pathologies

Although Newton often does work well and very fast, we must always be careful not to excite pathological (i.e., atypically bad) behavior through our choice of initial guess or through the nonlinear function itself.

For example, we can easily — and, thus, this might be less pathology than generally bad behavior — arrive at an "incorrect" solution with Newton if our initial guess is poor. For instance, in our earlier example of Figure 29.4, say that we are interested in finding a positive root, $Z > 0$, of $f(z)$. If we had chosen an initial guess of $\hat{z}^0 = -4$ instead of $\hat{z}^0 = +4$, we would have (deservingly) found the root $Z = -3$ instead of $Z = 1$. Although in the univariate case we can often avoid this behavior by basing our initial guess on a prior inspection of the function, this approach becomes more difficult in higher dimensions.

Even more diabolical behavior is possible. If the nonlinear function has a local maximum or minimum, it is often possible to excite oscillatory behavior in Newton through our choice of initial guess. For example, if the linear approximations at two points on the function both return the other point as their solution, Newton will oscillate between the two indefinitely and never converge to any roots. Similarly, if the first derivative is not well behaved in the vicinity of the root, then Newton may diverge to infinity.

We will later address these issues (when possible) by continuation and homotopy methods.

29.3 Multivariate Newton

29.3.1 A Model Problem

Now we will apply the Newton method to solve multivariate nonlinear systems of equations. For example, we can consider the simple *bivariate* system of nonlinear equations

$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 = 0 \,, \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 = 0 \,. \end{aligned} \tag{29.34}$$

Note that f_1 and f_2 are the two paraboloids each with principal axes aligned with the coordinate directions; we plot f_1 and f_2 in shown in Figure 29.5. We wish to find a zero $\boldsymbol{Z} = (z_1 \;\; z_2)^{\mathrm{T}}$ of $\boldsymbol{f}(\boldsymbol{z}) = (f_1(z_1, z_2) \;\; f_2(z_1, z_2))^{\mathrm{T}}$ such that $\boldsymbol{f}(\boldsymbol{Z}) = \boldsymbol{0}$. We can visualize $\boldsymbol{Z}$ as the intersections of the ellipses $f_1 = 0$ (the intersection of paraboloid f_1 with the zero plane) and $f_2 = 0$ (the intersection of paraboloid f_2 with the zero plane). The four solutions $(Z_1, Z_2) = (\pm 2, \pm 3)$ are shown as the red circles in the contour plot of Figure 29.6.

29.3.2 The Method

The Newton method for the multivariate case follows the same approach as for the univariate case:

- Start with an initial approximation $\hat{\boldsymbol{z}}^0$ of a root to the nonlinear system $\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{0}$.
- Linearize the system at $\hat{\boldsymbol{z}}^0$ and solve the resulting linear system to derive a better approximation $\hat{\boldsymbol{z}}^1$.
- Continue linearizing and solving until the norm of $\boldsymbol{f}(\hat{\boldsymbol{z}}^N)$ is within some desired tolerance of zero.

However, the multivariate case can be much more challenging computationally.

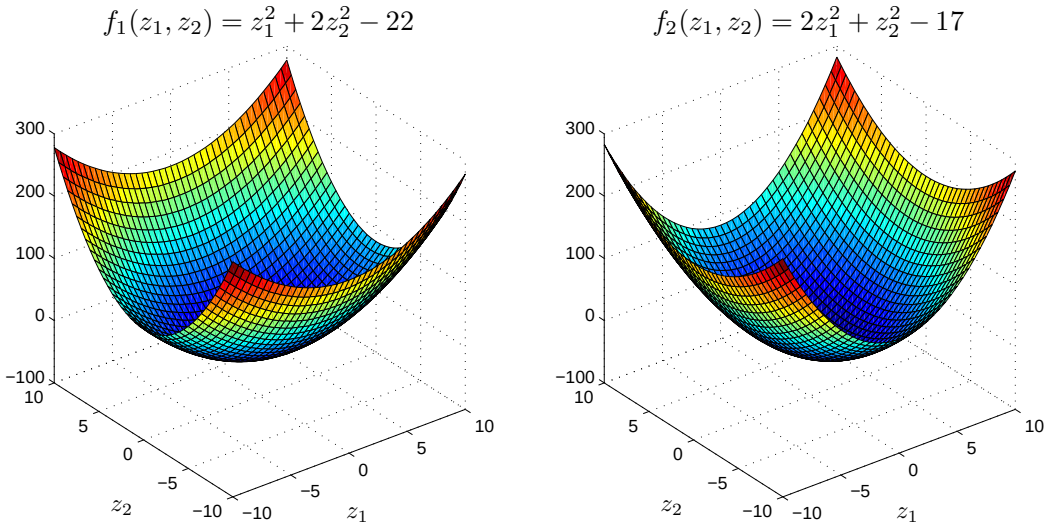


Figure 29.5: Elliptic Paraboloids f_1 and f_2.

