DEFINITION D.15 Order less than n^δ
A sequence c_n, is of order less than n^δ, denoted $o(n^\delta)$, if and only if $\text{plim}(1/n^\delta)c_n$ *equals zero.*

Thus, in our examples, γ_n^2 is $O(n^{-1})$, $\text{Var}[x_{(1),n}]$ is $O(n^{-2})$ and $o(n^{-1})$, S_n is $O(n^2)(\delta$ equals $+2$ in this case), $\ln L(\theta)$ is $O(n)(\delta$ equals $+1)$, and c_n is $O(1)(\delta = 0)$. Important particular cases that we will encounter repeatedly in our work are sequences for which $\delta = 1$ or -1.

The notion of order of a sequence is often of interest in econometrics in the context of the variance of an estimator. Thus, we see in Section D.3 that an important element of our strategy for forming an asymptotic distribution is that the variance of the limiting distribution of $\sqrt{n}(\bar{x}_n - \mu)/\sigma$ is $O(1)$. In Example D.10 the variance of m_2 is the sum of three terms that are $O(n^{-1})$, $O(n^{-2})$, and $O(n^{-3})$. The sum is $O(n^{-1})$, because $n\,\text{Var}[m_2]$ converges to $\mu_4 - \sigma^4$, the numerator of the first, or *leading term*, whereas the second and third terms converge to zero. This term is also the *dominant term* of the sequence. Finally, consider the two divergent examples in the preceding list. S_n is simply a deterministic function of n that explodes. However, $\ln L(\theta) = n \ln \theta - \theta \Sigma_i x_i$ is the sum of a constant that is $O(n)$ and a random variable with variance equal to n/θ. The random variable "diverges" in the sense that its variance grows without bound as n increases.

APPENDIX E

COMPUTATION AND OPTIMIZATION

E.1 INTRODUCTION

FN 1

FN 2

The computation of empirical estimates by econometricians involves using digital computers and software written either by the researchers themselves or by others.[1] It is also a surprisingly balanced mix of art and science. It is important for software users to be aware of how results are obtained, not only to understand routine computations, but also to be able to explain the occasional strange and contradictory results that do arise. This appendix will describe some of the basic elements of computing and a number of tools that are used by econometricians.[2] Section E.2

[1] It is one of the interesting aspects of the development of econometric methodology that the adoption of certain classes of techniques has proceeded in discrete jumps with the development of software. Noteworthy examples include the appearance, both around 1970, of G. K. Joreskog's LISREL [Joreskog and Sorbom (1981)] program, which spawned a still-growing industry in linear structural modeling, and TSP [Hall (1982)], which was among the first computer programs to accept symbolic representations of econometric models and which provided a significant advance in econometric practice with its LSQ procedure for systems of equations. An extensive survey of the evolution of econometric software is given in Renfro (2007).

[2] This discussion is not intended to teach the reader how to write computer programs. For those who expect to do so, there are whole libraries of useful sources. Three very useful works are Kennedy and Gentle (1980), Abramovitz and Stegun (1971), and especially Press et al. (1986). The third of these provides a wealth of expertly written programs and a large amount of information about how to do computation efficiently and accurately. A recent survey of many areas of computation is Judd (1998).

then describes some techniques for computing certain integrals and derivatives that are recurrent in econometric applications. Section E.3 presents methods of optimization of functions. Some examples are given in Section E.4.

E.2 COMPUTATION IN ECONOMETRICS

This section will discuss some methods of computing integrals that appear frequently in econometrics.

E.2.1 COMPUTING INTEGRALS

One advantage of computers is their ability rapidly to compute approximations to complex functions such as logs and exponents. The basic functions, such as these, trigonometric functions, and so forth, are standard parts of the libraries of programs that accompany all scientific computing installations.[3] But one of the very common applications that often requires some high-level creativity by econometricians is the evaluation of integrals that do not have simple closed forms and that do not typically exist in "system libraries." We will consider several of these in this section. We will not go into detail on the nuts and bolts of how to compute integrals with a computer; rather, we will turn directly to the most common applications in econometrics.

FN 3

E.2.2 THE STANDARD NORMAL CUMULATIVE DISTRIBUTION FUNCTION

The standard normal cumulative distribution function (cdf) is ubiquitous in econometric models. Yet this most homely of applications must be computed by approximation. There are a number of ways to do so.[4] Recall that what we desire is

FN 4

$$\Phi(x) = \int_{-\infty}^{x} \phi(t)\,dt, \quad \text{where } \phi(t) = \frac{1}{\sqrt{2\pi}}e^{-t^2/2}.$$

One way to proceed is to use a Taylor series:

$$\Phi(x) \approx \sum_{i=0}^{M} \frac{1}{i!}\frac{d^i\Phi(x_0)}{dx_0^i}(x - x_0)^i.$$

The normal cdf has some advantages for this approach. First, the derivatives are simple and not integrals. Second, the function is **analytic**; as $M \longrightarrow \infty$, the approximation converges to the true value. Third, the derivatives have a simple form; they are the **Hermite polynomials** and they can be computed by a simple recursion. The 0th term in the preceding expansion is $\Phi(x)$ evaluated at the expansion point. The first derivative of the cdf is the pdf, so the terms from 2 onward are the derivatives of $\phi(x)$, once again evaluated at x_0. The derivatives of the standard normal pdf obey the recursion

$$\phi^i/\phi(x) = -x\phi^{i-1}/\phi(x) - (i-1)\phi^{i-2}/\phi(x),$$

where ϕ^i is $d^i\phi(x)/dx^i$. The zero and one terms in the sequence are one and $-x$. The next term is $x^2 - 1$, followed by $3x - x^3$ and $x^4 - 6x^2 + 3$, and so on. The approximation can be made

[3] Of course, at some level, these must have been programmed as approximations by someone.

[4] Many system libraries provide a related function, the *error function*, $\text{erf}(x) = (2/\sqrt{\pi})\int_0^x e^{-t^2}\,dt$. If this is available, then the normal cdf can be obtained from $\Phi(x) = \frac{1}{2} + \frac{1}{2}\text{erf}(x/\sqrt{2}), x \geq 0$ and $\Phi(x) = 1 - \Phi(-x), x \leq 0$.

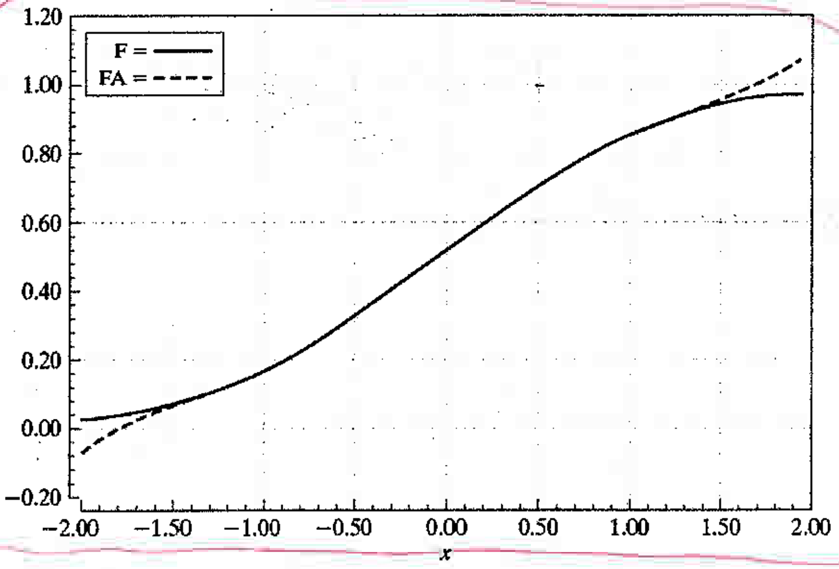


FIGURE E.1 Approximation to Normal cdf.

more accurate by adding terms. Consider using a fifth-order Taylor series approximation around the point $x = 0$, where $\Phi(0) = 0.5$ and $\phi(0) = 0.3989423$. Evaluating the derivatives at zero and assembling the terms produces the approximation

$$\Phi(x) \approx \tfrac{1}{2} + 0.3989423[x - x^3/6 + x^5/40].$$

FIG E.1

[Some of the terms (every other one, in fact) will conveniently drop out.] Figure E.1 shows the actual values (F) and approximate values (FA) over the range −2 to 2. The figure shows two important points. First, the approximation is remarkably good over most of the range. Second, as is usually true for Taylor series approximations, the quality of the approximation deteriorates as one gets far from the expansion point.

Unfortunately, it is the tail areas of the standard normal distribution that are usually of interest, so the preceding is likely to be problematic. An alternative approach that is used much more often is a polynomial approximation reported by Abramovitz and Stegun (1971, p. 932):

$$\Phi(-|x|) = \phi(x)\sum_{i=1}^{5} a_i t^i + \varepsilon(x), \quad \text{where } t = 1/[1 + a_0|x|].$$

(The complement is taken if x is positive.) The error of approximation is less than $\pm 7.5 \times 10^{-8}$ for all x. (Note that the error exceeds the function value at $|x| > 5.7$, so this is the operational limit of this approximation.)

E.2.3 THE GAMMA AND RELATED FUNCTIONS

The standard normal cdf is probably the most common application of numerical integration of a function in econometrics. Another very common application is the class of gamma functions. For positive constant P, the gamma function is

$$\Gamma(P) = \int_0^\infty t^{P-1}e^{-t}\,dt.$$

The gamma function obeys the recursion $\Gamma(P) = (P-1)\Gamma(P-1)$, so for integer values of P, $\Gamma(P) = (P-1)!$. This result suggests that the gamma function can be viewed as a generalization of the factorial function for noninteger values. Another convenient value is $\Gamma(\frac{1}{2}) = \sqrt{\pi}$. By making a change of variable, it can be shown that for positive constants a, c, and P,

$$\int_0^\infty t^{P-1}e^{-at^c}\,dt = \int_0^\infty t^{-(P+1)}e^{-a/t^c}\,dt = \left(\frac{1}{c}\right)a^{-P/c}\Gamma\left(\frac{P}{c}\right). \tag{E-1}$$

As a generalization of the factorial function, the gamma function will usually overflow for the sorts of values of P that normally appear in applications. The log of the function should normally be used instead. The function $\ln\Gamma(P)$ can be approximated remarkably accurately with only a handful of terms and is very easy to program. A number of approximations appear in the literature; they are generally modifications of **Sterling's approximation** to the factorial function $P! \approx (2\pi P)^{1/2}P^Pe^{-P}$, so i

$$\ln\Gamma(P) \approx (P-0.5)\ln P - P + 0.5\ln(2\pi) + C + \varepsilon(P),$$

where C is the correction term [see, e.g., Abramovitz and Stegun (1971, p. 257), Press et al. (1986, p. 157), or Rao (1973, p. 59)] and $\varepsilon(P)$ is the approximation error.[5] FN 5

The derivatives of the gamma function are

$$\frac{d^r\Gamma(P)}{dP^r} = \int_0^\infty (\ln \overset{t}{\cancel{P}})^r t^{P-1}e^{-t}\,dt.$$

The first two derivatives of $\ln\Gamma(P)$ are denoted $\Psi(P) = \Gamma'/\Gamma$ and $\Psi'(P) = (\Gamma\Gamma'' - \Gamma'^2)/\Gamma^2$ and are known as the **digamma** and **trigamma** functions.[6] The **beta function**, denoted $\beta(a, b)$, FN 6

$$\beta(a, b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,dt = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)},$$

is related.

E.2.4 APPROXIMATING INTEGRALS BY QUADRATURE

The digamma and trigamma functions, and the gamma function for noninteger values of P and values that are not integers plus $\frac{1}{2}$, do not exist in closed form and must be approximated. Most other applications will also involve integrals for which no simple computing function exists. The simplest approach to approximating

$$F(x) = \int_{L(x)}^{U(x)} f(t)\,dt$$

[5] For example, one widely used formula is $C = z^{-1}/12 - z^{-3}/360 - z^{-5}/1260 + z^{-7}/1680 - q$, where $z = P$ and $q = 0$ if $P > 18$, or $z = P + J$ and $q = \ln[P(P+1)(P+2)\cdots(P+J-1)]$, where $J = 18 - \text{INT}(P)$, if not. Note, in the approximation, we write $\Gamma(P) = (P!)/P +$ a correction.

[6] Tables of specific values for the gamma, digamma, and trigamma functions appear in Abramovitz and Stegun (1971). Most contemporary econometric programs have built-in functions for these common integrals, so the tables are not generally needed.

is likely to be a variant of Simpson's rule, or the trapezoid rule. For example, one approximation [see Press et al. (1986, p. 108)] is

$$F(x) \approx \Delta\left[\tfrac{1}{3}f_1 + \tfrac{4}{3}f_2 + \tfrac{2}{3}f_3 + \tfrac{4}{3}f_4 + \cdots + \tfrac{2}{3}f_{N-2} + \tfrac{4}{3}f_{N-1} + \tfrac{1}{3}f_N\right],$$

where f_j is the function evaluated at N equally spaced points in $[L, U]$ including the endpoints and $\Delta = (L - U)/(N - 1)$. There are a number of problems with this method, most notably that it is difficult to obtain satisfactory accuracy with a moderate number of points.

Gaussian quadrature is a popular method of computing integrals. The general approach is to use an approximation of the form

$$\int_L^U W(x)f(x)\,dx \approx \sum_{j=1}^{M} w_j f(a_j),$$

where $W(x)$ is viewed as a "weighting" function for integrating $f(x)$, w_j is the **quadrature weight,** and a_j is the **quadrature abscissa.** Different weights and abscissas have been derived for several weighting functions. Two weighting functions common in econometrics are

$$W(x) = x^c e^{-x}, \quad x \in [0, \infty),$$

for which the computation is called **Gauss–Laguerre quadrature,** and

$$W(x) = e^{-x^2}, \quad x \in (-\infty, \infty),$$

for which the computation is called **Gauss–Hermite quadrature.** The theory for deriving weights and abscissas is given in Press et al. (1986, pp. 121–125). Tables of weights and abscissas for many values of M are given by Abramovitz and Stegun (1971). Applications of the technique appear in ~~Section 16.9.6.b and Chapter 23.~~ Chapters 14 and 17.