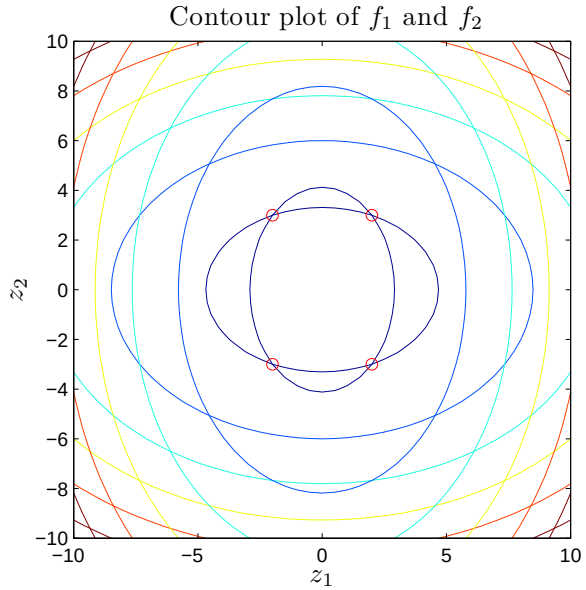


Figure 29.6: Contour plots of f_1 and f_2 with their intersections of the zero contours (the solutions to $\boldsymbol{f}(\boldsymbol{z}) = \mathbf{0}$) shown as red circles.

We present the method for the general case in which $\boldsymbol{z}$ is an n-vector, $(z_1 \;\; z_2 \;\; \cdots \;\; z_n)^{\mathrm{T}}$ and $\boldsymbol{f}(\boldsymbol{z})$ is also an n-vector, $(f_1(\boldsymbol{z}) \;\; f_2(\boldsymbol{z}) \;\; \cdot \;\; f_n(\boldsymbol{z}))^{\mathrm{T}}$. This represents n *nonlinear* equations in n unknowns. (Of course, even more generally, we could consider more equations than unknowns or less equations than unknowns.) To linearize the multivariate system, we again use a first-order Taylor series expansion, which, for a single multivariate function linearized at the point $\hat{\boldsymbol{z}}^k$, is given by

$$f^k_{\text{linear}}(\boldsymbol{z}) \equiv \left.\frac{\partial f}{\partial z_1}\right|_{\hat{\boldsymbol{z}}^k} (z_1 - \hat{z}^k_1) + \left.\frac{\partial f}{\partial z_2}\right|_{\hat{\boldsymbol{z}}^k} (z_2 - \hat{z}^k_2) + \cdots + \left.\frac{\partial f}{\partial z_n}\right|_{\hat{\boldsymbol{z}}^k} (z_n - \hat{z}^k_n) + f(\hat{\boldsymbol{z}}^k) \tag{29.35}$$

(cf. equation (29.4) in the univariate case). Because we have n equations in n variables, we must linearize each of the equations $(f_1(\boldsymbol{z}) \;\; f_2(\boldsymbol{z}) \;\; \ldots \;\; f_n(\boldsymbol{z}))^{\mathrm{T}}$ and then, per the Newton recipe, set $(f_1(\hat{\boldsymbol{z}}^k) = 0 \;\; \ldots \;\; f_n(\hat{\boldsymbol{z}}^k) = 0)^{\mathrm{T}}$. Our full linearized system looks like

$$f^k_{1,\text{linear}}(\hat{\boldsymbol{z}}^{k+1}) \equiv \left.\frac{\partial f_1}{\partial z_1}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_1 - \hat{z}^k_1) + \left.\frac{\partial f_1}{\partial z_2}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_2 - \hat{z}^k_2) + \cdots + \left.\frac{\partial f_1}{\partial z_n}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_n - \hat{z}^k_n) + f_1(\hat{\boldsymbol{z}}^k) = 0 \;, \tag{29.36}$$

$$f^k_{2,\text{linear}}(\hat{\boldsymbol{z}}^{k+1}) \equiv \left.\frac{\partial f_2}{\partial z_1}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_1 - \hat{z}^k_1) + \left.\frac{\partial f_2}{\partial z_2}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_2 - \hat{z}^k_2) + \cdots + \left.\frac{\partial f_2}{\partial z_n}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_n - \hat{z}^k_n) + f_2(\hat{\boldsymbol{z}}^k) = 0 \;, \tag{29.37}$$

up through

$$f^k_{n,\text{linear}}(\hat{\boldsymbol{z}}^{k+1}) \equiv \left.\frac{\partial f_n}{\partial z_1}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_1 - \hat{z}^k_1) + \left.\frac{\partial f_n}{\partial z_2}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_2 - \hat{z}^k_2) + \cdots + \left.\frac{\partial f_n}{\partial z_n}\right|_{\hat{\boldsymbol{z}}^k} (\hat{z}^{k+1}_n - \hat{z}^k_n) + f_n(\hat{\boldsymbol{z}}^k) = 0 \tag{29.38}$$

(cf. equation (29.5) in the univariate case).

We can write the *linear* system of equations (29.36)–(29.38) in matrix form as

$$\boldsymbol{f}^k_{\text{linear}}(\hat{\boldsymbol{z}}^{k+1}) \equiv \left.\begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \cdots & \frac{\partial f_1}{\partial z_n} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \cdots & \frac{\partial f_2}{\partial z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z_1} & \frac{\partial f_n}{\partial z_2} & \cdots & \frac{\partial f_n}{\partial z_n} \end{bmatrix}\right|_{\hat{\boldsymbol{z}}^k} \begin{bmatrix} (\hat{z}^{k+1}_1 - \hat{z}^k_1) \\ (\hat{z}^{k+1}_2 - \hat{z}^k_2) \\ \vdots \\ (\hat{z}^{k+1}_n - \hat{z}^k_n) \end{bmatrix} + \begin{bmatrix} f_1(\hat{z}^k) \\ f_2(\hat{z}^k) \\ \vdots \\ f_n(\hat{z}^k) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{29.39}$$

or, equivalently,

$$\boldsymbol{J}(\hat{\boldsymbol{z}}^k)\delta\hat{\boldsymbol{z}}^k = -\boldsymbol{f}(\hat{\boldsymbol{z}}^k) \;. \tag{29.40}$$