E.3 OPTIMIZATION

Nonlinear optimization (e.g., maximizing log-likelihood functions) is an intriguing practical problem. Theory provides few hard and fast rules, and there are relatively few cases in which it is obvious how to proceed. This section introduces some of the terminology and underlying theory of nonlinear optimization.[7] We begin with a general discussion on how to search for a solution to a nonlinear optimization problem and describe some specific commonly used methods. We then consider some practical problems that arise in optimization. An example is given in the final section.

FN 7

Consider maximizing the quadratic function

$$F(\theta) = a + \mathbf{b}'\theta - \tfrac{1}{2}\theta'\mathbf{C}\theta,$$

where $\mathbf{C}$ is a positive definite matrix. The first-order condition for a maximum is

$$\frac{\partial F(\theta)}{\partial \theta} = \mathbf{b} - \mathbf{C}\theta = \mathbf{0}. \qquad \text{(E-2)}$$

This set of *linear* equations has the unique solution

$$\theta = \mathbf{C}^{-1}\mathbf{b}. \qquad \text{(E-3)}$$

[7] There are numerous excellent references that offer a more complete exposition. Among these are Quandt (1983), Bazaraa and Shetty (1979), Fletcher (1980), and Judd (1998).

FN 8

This is a linear optimization problem. Note that it has a **closed-form solution;** for any a, $\mathbf{b}$, and $\mathbf{C}$, the solution can be computed directly.[8] In the more typical situation,

$$\frac{\partial F(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0} \tag{E-4}$$

FN 9

is a set of nonlinear equations that cannot be solved explicitly for $\boldsymbol{\theta}$.[9] The techniques considered in this section provide systematic means of searching for a solution.

We now consider the general problem of maximizing a function of several variables:

$$\text{maximize}_{\boldsymbol{\theta}}\, F(\boldsymbol{\theta}), \tag{E-5}$$

where $F(\boldsymbol{\theta})$ may be a log-likelihood or some other function. Minimization of $F(\boldsymbol{\theta})$ is handled by maximizing $-F(\boldsymbol{\theta})$. Two special cases are

$$F(\boldsymbol{\theta}) = \sum_{i=1}^{n} f_i(\boldsymbol{\theta}), \tag{E-6}$$

FN 10

which is typical for maximum likelihood problems, and the **least squares problem,**[10]

$$f_i(\boldsymbol{\theta}) = -(y_i - f(\mathbf{x}_i, \boldsymbol{\theta}))^2. \tag{E-7}$$

AU: Confirm x-ref to Chap 11

We treated the nonlinear least squares problem in detail in Chapter 11. An obvious way to search for the $\boldsymbol{\theta}$ that maximizes $F(\boldsymbol{\theta})$ is by trial and error. If $\boldsymbol{\theta}$ has only a single element and it is known approximately where the optimum will be found, then a **grid search** will be a feasible strategy. An example is a common time-series problem in which a one-dimensional search for a correlation coefficient is made in the interval $(-1, 1)$. The grid search can proceed in the obvious fashion—that is, $\ldots, -0.1, 0, 0.1, 0.2, \ldots$, then $\hat{\theta}_{\max} - 0.1$ to $\hat{\theta}_{\max} + 0.1$ in increments of 0.01, and so on—until the desired precision is achieved.[11] If $\boldsymbol{\theta}$ contains more than one parameter, then a grid search is likely to be extremely costly, particularly if little is known about the parameter vector at the outset. Nonetheless, relatively efficient methods have been devised. Quandt (1983) and Fletcher (1980) contain further details.

FN 11

There are also systematic, derivative-free methods of searching for a function optimum that resemble in some respects the algorithms that we will examine in the next section. The **downhill simplex** (and other simplex) methods[12] have been found to be very fast and effective for some problems. A recent entry in the econometrics literature is the method of **simulated annealing.**[13] These derivative-free methods, particularly the latter, are often very effective in problems with many variables in the objective function, but they usually require far more function evaluations than the methods based on derivatives that are considered below. Because the problems typically analyzed in econometrics involve relatively few parameters but often quite complex functions involving large numbers of terms in a summation, on balance, the gradient methods are usually going to be preferable.[14]

FN 12

FN 13

FN 14

[8] Notice that the constant a is irrelevant to the solution. Many maximum likelihood problems are presented with the preface "neglecting an irrelevant constant." For example, the log-likelihood for the normal linear regression model contains a term—$(n/2)\ln(2\pi)$—that can be discarded.

[9] See, for example, the normal equations for the nonlinear least squares estimators of Chapter ~~11~~ 7.

[10] Least squares is, of course, a minimization problem. The negative of the criterion is used to maintain consistency with the general formulation.

[11] There are more efficient methods of carrying out a one-dimensional search, for example, the golden section method. See Press et al. (1986, Chap. 10).

[12] See Nelder and Mead (1965) and Press et al. (1986).

[13] See Goffe, Ferrier, and Rodgers (1994) and Press et al. (1986, pp. 326–334).

[14] Goffe, Ferrier, and Rodgers (1994) did find that the method of simulated annealing was quite adept at finding the best among multiple solutions. This problem is common for derivative-based methods, because they usually have no method of distinguishing between a local optimum and a global one.

E.3.1 ALGORITHMS

A more effective means of solving most nonlinear maximization problems is by an **iterative algorithm:**

> Beginning from initial value $\boldsymbol{\theta}_0$, at entry to iteration t, if $\boldsymbol{\theta}_t$ is not the optimal value for $\boldsymbol{\theta}$, compute direction vector $\boldsymbol{\Delta}_t$, step size λ_t, then
>
> $$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda_t \boldsymbol{\Delta}_t. \tag{E-8}$$