Here the $n \times n$ *Jacobian* matrix $\boldsymbol{J}$ is defined as the matrix of all first-order partial derivatives of the function vector $(f_1 \;\; \ldots \;\; f_n)^{\mathrm{T}}$ with respect to the state vector $(z_1 \;\; \ldots \;\; z_n)^{\mathrm{T}}$:

$$\boldsymbol{J}(\boldsymbol{z}) = \left.\begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \cdots & \frac{\partial f_1}{\partial z_n} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \cdots & \frac{\partial f_2}{\partial z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z_1} & \frac{\partial f_n}{\partial z_2} & \cdots & \frac{\partial f_n}{\partial z_n} \end{bmatrix}\right|_{\boldsymbol{z}}, \tag{29.41}$$

such that the i, j^{th} component of the Jacobian corresponds to the partial derivative of the i^{th} function with respect to the j^{th} variable,

$$J_{ij}(\boldsymbol{z}) = \frac{\partial f_i}{\partial z_j}(\boldsymbol{z}) \ . \tag{29.42}$$

Thus $\boldsymbol{J}(\hat{\boldsymbol{z}}^k)$ denotes the Jacobian matrix for the system evaluated at the point $\hat{\boldsymbol{z}}^k$. Note also that $\delta\hat{\boldsymbol{z}}^k$ is the displacement vector pointing from the current approximation $\hat{\boldsymbol{z}}^k$ to the next approximation $\hat{\boldsymbol{z}}^{k+1}$

$$\delta\hat{\boldsymbol{z}}^k = \begin{bmatrix} (\hat{z}_1^{k+1} - \hat{z}_1^k) \\ (\hat{z}_2^{k+1} - \hat{z}_2^k) \\ \vdots \\ (\hat{z}_n^{k+1} - \hat{z}_n^k) \end{bmatrix} . \tag{29.43}$$

Hence $\delta\hat{\boldsymbol{z}}^k$ is the Newton update to our current iterate.

29.3.3 An Example

We can now apply Newton to solve the $n = 2$ model problem described in Section 29.3.1:

$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 = 0 \ , \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 = 0 \ . \end{aligned} \tag{29.44}$$

We first compute the elements of the Jacobian matrix as

$$J_{11} = \frac{\partial f_1}{\partial z_1} = 2z_1, \quad J_{12} = \frac{\partial f_1}{\partial z_2} = 4z_2, \quad J_{21} = \frac{\partial f_2}{\partial z_1} = 4z_1, \quad J_{22} = \frac{\partial f_2}{\partial z_2} = 2z_2 \ , \tag{29.45}$$

and write the full matrix as

$$\boldsymbol{J}(\boldsymbol{z}) = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} 2z_1 & 4z_2 \\ 4z_1 & 2z_2 \end{bmatrix} . \tag{29.46}$$

We can now perform the iteration.

We start with initial guess $\hat{\boldsymbol{z}}^0 = (10 \ \ 10)^{\text{T}}$. We next linearize around $\hat{\boldsymbol{z}}^0$ to get

$$\boldsymbol{f}^0_{\text{linear}}(\boldsymbol{z}) \equiv \boldsymbol{J}(\hat{\boldsymbol{z}}^0)(\boldsymbol{z} - \hat{\boldsymbol{z}}^0) + \boldsymbol{f}(\hat{\boldsymbol{z}}^0) = \begin{bmatrix} 20 & 40 \\ 40 & 20 \end{bmatrix} \delta\hat{\boldsymbol{z}}^0 + \begin{bmatrix} 278 \\ 283 \end{bmatrix} . \tag{29.47}$$

Note that we can visualize this system as two planes tangent to f_1 and f_2 at $\hat{\boldsymbol{z}}^0$. We now solve the linearized system

$$\boldsymbol{f}^0_{\text{linear}}(\hat{\boldsymbol{z}}^1) \equiv \begin{bmatrix} 20 & 40 \\ 40 & 20 \end{bmatrix} \delta\hat{\boldsymbol{z}}^0 + \begin{bmatrix} 278 \\ 283 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{29.48}$$

or

$$\begin{bmatrix} 20 & 40 \\ 40 & 20 \end{bmatrix} \delta\boldsymbol{z}^0 = \begin{bmatrix} -278 \\ -283 \end{bmatrix} \tag{29.49}$$

to find

$$\delta\hat{\boldsymbol{z}}^0 = \begin{bmatrix} -4.8 \\ -4.55 \end{bmatrix}; \tag{29.50}$$

thus the next approximation for the root of $\boldsymbol{f}(z)$ is given by

$$\hat{\boldsymbol{z}}^1 = \hat{\boldsymbol{z}}^0 + \delta\hat{\boldsymbol{z}}^0 = \begin{bmatrix} 5.2 \\ 5.45 \end{bmatrix}. \tag{29.51}$$

We repeat the procedure to find

$$\boldsymbol{f}^1_{\text{linear}}(\boldsymbol{z}) \equiv \boldsymbol{J}(\hat{\boldsymbol{z}}^1)(\boldsymbol{z} - \hat{\boldsymbol{z}}^1) + \boldsymbol{f}(\hat{\boldsymbol{z}}^1) = \begin{bmatrix} 10.4 & 21.8 \\ 20.8 & 10.9 \end{bmatrix} \delta\hat{\boldsymbol{z}}^1 + \begin{bmatrix} 64.445 \\ 66.7825 \end{bmatrix}, \tag{29.52}$$

$$\boldsymbol{f}^1_{\text{linear}}(\hat{\boldsymbol{z}}^2) = \boldsymbol{0} \tag{29.53}$$

$$\hat{\boldsymbol{z}}^2 = \begin{bmatrix} 2.9846 \\ 3.5507 \end{bmatrix}; \tag{29.54}$$

$$\boldsymbol{f}^2_{\text{linear}}(\boldsymbol{z}) =\equiv \boldsymbol{J}(\hat{\boldsymbol{z}}^2)(\boldsymbol{z} - \hat{\boldsymbol{z}}^2) + \boldsymbol{f}(\hat{\boldsymbol{z}}^2) = \begin{bmatrix} 5.9692 & 14.2028 \\ 11.9385 & 7.1014 \end{bmatrix} \delta\hat{\boldsymbol{z}}^2 + \begin{bmatrix} 12.1227 \\ 13.4232 \end{bmatrix}, \tag{29.55}$$

$$\boldsymbol{f}^2_{\text{linear}}(\hat{\boldsymbol{z}}^3) = \boldsymbol{0} \tag{29.56}$$

$$\hat{\boldsymbol{z}}^3 = \begin{bmatrix} 2.1624 \\ 3.0427 \end{bmatrix}. \tag{29.57}$$

We see that the solution rapidly approaches the (nearest) exact solution $\boldsymbol{Z} = (2 \;\; 3)^{\mathrm{T}}$.

29.3.4 The Algorithm

The multivariate Newton algorithm is identical to the univariate algorithm, except that now for each pass through the **while** loop we must now solve a linearized *system* of equations involving the Jacobian matrix.

Algorithm 3 Multivariate Newton algorithm without storage

$\hat{\boldsymbol{z}} \leftarrow \hat{\boldsymbol{z}}^0$
while $\|\boldsymbol{f}(\hat{\boldsymbol{z}})\| > \text{tol}$ **do**
  {Solve the linearized system for $\delta\hat{\boldsymbol{z}}$.}
  $\boldsymbol{J}(\hat{\boldsymbol{z}})\delta\hat{\boldsymbol{z}} = -\boldsymbol{f}(\hat{\boldsymbol{z}})$
  $\hat{\boldsymbol{z}} \leftarrow \hat{\boldsymbol{z}} + \delta\hat{\boldsymbol{z}}$
end while
$\boldsymbol{Z} \leftarrow \hat{\boldsymbol{z}}$

We can see that the computational cost for the multivariate Newton iteration is essentially the total number of iterations multiplied by the cost to solve an $n \times n$ linear system — which, if dense,

could be as much as $\mathcal{O}(n^3)$ and, if sparse, as little as $\mathcal{O}(n)$ operations. Additionally, for "pure" Newton, the Jacobian needs to be recomputed ($\mathcal{O}(n^2)$ operations) at each iteration. (In "impure" Newton the Jacobian sometimes is held fixed for several iterations or updated selectively.)

Note that, as before in the univariate case, we can substitute a finite difference approximation for the Jacobian if we can not (or choose not to) use the analytical expressions. Here we first introduce a scalar Δz (small); note that Δz is not related to $\delta \boldsymbol{z}$ — i.e., this is not a secant approach. Then we approximate, for $1 \leq i \leq n$, $1 \leq j \leq n$,

$$J_{ij}(\boldsymbol{z}) \equiv \frac{\partial f_i}{\partial z_j}(\boldsymbol{z}) \approx \frac{f_i(\boldsymbol{z} + \Delta z \boldsymbol{e}^j) - f_i(\boldsymbol{z})}{\Delta z} \equiv \widetilde{J}_{ij}(\boldsymbol{z}), \tag{29.58}$$

where $\boldsymbol{e}^j$ is the unit vector in the j-direction such that $\boldsymbol{z} + \Delta z \boldsymbol{e}^j$ differs from $\boldsymbol{z}$ in only the j^{th} component — a partial difference approximation to the partial derivative $\frac{\partial f_i}{\partial z_j}$. Note that to compute the full finite difference approximation $n \times n$ matrix $\widetilde{\boldsymbol{J}}(\boldsymbol{z})$ we need $\mathcal{O}(n^2)$ function evaluations.

29.3.5 Comments on Multivariate Newton

For multivariate Newton, both the convergence rate and the pathologies are similar to the univariate case, although the pathologies are somewhat more likely in the multivariate case given the greater number of degrees of freedom.

The main difference for the multivariate case, then, is the relative cost of each Newton iteration (worst case $\mathcal{O}(n^3)$ operations) owing to the size of the linear system which must be solved. For this reason, Newton's rapid convergence becomes more crucial with growing dimensionality. Thanks to the rapid convergence, Newton often outperforms "simpler" approaches which are less computationally expensive per iteration but require many more iterations to converge to a solution.