FIG E.2

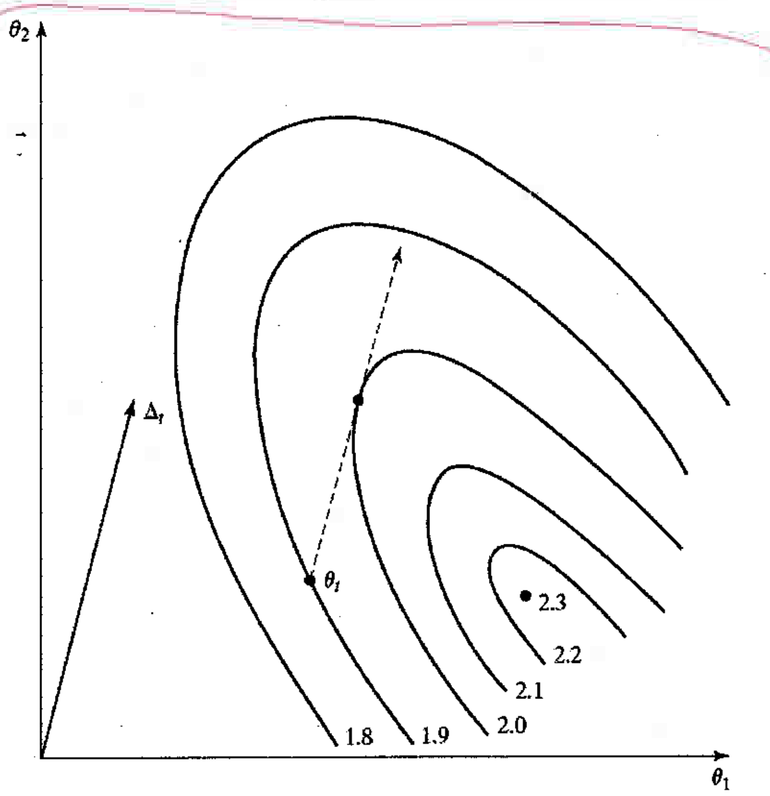
Figure E.2 illustrates the structure of an iteration for a hypothetical function of two variables. The direction vector $\boldsymbol{\Delta}_t$ is shown in the figure with $\boldsymbol{\theta}_t$. The dashed line is the set of points $\boldsymbol{\theta}_t + \lambda_t \boldsymbol{\Delta}_t$. Different values of λ_t lead to different contours; for this $\boldsymbol{\theta}_t$ and $\boldsymbol{\Delta}_t$, the best value of λ_t is about 0.5.

FIG E3

Notice in Figure E.2 that for a given direction vector $\boldsymbol{\Delta}_t$ and current parameter vector $\boldsymbol{\theta}_t$, a secondary optimization is required to find the best λ_t. Translating from Figure E.2, we obtain the form of this problem as shown in Figure E.3. This subsidiary search is called a **line search,** as we search along the line $\boldsymbol{\theta}_t + \lambda_t \boldsymbol{\Delta}_t$ for the optimal value of $F(.)$. The formal solution to the line search problem would be the λ_t that satisfies

$$\frac{\partial F(\boldsymbol{\theta}_t + \lambda_t \boldsymbol{\Delta}_t)}{\partial \lambda_t} = \mathbf{g}(\boldsymbol{\theta}_t + \lambda_t \boldsymbol{\Delta}_t)' \boldsymbol{\Delta}_t = 0, \tag{E-9}$$

FIGURE E.2 Iteration.



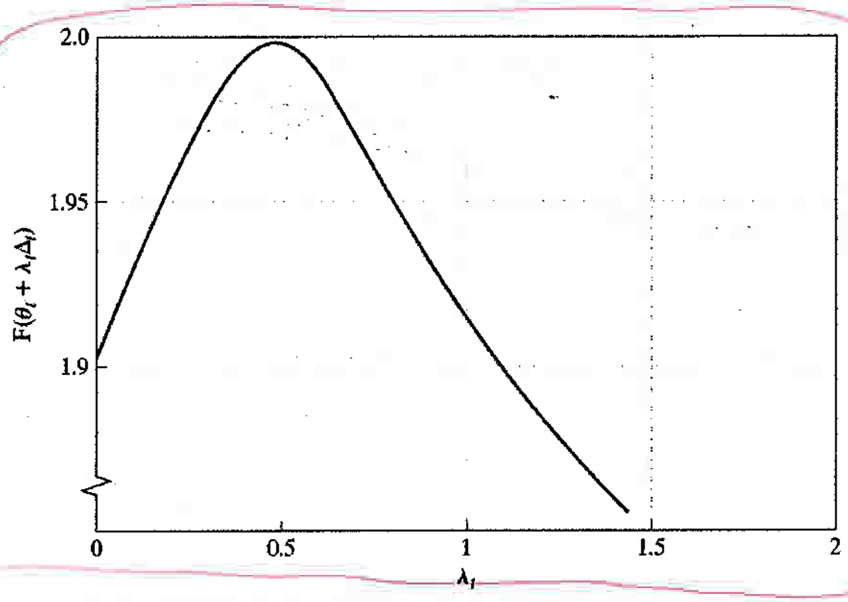


FIGURE E.3 Line Search.

where $\mathbf{g}$ is the vector of partial derivatives of $F(.)$ evaluated at $\theta_t + \lambda_t\Delta_t$. In general, this problem will also be a nonlinear one. In most cases, adding a formal search for λ_t will be too expensive, as well as unnecessary. Some approximate or ad hoc method will usually be chosen. It is worth emphasizing that finding the λ_t that maximizes $F(\theta_t + \lambda_t\Delta_t)$ at a given iteration does not generally lead to the overall solution in that iteration. This situation is clear in Figure E.3, where the optimal value of λ_t leads to $F(.) = 2.0$, at which point we reenter the iteration.

E.3.2 COMPUTING DERIVATIVES

For certain functions, the programming of derivatives may be quite difficult. Numeric approximations can be used, although it should be borne in mind that analytic derivatives obtained by formally differentiating the functions involved are to be preferred. First derivatives can be approximated by using

$$\frac{\partial F(\theta)}{\partial \theta_i} \approx \frac{F(\cdots\theta_i + \varepsilon\cdots) - F(\cdots\theta_i - \varepsilon\cdots)}{2\varepsilon}.$$

The choice of ε is a remaining problem. Extensive discussion may be found in Quandt (1983).

There are three drawbacks to this means of computing derivatives compared with using the analytic derivatives. A possible major consideration is that it may substantially increase the amount of computation needed to obtain a function and its gradient. In particular, $K + 1$ function evaluations (the criterion and K derivatives) are replaced with $2K + 1$ functions. The latter may be more burdensome than the former, depending on the complexity of the partial derivatives compared with the function itself. The comparison will depend on the application. But in most settings, careful programming that avoids superfluous or redundant calculation can make the advantage of the analytic derivatives substantial. Second, the choice of ε can be problematic. If it is chosen too large, then the approximation will be inaccurate. If it is chosen too small, then there may be insufficient variation in the function to produce a good estimate of the derivative.

A compromise that is likely to be effective is to compute ε_i separately for each parameter, as in

$$\varepsilon_i = \text{Max}[\alpha|\theta_i|, \gamma]$$

[see Goldfeld and Quandt (1971)]. The values α and γ should be relatively small, such as 10^{-5}. Third, although numeric derivatives computed in this fashion are likely to be reasonably accurate, in a sum of a large number of terms, say, several thousand, enough approximation error can accumulate to cause the numerical derivatives to differ significantly from their analytic counterparts. Second derivatives can also be computed numerically. In addition to the preceding problems, however, it is generally not possible to ensure negative definiteness of a Hessian computed in this manner. Unless the choice of ε is made extremely carefully, an indefinite matrix is a possibility. In general, the use of numeric derivatives should be avoided if the analytic derivatives are available.