29.4 Continuation and Homotopy

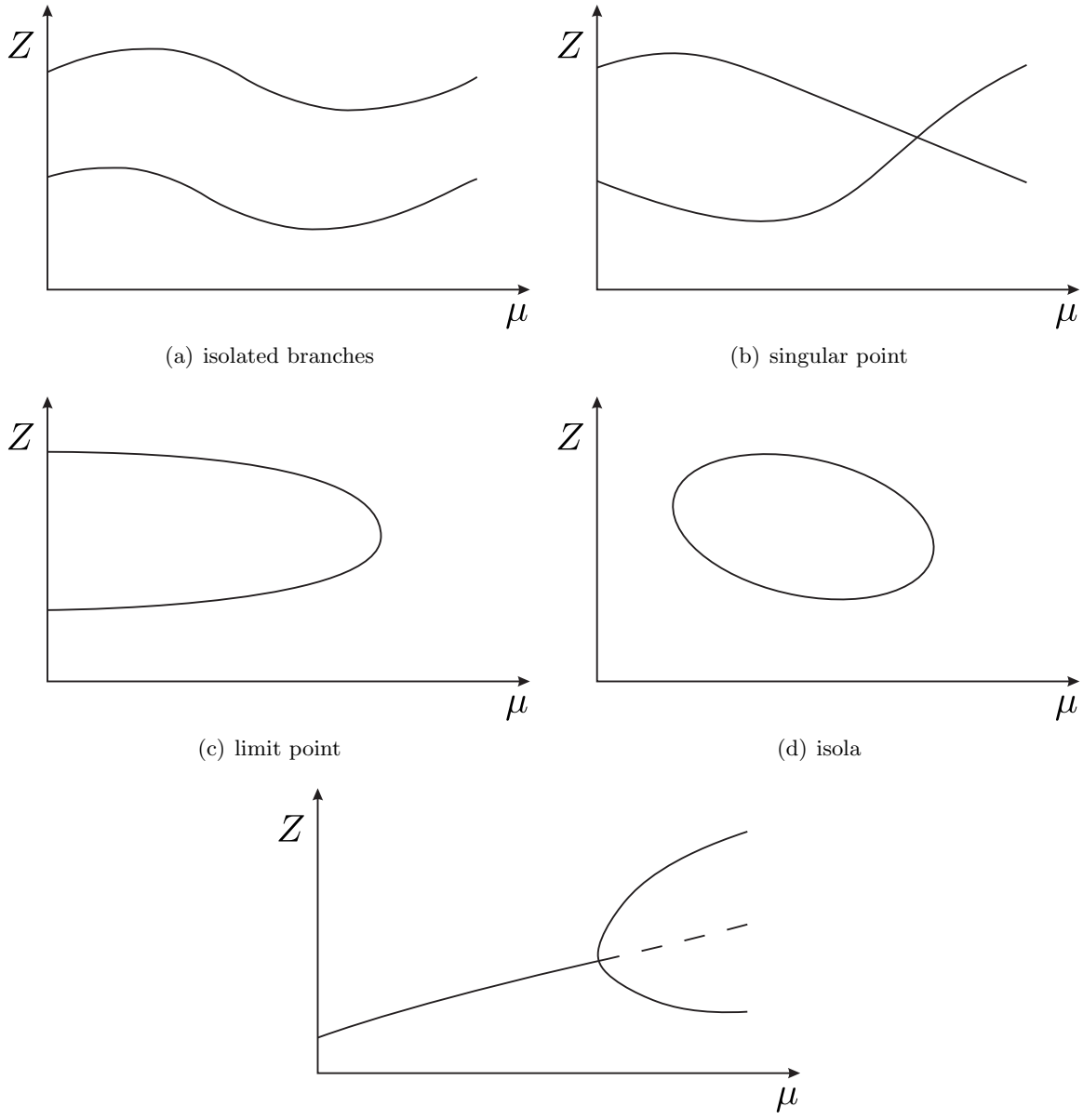
Often we are interested not in solving just a single nonlinear problem, but rather a family of nonlinear problems $\boldsymbol{f}(\boldsymbol{Z}; \mu) = \boldsymbol{0}$ with real parameter μ which can take on a sequence of values $\mu_{(1)}, \ldots, \mu_{(p)}$. Typically we supplement $\boldsymbol{f}(\boldsymbol{Z}; \mu) = \boldsymbol{0}$ with some simple constraints or continuity conditions which select a particular solution from several possible solutions.

We then wish to ensure that, (*i*) we are able to start out at all, often by transforming the initial problem for $\mu = \mu_{(1)}$ (and perhaps also subsequent problems for $\mu = \mu_{(i)}$) into a series of simpler problems (homotopy) and, (*ii*) once started, and as the parameter μ is varied, we continue to converge to "correct" solutions as defined by our constraints and continuity conditions (continuation).

29.4.1 Parametrized Nonlinear Problems: A Single Parameter

Given $\boldsymbol{f}(\boldsymbol{Z}; \mu) = 0$ with real single parameter μ, we are typically interested in how our solution $\boldsymbol{Z}$ changes as we change μ; in particular, we now interpret (*à la* the implicit function theorem) $\boldsymbol{Z}$ as a function $\boldsymbol{Z}(\mu)$. We can visualize this dependency by plotting Z (here $n = 1$) with respect to μ, giving us a *bifurcation diagram* of the problem, as depicted in Figure 29.7 for several common modes of behavior.

Figure 29.7(a) depicts two isolated solution branches with two, distinct real solutions over the whole range of μ. Figure 29.7(b) depicts two solution branches that converge at a *singular point*, where a change in μ can drive the solution onto either of the two branches. Figure 29.7(c) depicts two solution branches that converge at a *limit point*, beyond which there is no solution.



(a) isolated branches

(b) singular point

(c) limit point

(d) isola

(e) pitchfork

Figure 29.7: Bifurcation Diagram: Several Common Modes of Behavior.