E.3.3 GRADIENT METHODS

The most commonly used algorithms are **gradient methods,** in which

$$\boldsymbol{\Delta}_t = \mathbf{W}_t\mathbf{g}_t, \tag{E-10}$$

where $\mathbf{W}_t$ is a positive definite matrix and $\mathbf{g}_t$ is the **gradient** of $F(\boldsymbol{\theta}_t)$:

$$\mathbf{g}_t = \mathbf{g}(\boldsymbol{\theta}_t) = \frac{\partial F(\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t}. \tag{E-11}$$

These methods are motivated partly by the following. Consider a linear Taylor series approximation to $F(\boldsymbol{\theta}_t + \lambda_t\boldsymbol{\Delta}_t)$ around $\lambda_t = 0$:

$$F(\boldsymbol{\theta}_t + \lambda_t\boldsymbol{\Delta}_t) \simeq F(\boldsymbol{\theta}_t) + \lambda_t\mathbf{g}(\boldsymbol{\theta}_t)'\boldsymbol{\Delta}_t. \tag{E-12}$$

Let $F(\boldsymbol{\theta}_t + \lambda_t\boldsymbol{\Delta}_t)$ equal F_{t+1}. Then,

$$F_{t+1} - F_t \simeq \lambda_t\mathbf{g}_t'\boldsymbol{\Delta}_t.$$

If $\boldsymbol{\Delta}_t = \mathbf{W}_t\mathbf{g}_t$, then

$$F_{t+1} - F_t \simeq \lambda_t\mathbf{g}_t'\mathbf{W}_t\mathbf{g}_t.$$

If $\mathbf{g}_t$ is not $\mathbf{0}$ and λ_t is small enough, then $F_{t+1} - F_t$ must be positive. Thus, if $F(\boldsymbol{\theta})$ is not already at its maximum, then we can always find a step size such that a gradient-type iteration will lead to an increase in the function. (Recall that $\mathbf{W}_t$ is assumed to be positive definite.)

In the following, we will omit the iteration index t, except where it is necessary to distinguish one vector from another. The following are some commonly used algorithms.[15]

FN 15

Steepest Ascent The simplest algorithm to employ is the **steepest ascent** method, which uses

$$\mathbf{W} = \mathbf{I} \text{ so that } \boldsymbol{\Delta} = \mathbf{g}. \tag{E-13}$$

As its name implies, the direction is the one of greatest increase of $F(.)$. Another virtue is that the line search has a straightforward solution; at least near the maximum, the optimal λ is

$$\lambda = \frac{-\mathbf{g}'\mathbf{g}}{\mathbf{g}'\mathbf{H}\mathbf{g}}, \tag{E-14}$$

[15] A more extensive catalog may be found in Judge et al. (1985, Appendix B). Those mentioned here are some of the more commonly used ones and are chosen primarily because they illustrate many of the important aspects of nonlinear optimization.

where

$$\mathbf{H} = \frac{\partial^2 F(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}'}.$$

Therefore, the steepest ascent iteration is

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\mathbf{g}_t'\mathbf{g}_t}{\mathbf{g}_t'\mathbf{H}_t\mathbf{g}_t}\mathbf{g}_t. \tag{E-15}$$

Computation of the second derivatives matrix may be extremely burdensome. Also, if $\mathbf{H}_t$ is not negative definite, which is likely if $\boldsymbol{\theta}_t$ is far from the maximum, the iteration may diverge. A systematic line search can bypass this problem. This algorithm usually converges very slowly, however, so other techniques are usually used.

Newton's Method

The template for most gradient methods in common use is Newton's method. The basis for **Newton's method** is a linear Taylor series approximation. Expanding the first-order conditions,

$$\frac{\partial F(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0},$$

equation by equation, in a linear Taylor series around an arbitrary $\boldsymbol{\theta}^0$ yields

$$\frac{\partial F(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \simeq \mathbf{g}^0 + \mathbf{H}^0(\boldsymbol{\theta} - \boldsymbol{\theta}^0) = \mathbf{0}, \tag{E-16}$$

where the superscript indicates that the term is evaluated at $\boldsymbol{\theta}^0$. Solving for $\boldsymbol{\theta}$ and then equating $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_{t+1}$ and $\boldsymbol{\theta}^0$ to $\boldsymbol{\theta}_t$, we obtain the iteration

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{H}_t^{-1}\mathbf{g}_t. \tag{E-17}$$

Thus, for Newton's method,

$$\mathbf{W} = -\mathbf{H}^{-1}, \qquad \boldsymbol{\Delta} = -\mathbf{H}^{-1}\mathbf{g}, \qquad \lambda = 1. \tag{E-18}$$

Newton's method will converge very rapidly in many problems. If the function is quadratic, then this method will reach the optimum in one iteration from any starting point. If the criterion function is globally concave, as it is in a number of problems that we shall examine in this text, then it is probably the best algorithm available. This method is very well suited to maximum likelihood estimation.

Alternatives to Newton's Method

Newton's method is very effective in some settings, but it can perform very poorly in others. If the function is not approximately quadratic or if the current estimate is very far from the maximum, then it can cause wide swings in the estimates and even fail to converge at all. A number of algorithms have been devised to improve upon Newton's method. An obvious one is to include a line search at each iteration rather than use $\lambda = 1$. Two problems remain, however. At points distant from the optimum, the second derivatives matrix may not be negative definite, and, in any event, the computational burden of computing $\mathbf{H}$ may be excessive.

The **quadratic hill-climbing method** proposed by Goldfeld, Quandt, and Trotter (1966) deals directly with the first of these problems. In any iteration, if $\mathbf{H}$ is not negative definite, then it is replaced with

$$\mathbf{H}_\alpha = \mathbf{H} - \alpha\mathbf{I}, \tag{E-19}$$

where α is a positive number chosen large enough to ensure the negative definiteness of $\mathbf{H}_\alpha$. Another suggestion is that of Greenstadt (1967), which uses, at every iteration,

$$\mathbf{H}_\pi = -\sum_{i=1}^{n} |\pi_i|\,\mathbf{c}_i\mathbf{c}_i', \tag{E-20}$$

where π_i is the ith characteristic root of $\mathbf{H}$ and $\mathbf{c}_i$ is its associated characteristic vector. Other proposals have been made to ensure the negative definiteness of the required matrix at each iteration.[16]

FN 16

Quasi-Newton Methods: Davidon–Fletcher–Powell A very effective class of algorithms has been developed that eliminates second derivatives altogether and has excellent convergence properties, even for ill-behaved problems. These are the **quasi-Newton methods,** which form

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mathbf{E}_t,$$

FN 17

where $\mathbf{E}_t$ is a positive definite matrix.[17] As long as $\mathbf{W}_0$ is positive definite—$\mathbf{I}$ is commonly used—$\mathbf{W}_t$ will be positive definite at every iteration. In the **Davidon–Fletcher–Powell (DFP) method,** after a sufficient number of iterations, $\mathbf{W}_{t+1}$ will be an approximation to $-\mathbf{H}^{-1}$. Let

$$\boldsymbol{\delta}_t = \lambda_t\boldsymbol{\Delta}_t, \quad \text{and} \quad \boldsymbol{\gamma}_t = \mathbf{g}(\boldsymbol{\theta}_{t+1}) - \mathbf{g}(\boldsymbol{\theta}_t). \tag{E-21}$$

The DFP **variable metric algorithm** uses

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \frac{\boldsymbol{\delta}_t\boldsymbol{\delta}_t'}{\boldsymbol{\delta}_t'\boldsymbol{\gamma}_t} + \frac{\mathbf{W}_t\boldsymbol{\gamma}_t\boldsymbol{\gamma}_t'\mathbf{W}_t}{\boldsymbol{\gamma}_t'\mathbf{W}_t\boldsymbol{\gamma}_t}. \tag{E-22}$$

Notice that in the DFP algorithm, the change in the first derivative vector is used in $\mathbf{W}$; an estimate of the inverse of the second derivatives matrix is being accumulated.

The variable metric algorithms are those that update $\mathbf{W}$ at each iteration while preserving its definiteness. For the DFP method, the accumulation of $\mathbf{W}_{t+1}$ is of the form

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mathbf{a}\mathbf{a}' + \mathbf{b}\mathbf{b}' = \mathbf{W}_t + [\mathbf{a} \quad \mathbf{b}][\mathbf{a} \quad \mathbf{b}]'.$$

The two-column matrix $[\mathbf{a} \quad \mathbf{b}]$ will have rank two; hence, DFP is called a **rank two update** or **rank two correction.** The **Broyden–Fletcher–Goldfarb–Shanno (BFGS)** method is a rank three correction that subtracts $v\mathbf{dd}'$ from the DFP update, where $v = (\boldsymbol{\gamma}_t'\mathbf{W}_t\boldsymbol{\gamma}_t)$ and

$$\mathbf{d}_t = \left(\frac{1}{\boldsymbol{\delta}_t'\boldsymbol{\gamma}_t}\right)\boldsymbol{\delta}_t - \left(\frac{1}{\boldsymbol{\gamma}_t'\mathbf{W}_t\boldsymbol{\gamma}_t}\right)\mathbf{W}_t\boldsymbol{\gamma}_t.$$

There is some evidence that this method is more efficient than DFP. Other methods, such as **Broyden's method,** involve a rank one correction instead. Any method that is of the form

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mathbf{QQ}'$$

will preserve the definiteness of $\mathbf{W}$, regardless of the number of columns in $\mathbf{Q}$.

The DFP and BFGS algorithms are extremely effective and are among the most widely used of the gradient methods. An important practical consideration to keep in mind is that although $\mathbf{W}_t$ accumulates an estimate of the negative inverse of the second derivatives matrix for both algorithms, in maximum likelihood problems it rarely converges to a very good estimate of the covariance matrix of the estimator and should generally not be used as one.

[16] See, for example, Goldfeld and Quandt (1971).

[17] See Fletcher (1980).

E.3.4 ASPECTS OF MAXIMUM LIKELIHOOD ESTIMATION

Newton's method is often used for maximum likelihood problems. For solving a maximum likelihood problem, the **method of scoring** replaces $\mathbf{H}$ with

$$\overline{\mathbf{H}} = E[\mathbf{H}(\theta)], \tag{E-23}$$

which will be recognized as the asymptotic covariance of the maximum likelihood estimator. There is some evidence that where it can be used, this method performs better than Newton's method. The exact form of the expectation of the Hessian of the log likelihood is rarely known, however.[18] Newton's method, which uses actual instead of expected second derivatives, is generally used instead.

FN 18

One-Step Estimation A convenient variant of Newton's method is the **one-step maximum likelihood estimator.** It has been shown that if θ^0 is *any* consistent initial estimator of θ and $\mathbf{H}^*$ is $\mathbf{H}$, $\overline{\mathbf{H}}$, or any other asymptotically equivalent estimator of $\text{Var}[\mathbf{g}(\hat{\theta}_{\text{MLE}})]$, then

$$\theta^1 = \theta^0 - (\mathbf{H}^*)^{-1}\mathbf{g}^0 \tag{E-24}$$

is an estimator of θ that has the same asymptotic properties as the maximum likelihood estimator.[19] (Note that it is *not* the maximum likelihood estimator. As such, for example, it should not be used as the basis for likelihood ratio tests.)

FN 19

Covariance Matrix Estimation In computing maximum likelihood estimators, a commonly used method of estimating $\mathbf{H}$ simultaneously simplifies the calculation of $\mathbf{W}$ and solves the occasional problem of indefiniteness of the Hessian. The method of Berndt et al. (1974) replaces $\mathbf{W}$ with

$$\hat{\mathbf{W}} = \left[\sum_{i=1}^{n} \mathbf{g}_i\mathbf{g}_i'\right]^{-1} = (\mathbf{G}'\mathbf{G})^{-1}, \tag{E-25}$$

where

$$\mathbf{g}_i = \frac{\partial \ln f(y_i \,|\, \mathbf{x}_i, \theta)}{\partial \theta}. \tag{E-26}$$

Then, $\mathbf{G}$ is the $n \times K$ matrix with ith row equal to $\mathbf{g}_i'$. Although $\hat{\mathbf{W}}$ and other suggested estimators of $(-\mathbf{H})^{-1}$ are asymptotically equivalent, $\hat{\mathbf{W}}$ has the additional virtues that it is always nonnegative definite, and it is only necessary to differentiate the log-likelihood once to compute it.

The Lagrange Multiplier Statistic The use of $\hat{\mathbf{W}}$ as an estimator of $(-\mathbf{H})^{-1}$ brings another intriguing convenience in maximum likelihood estimation. When testing restrictions on parameters estimated by maximum likelihood, one approach is to use the **Lagrange multiplier** statistic. We will examine this test at length at various points in this book, so we need only sketch it briefly here. The logic of the LM test is as follows. The gradient $\mathbf{g}(\theta)$ of the log-likelihood function equals $\mathbf{0}$ at the unrestricted maximum likelihood estimators (that is, at least to within the precision of the computer program in use). If $\hat{\theta}_r$ is an MLE that is computed subject to some restrictions on θ, then we know that $\mathbf{g}(\hat{\theta}_r) \neq \mathbf{0}$. The LM test is used to test whether, at $\hat{\theta}_r$, $\mathbf{g}_r$ is *significantly* different from $\mathbf{0}$ or whether the deviation of $\mathbf{g}_r$ from $\mathbf{0}$ can be viewed as sampling variation. The covariance matrix of the gradient of the log-likelihood is $-\mathbf{H}$, so the Wald statistic for testing this hypothesis is $W = \mathbf{g}'(-\mathbf{H})^{-1}\mathbf{g}$. Now, suppose that we use $\hat{\mathbf{W}}$ to estimate $-\mathbf{H}^{-1}$. Let $\mathbf{G}$ be the $n \times K$ matrix with ith row equal to $\mathbf{g}_i'$, and let $\mathbf{i}$ denote an $n \times 1$ column of ones. Then the LM statistic can be

[18] Amemiya (1981) provides a number of examples.