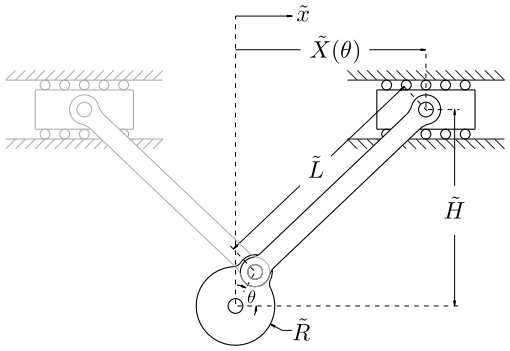


Figure 29.8: Simple mechanical linkage.

Figure 29.7(d) depicts an *isola*, an isolated interval of two solution branches with two limit points for endpoints. Figure 29.7(e) depicts a single solution branch that "bifurcates" into several solutions at a *pitchfork*; pitchfork bifurcations often correspond to nonlinear dynamics that can display either *stable* (represented by solid lines in the figure) or *unstable* (represented by dotted lines) behavior, where stability refers to the ability of the state variable $\boldsymbol{Z}$ to return to the solution when perturbed.

Note that, for all of the mentioned cases, when we reach a singular or limit point — characterized by the convergence of solution branches — the Jacobian becomes singular (non-invertible) and hence Newton breaks down unless supplemented by additional conditions.

29.4.2 A Simple Example

We can develop an intuitive understanding of these different modes of behavior and corresponding bifurcation diagrams by considering a simple example.

We wish to analyze the simple mechanical linkage shown in Figure 29.8 by finding $\widetilde{X}$ corresponding to an arbitrary θ for given (constant) $\widetilde{H}$, $\widetilde{R}$, and $\widetilde{L}$. In this example, then, θ corresponds to the earlier discussed generic parameter μ.

We can find an analytical solution for $\widetilde{X}(\theta; \widetilde{R}, \widetilde{H}, \widetilde{L})$ by solving the geometric constraint

$$(\widetilde{X} - \widetilde{R}\cos\theta)^2 + (\widetilde{H} - \widetilde{R}\sin\theta)^2 = \widetilde{L}^2 \ , \tag{29.59}$$

which defines the distance between the two joints as $\widetilde{L}$. This is clearly a nonlinear equation, owing to the quadratic term in $\tilde{x}$. We can eliminate one parameter from the equation by non-dimensionalizing with respect to $\widetilde{L}$, giving us

$$(X - R\cos\theta)^2 + (H - R\sin\theta)^2 = 1 \ , \tag{29.60}$$

where $X = \frac{\widetilde{X}}{\widetilde{L}}$, $R = \frac{\widetilde{R}}{\widetilde{L}}$, and $H = \frac{\widetilde{H}}{\widetilde{L}}$. Expanding and simplifying, we get

$$aX^2 + bX + c \equiv f(X; \theta; R, H) = 0 \ , \tag{29.61}$$

where $a = 1$, $b = -2R\cos\theta$, and $c = R^2 + H^2 - 2HR\sin\theta - 1$. A direct application of the quadratic formula then gives us the two roots

$$\begin{aligned} X_+ &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \ , \\ X_- &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} \ , \end{aligned} \tag{29.62}$$

which may be real or complex.

We observe three categories for the number of solutions to our quadratic equation depending on the value of the *discriminant* $\Delta(\theta; R, H) \equiv b^2 - 4ac$. First, if $\Delta < 0$, $\sqrt{\Delta}$ is imaginary and there is no (real) solution. An example is the case in which $\widetilde{H} > \widetilde{L} + \widetilde{R}$. Second, if $\Delta = 0$, there is exactly one solution, $X = \frac{-b}{2a}$. An example is the case in which $\widetilde{H} = \widetilde{L} + \widetilde{R}$ and $\theta = \frac{\pi}{2}$. Third, if $\Delta > 0$, there are two distinct solutions, X_+ and X_-; an example is shown in Figure 29.8. Note that with our simple crank example we can obtain all the cases of Figure 29.7 except Figure 29.7(e).

We note that the case of two distinct solutions is new — our linear systems of equations (for the univariate case, $f(x) = Ax - b$) had either no solution ($A = 0$, $b = 0$; line parallel to x axis), exactly one solution ($A = 0$; line intersecting x axis), or an infinite number of solutions ($A = 0$, $b = 0$; line on x axis). Nonlinear equations, on the other hand, have no such restrictions. They can have no solution, one solution, two solutions (as in our quadratic case above), three solutions (e.g., a cubic equation) — *any* finite number of solutions, depending on the nature of the particular function $f(z)$ — or an infinite number of solutions (e.g., a sinusoidal equation). For example, if $f(z)$ is an n^{th}-order polynomial, there could be anywhere from zero to n (real) solutions, depending on the values of the $n + 1$ parameters of the polynomial.

It is important to note, for the cases in which there are two, distinct solution branches corresponding to X_+ and X_-, that, as we change the θ of the crank, it would be physically impossible to jump from one branch to the other — unless we stopped the mechanism, physically disassembled it, and then reassembled it as a mirror image of itself. Thus for the physically relevant solution we must require a continuity condition, or equivalently a constraint that requires $|X(\theta_{(i)}) - X(\theta_{(i-1)})|$ not too large or perhaps $X(\theta_{(i)})X(\theta_{(i-1)}) > 0$; here $\theta_{(i)}$ and $\theta_{(i-1)}$ are successive parameters in our family of solutions.

In the Introduction, Section 29.1, we provide an example of a linkage with two degrees of freedom. In this robot arm example the parameter $\boldsymbol{\mu}$ is given by $\boldsymbol{X}$, the desired position of the end effector.

29.4.3 Path Following: Continuation

As already indicated, as we vary our parameter μ (corresponding to θ in the crank example), we must reflect any constraint (such as, in the crank example, no "re-assembly") in our numerical approach to ensure that the solution to which we converge is indeed the "correct" one. One approach is through an appropriate choice of initial guess. Inherent in this imperative is an opportunity — we can exploit information about our previously converged solution not only to keep us on the appropriate solution branch, but also to assist continued (rapid) convergence of the Newton iteration.

We denote our previously converged solution to $f(Z; \mu) = 0$ as $Z(\mu_{(i-1)})$ (we consider here the univariate case). We wish to choose an initial guess $\widehat{Z}(\mu_{(i)})$ to converge to (a nearby root) $Z(\mu_{(i)}) = Z(\mu_{(i-1)} + \delta\mu)$ for some step $\delta\mu$ in μ. The simplest approach is to use the previously converged solution itself as our initial guess for the next step,

$$\widehat{Z}(\mu_{(i)}) = Z(\mu_{(i-1)}) \ . \tag{29.63}$$

This is often sufficient for small changes $\delta\mu$ in μ and it is certainly the simplest approach.

We can improve our initial guess if we use our knowledge of the rate of change of $Z(\mu)$ with respect to μ to help us extrapolate, to wit

$$\widehat{Z}(\mu_{(i)}) = Z(\mu_{(i-1)}) + \frac{\mathrm{d}Z}{\mathrm{d}\mu}\delta\mu \ . \tag{29.64}$$

We can readily calculate $\frac{\mathrm{d}Z}{\mathrm{d}\mu}$ as

$$\frac{\mathrm{d}Z}{\mathrm{d}\mu} = \frac{-\frac{\partial f}{\partial \mu}}{\frac{\partial f}{\partial z}} \ , \tag{29.65}$$

since

$$f(Z(\mu);\mu) = 0 \Rightarrow \frac{\mathrm{d}f}{\mathrm{d}\mu}(Z(\mu);\mu) \equiv \frac{\partial f}{\partial z}\frac{\mathrm{d}Z}{\mathrm{d}\mu} + \frac{\partial f}{\partial \mu}\frac{\mathrm{d}\mu}{\mathrm{d}\mu}^{1} = 0 \Rightarrow \frac{\mathrm{d}Z}{\mathrm{d}\mu} = \frac{-\frac{\partial f}{\partial \mu}}{\frac{\partial f}{\partial z}} \ , \tag{29.66}$$

by the chain rule.

29.4.4 Cold Start: Homotopy

In many cases, given a previous solution $Z(\mu_{(i-1)})$, we can use either of equations (29.63) or (29.64) to arrive at an educated guess $Z(\mu_{(i)})$ for the updated parameter $\mu_{(i)}$. If we have no previous solution, however, (e.g., $i = 1$) or our continuation techniques fail, we need some other means of generating an initial guess $Z(\mu_{(i)})$ that will be sufficiently good to converge to a correct solution.

A common approach to the "cold start" problem is to transform the original nonlinear problem $f(Z(\mu_{(i)});\mu_{(i)}) = 0$ into a form $\tilde{f}(Z(\mu_{(i)},t);\mu_{(i)},t) = 0$, i.e., we replace $f(Z;\mu_{(i)}) = 0$ with $\tilde{f}(\widetilde{Z};\mu_{(i)},t) = 0$. Here t is an additional, *artificial*, continuation parameter such that, *when* $t = 0$, the solution of the nonlinear problem

$$\tilde{f}(\widetilde{Z}(\mu_{(i)},t=0);\mu_{(i)},t=0) = 0 \tag{29.67}$$

is relatively simple (e.g., linear) or, perhaps, coincides with a preceding, known solution, and, *when* $t = 1$,

$$\tilde{f}(z;\mu_{(i)},t=1) = f(z;\mu_{(i)}) \tag{29.68}$$

such that $\tilde{f}(\widetilde{Z}(\mu_{(i)},t=1);\mu_{(i)},t=1) = 0$ implies $f(\widetilde{Z}(\mu_{(i)});\mu_{(i)}) = 0$ and hence $Z(\mu_{(i)})$ (the desired solution) $= \widetilde{Z}(\mu_{(i)},t=1)$.

We thus transform the "cold start" problem to a continuation problem in the artificial parameter t as t is varied from 0 to 1 with a start of its own (when $t = 0$) made significantly less "cold" by its — by construction — relative simplicity, and an end (when $t = 1$) that brings us smoothly to the solution of the original "cold start" problem.

As an example, we could replace the crank function f of (29.61) with a function $\tilde{f}(X;\theta,t;R,H) =$ *at* $X^2 + bX + c$ such that for $t = 0$ the problem is linear and the solution readily obtained.