[19] See, for example, Rao (1973).

computed as

$$\text{LM} = \mathbf{i}'\mathbf{G}(\mathbf{G}'\mathbf{G})^{-1}\mathbf{G}'\mathbf{i}.$$

Because $\mathbf{i}'\mathbf{i} = n$,

$$\text{LM} = n[\mathbf{i}'\mathbf{G}(\mathbf{G}'\mathbf{G})^{-1}\mathbf{G}'\mathbf{i}/n] = nR_i^2,$$

where R_i^2 is the *uncentered* R^2 in a regression of a column of ones on the derivatives of the log-likelihood function.

The Concentrated Log-Likelihood Many problems in maximum likelihood estimation can be formulated in terms of a partitioning of the parameter vector $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2]$ such that at the solution to the optimization problem, $\boldsymbol{\theta}_{2,\text{ML}}$, can be written as an explicit function of $\boldsymbol{\theta}_{1,\text{ML}}$. When the solution to the likelihood equation for $\boldsymbol{\theta}_2$ produces

$$\boldsymbol{\theta}_{2,\text{ML}} = \mathbf{t}(\boldsymbol{\theta}_{1,\text{ML}}),$$

then, if it is convenient, we may "concentrate" the log-likelihood function by writing

$$F^*(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = F[\boldsymbol{\theta}_1, \mathbf{t}(\boldsymbol{\theta}_1)] = F_c(\boldsymbol{\theta}_1).$$

The unrestricted solution to the problem $\text{Max}_{\boldsymbol{\theta}_1} F_c(\boldsymbol{\theta}_1)$ provides the full solution to the optimization problem. Once the optimizing value of $\boldsymbol{\theta}_1$ is obtained, the optimizing value of $\boldsymbol{\theta}_2$ is simply $\mathbf{t}(\hat{\boldsymbol{\theta}}_{1,\text{ML}})$. Note that $F^*(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ is a subset of the set of values of the log-likelihood function, namely those values at which the second parameter vector satisfies the first-order conditions.[20]

EN 20

E.3.5 OPTIMIZATION WITH CONSTRAINTS

Occasionally, some of or all the parameters of a model are constrained, for example, to be positive in the case of a variance or to be in a certain range, such as a correlation coefficient. Optimization subject to constraints is often yet another art form. The elaborate literature on the general problem provides some guidance—see, for example, Appendix B in Judge et al. (1985)—but applications still, as often as not, require some creativity on the part of the analyst. In this section, we will examine a few of the most common forms of constrained optimization as they arise in econometrics.

Parametric constraints typically come in two forms, which may occur simultaneously in a problem. Equality constraints can be written $\mathbf{c}(\boldsymbol{\theta}) = \mathbf{0}$, where $c_j(\boldsymbol{\theta})$ is a continuous and differentiable function. Typical applications include linear constraints on slope vectors, such as a requirement that a set of elasticities in a log-linear model add to one; exclusion restrictions, which are often cast in the form of interesting hypotheses about whether or not a variable should appear in a model (i.e., whether a coefficient is zero or not); and equality restrictions, such as the symmetry restrictions in a translog model, which require that parameters in two different equations be equal to each other. Inequality constraints, in general, will be of the form $a_j \leq c_j(\boldsymbol{\theta}) \leq b_j$, where a_j and b_j are known constants (either of which may be infinite). Once again, the typical application in econometrics involves a restriction on a single parameter, such as $\sigma > 0$ for a variance parameter, $-1 \leq \rho \leq 1$ for a correlation coefficient, or $\beta_j \geq 0$ for a particular slope coefficient in a model. We will consider the two cases separately.

In the case of equality constraints, for practical purposes of optimization, there are usually two strategies available. One can use a Lagrangean multiplier approach. The new optimization problem is

$$\text{Max}_{\boldsymbol{\theta},\boldsymbol{\lambda}}\, L(\boldsymbol{\theta}, \boldsymbol{\lambda}) = F(\boldsymbol{\theta}) + \boldsymbol{\lambda}'\mathbf{c}(\boldsymbol{\theta}).$$

[20] A formal proof that this is a valid way to proceed is given by Amemiya (1985, pp. 125–127).

The necessary conditions for an optimum are

$$\frac{\partial L(\boldsymbol{\theta},\boldsymbol{\lambda})}{\partial\boldsymbol{\theta}} = \mathbf{g}(\boldsymbol{\theta}) + \mathbf{C}(\boldsymbol{\theta})'\boldsymbol{\lambda} = \mathbf{0},$$

$$\frac{\partial L(\boldsymbol{\theta},\boldsymbol{\lambda})}{\partial\boldsymbol{\lambda}} = \mathbf{c}(\boldsymbol{\theta}) = \mathbf{0},$$

where $\mathbf{g}(\boldsymbol{\theta})$ is the familiar gradient of $F(\boldsymbol{\theta})$ and $\mathbf{C}(\boldsymbol{\theta})$ is a $J \times K$ matrix of derivatives with jth row equal to $\partial c_j/\partial\boldsymbol{\theta}'$. The joint solution will provide the constrained optimizer, as well as the Lagrange multipliers, which are often interesting in their own right. The disadvantage of this approach is that it increases the dimensionality of the optimization problem. An alternative strategy is to eliminate some of the parameters by either imposing the constraints directly on the function or by solving out the constraints. For exclusion restrictions, which are usually of the form $\theta_j = 0$, this step usually means dropping a variable from a model. Other restrictions can often be imposed just by building them into the model. For example, in a function of θ_1, θ_2, and θ_3, if the restriction is of the form $\theta_3 = \theta_1\theta_2$, then θ_3 can be eliminated from the model by a direct substitution.

Inequality constraints are more difficult. For the general case, one suggestion is to transform the constrained problem into an unconstrained one by imposing some sort of penalty function into the optimization criterion that will cause a parameter vector that violates the constraints, or nearly does so, to be an unattractive choice. For example, to force a parameter θ_j to be nonzero, one might maximize the augmented function $F(\boldsymbol{\theta}) - |1/\theta_j|$. This approach is feasible, but it has the disadvantage that because the penalty is a function of the parameters, different penalty functions will lead to different solutions of the optimization problem. For the most common problems in econometrics, a simpler approach will usually suffice. One can often reparameterize a function so that the new parameter is unconstrained. For example, the "method of squaring" is sometimes used to force a parameter to be positive. If we require θ_j to be positive, then we can define $\theta_j = \alpha^2$ and substitute α^2 for θ_j wherever it appears in the model. Then an unconstrained solution for α is obtained. An alternative reparameterization for a parameter that must be positive that is often used is $\theta_j = \exp(\alpha)$. To force a parameter to be between zero and one, we can use the function $\theta_j = 1/[1 + \exp(\alpha)]$. The range of α is now unrestricted. Experience suggests that a third, less orthodox approach works very well for many problems. When the constrained optimization is begun, there is a starting value $\boldsymbol{\theta}^0$ that begins the iterations. Presumably, $\boldsymbol{\theta}^0$ obeys the restrictions. (If not, and none can be found, then the optimization process must be terminated immediately.) The next iterate, $\boldsymbol{\theta}^1$, is a step away from $\boldsymbol{\theta}^0$, by $\boldsymbol{\theta}^1 = \boldsymbol{\theta}^0 + \lambda_0\boldsymbol{\delta}^0$. Suppose that $\boldsymbol{\theta}^1$ violates the constraints. By construction, we know that there is some value $\boldsymbol{\theta}^1_*$ between $\boldsymbol{\theta}^0$ and $\boldsymbol{\theta}^1$ that does not violate the constraint, where "between" means only that a shorter step is taken. Therefore, the next value for the iteration can be $\boldsymbol{\theta}^1_*$. The logic is true at every iteration, so a way to proceed is to alter the iteration so that the step length is shortened when necessary when a parameter violates the constraints.