29.4.5 A General Path Approach: Many Parameters

We consider now an n-vector of functions

$$\boldsymbol{f}(\boldsymbol{z};\boldsymbol{\mu}) = (f_1(\boldsymbol{z};\boldsymbol{\mu}) \quad f_2(\boldsymbol{z};\boldsymbol{\mu}) \quad \ldots \quad f_n(\boldsymbol{z};\boldsymbol{\mu}))^{\mathrm{T}} \tag{29.69}$$

that depends on an n-vector of unknowns $\boldsymbol{z}$

$$\boldsymbol{z} = (z_1 \quad z_2 \quad \ldots \quad z_n)^{\mathrm{T}} \tag{29.70}$$

and a parameter ℓ-vector $\boldsymbol{\mu}$ (independent of $\boldsymbol{z}$)

$$\boldsymbol{\mu} = (\mu_1 \quad \mu_2 \quad \ldots \quad \mu_\ell)^{\mathrm{T}} \ . \tag{29.71}$$

We also introduce an inequality constraint function

$$C(\boldsymbol{Z}) = \begin{cases} 1 & \text{if constraint satisfied} \\ 0 & \text{if constraint not satisfied} \end{cases} . \tag{29.72}$$

Note this is not a constraint on $\boldsymbol{z}$ but rather a constraint on the (desired) root $\boldsymbol{Z}$. Then, given $\boldsymbol{\mu}$, we look for $\boldsymbol{Z} = (Z_1 \;\; Z_2 \;\; \ldots \;\; Z_n)^{\mathrm{T}}$ such that

$$\begin{cases} \boldsymbol{f}(\boldsymbol{Z}; \boldsymbol{\mu}) = \boldsymbol{0} \\ C(\boldsymbol{Z}) = 1 \end{cases} . \tag{29.73}$$

In words, $\boldsymbol{Z}$ is a solution of n nonlinear equations in n unknowns subject which satisfies the constraint C.

Now we consider a path or "trajectory" — a sequence of p parameter vectors $\boldsymbol{\mu}_{(1)}, \ldots, \boldsymbol{\mu}_{(p)}$. We wish to determine $\boldsymbol{Z}_{(i)}$, $1 \leq i \leq p$, such that

$$\begin{cases} \boldsymbol{f}(\boldsymbol{Z}_{(i)}; \boldsymbol{\mu}_{(i)}) = \boldsymbol{0} \\ C(\boldsymbol{Z}_{(i)}) = 1 \end{cases} . \tag{29.74}$$

We assume that $\boldsymbol{Z}_{(1)}$ is known and that $\boldsymbol{Z}_{(2)}, \ldots, \boldsymbol{Z}_{(p)}$ remain to be determined. We can expect, then, that as long as consecutive parameter vectors $\boldsymbol{\mu}_{(i-1)}$ and $\boldsymbol{\mu}_{(i)}$ are sufficiently close, we should be able to use our continuation techniques equations (29.63) or (29.64) to converge to a correct (i.e., satisfying $C(\boldsymbol{Z}_{(i)}) = 1$) solution $\boldsymbol{Z}_{(i)}$. If, however, $\boldsymbol{\mu}_{(i-1)}$ and $\boldsymbol{\mu}_{(i)}$ are not sufficiently close, our continuation techniques will not be sufficient (i.e. we will fail to converge to a solution at all or we will fail to converge to a solution satisfying C) and we will need to apply a — we hope — more fail-safe homotopy.

We can thus combine our continuation and homotopy frameworks into a single algorithm. One such approach is summarized in Algorithm 4. The key points of this algorithm are that (i) we are using the simple continuation approach given by equation (29.63) (i.e., using the previous solution as the initial guess for the current problem), and (ii) we are using a bisection-type homotopy that, each time Newton fails to converge to a correct solution, inserts a new point in the trajectory halfway between the previous correct solution and the failed point. The latter, in the language of homotopy, can be expressed as

$$\tilde{\boldsymbol{f}}(\boldsymbol{z}; \boldsymbol{\mu}_{(i)}, t) = \boldsymbol{f}(\boldsymbol{z}; (1-t)\boldsymbol{\mu}_{(i-1)} + t\boldsymbol{\mu}_{(i)}) \tag{29.75}$$

with $t = 0.5$ for the inserted point.

Although there are numerous other approaches to non-convergence in addition to Algorithm 4, such as relaxed Newton — in which Newton provides the direction for the update, but then we take just some small fraction of the proposed step — Algorithm 4 is included mainly for its generality, simplicity, and apparent robustness.

Algorithm 4 General Path Following Algorithm

for $i = 2 \colon p$ **do**
  $\boldsymbol{Z}_{(i)} \leftarrow \boldsymbol{Z}_{(i-1)}$
  repeat
    $\boldsymbol{Z}_{(i)} \leftarrow \{$Solve $\boldsymbol{f}(\boldsymbol{z}; \boldsymbol{\mu}_{(i)}) = \boldsymbol{0}$ via Newton given initial guess $\boldsymbol{Z}_{(i)}\}$
    if Newton does not converge OR $C(\boldsymbol{Z}_{(i)}) \neq 1$ **then**
      for $j = p \colon -1 \colon i$ **do**
        $\boldsymbol{\mu}_{(j+1)} \leftarrow \boldsymbol{\mu}_{(j)}$ {Shift path parameters by one index to accommodate insertion}
      end for
      $\boldsymbol{\mu}_{(i)} \leftarrow \frac{1}{2}(\boldsymbol{\mu}_{(i-1)} + \boldsymbol{\mu}_{(i)})$ {Insert point in path halfway between $\boldsymbol{\mu}_{(i-1)}$ and $\boldsymbol{\mu}_{(i)}$}
      $p \leftarrow p + 1$ {Increment p to account for insertion}
    end if
  until Newton converges AND $C(\boldsymbol{Z}_{(i)}) = 1$
end for

2.086 Numerical Computation for Mechanical Engineers

Fall 2012