E.3.6 SOME PRACTICAL CONSIDERATIONS

The reasons for the good performance of many algorithms, including DFP, are unknown. Moreover, different algorithms may perform differently in given settings. Indeed, for some problems, one algorithm may fail to converge whereas another will succeed in finding a solution without great difficulty. In view of this, computer programs such as GQOPT,[21] Gauss, and MatLab that offer a menu of different preprogrammed algorithms can be particularly useful. It is sometimes worth the effort to try more than one algorithm on a given problem.

FN 21

[21] Goldfeld and Quandt (1972).

Step Sizes Except for the steepest ascent case, an optimal line search is likely to be infeasible or to require more effort than it is worth in view of the potentially large number of function evaluations required. In most cases, the choice of a step size is likely to be rather ad hoc. But within limits, the most widely used algorithms appear to be robust to inaccurate line searches. For example, one method employed by the widely used TSP computer program[22] is the method of *squeezing*, which tries $\lambda = 1, \frac{1}{2}, \frac{1}{4}$, and so on until an improvement in the function results. Although this approach is obviously a bit unorthodox, it appears to be quite effective when used with the Gauss–Newton method for nonlinear least squares problems. (See Chapter 11.) A somewhat more elaborate rule is suggested by Berndt et al. (1974). Choose an ε between 0 and $\frac{1}{2}$, and then find a λ such that

FN 22

AU: Confirm x-ref to Chap. 11

$$\varepsilon < \frac{F(\theta + \lambda\Delta) - F(\theta)}{\lambda \mathbf{g}'\Delta} < 1 - \varepsilon. \tag{E-27}$$

Of course, which value of ε to choose is still open, so the choice of λ remains ad hoc. Moreover, in neither of these cases is there any optimality to the choice; we merely find a λ that leads to a function improvement. Other authors have devised relatively efficient means of searching for a step size without doing the full optimization at each iteration.[23]

FN 23

Assessing Convergence Ideally, the iterative procedure should terminate when the gradient is zero. In practice, this step will not be possible, primarily because of accumulated rounding error in the computation of the function and its derivatives. Therefore, a number of alternative convergence criteria are used. Most of them are based on the relative changes in the function or the parameters. There is considerable variation in those used in different computer programs, and there are some pitfalls that should be avoided. A critical absolute value for the elements of the gradient or its norm will be affected by any scaling of the function, such as normalizing it by the sample size. Similarly, stopping on the basis of small absolute changes in the parameters can lead to premature convergence when the parameter vector approaches the maximizer. It is probably best to use several criteria simultaneously, such as the proportional change in both the function and the parameters. Belsley (1980) discusses a number of possible stopping rules. One that has proved useful and is immune to the scaling problem is to base convergence on $\mathbf{g}'\mathbf{H}^{-1}\mathbf{g}$.

Multiple Solutions It is possible for a function to have several local extrema. It is difficult to know a priori whether this is true of the one at hand. But if the function is not globally concave, then it may be a good idea to attempt to maximize it from several starting points to ensure that the maximum obtained is the global one. Ideally, a starting value near the optimum can facilitate matters; in some settings, this can be obtained by using a consistent estimate of the parameter for the starting point. The method of moments, if available, is sometimes a convenient device for doing so.

No Solution Finally, it should be noted that in a nonlinear setting the iterative algorithm can break down, even in the absence of constraints, for at least two reasons. The first possibility is that the problem being solved may be so numerically complex as to defy solution. The second possibility, which is often neglected, is that the proposed model may simply be inappropriate for the data. In a linear setting, a low R^2 or some other diagnostic test may suggest that the model and data are mismatched, but as long as the full rank condition is met by the regressor matrix, a linear regression can *always* be computed. Nonlinear models are not so forgiving. The failure of an iterative algorithm to find a maximum of the criterion function may be a warning that the model is not appropriate for this body of data.

[22] Hall (1982, p. 147).

[23] See, for example, Joreskog and Gruvaeus (1970), Powell (1964), Quandt (1983), and Hall (1982).

E.3.7 THE EM ALGORITHM

The latent class model can be characterized as a **missing data model.** Consider the mixture model we used for DocVis in ~~Example 16.21~~, which we will now generalize to allow more than two classes: Chapter 14 ~~16~~

$$f(y_{it} \mid \mathbf{x}_{it}, class_i = j) = \theta_{it,j}(1-\theta_{it,j})^{y_{it}},\ \theta_{it,j} = 1/(1+\lambda_{it,j}),\ \lambda_{it,j} = \exp(\mathbf{x}_{it}'\boldsymbol{\beta}_j),\ y_{it} = 0, 1, \ldots.$$

$$\text{Prob}(class_i = j \mid \mathbf{z}_i) = \frac{\exp(\mathbf{z}_i'\boldsymbol{\alpha}_j)}{\sum_{j=1}^{J}\exp(\mathbf{z}_i'\boldsymbol{\alpha}_j)},\ j = 1, 2, \ldots, J.$$

With all parts incorporated, the log-likelihood for this latent class model is

$(1-y_{it})$

$$\begin{aligned}\ln L_M &= \sum_{i=1}^{n}\ln L_{i,M}\\ &= \sum_{i=1}^{n}\ln\left\{\sum_{j=1}^{J}\frac{\exp(\mathbf{z}_i'\boldsymbol{\alpha}_j)}{\sum_{m=1}^{J}\exp(\mathbf{z}_i'\boldsymbol{\alpha}_m)}\prod_{t=1}^{T_i}\left(\frac{1}{1+\exp(\mathbf{x}_{it}'\boldsymbol{\beta}_j)}\right)\left(\frac{\exp(\mathbf{x}_{it}'\boldsymbol{\beta}_j)}{1+\exp(\mathbf{x}_{it}'\boldsymbol{\beta}_j)}\right)^{y_{it}}\right\}.\end{aligned}\tag{E-28}$$

Suppose the actual class memberships were known (i.e., observed). Then, the class probabilities in $\ln L_M$ would be unnecessary. The appropriate **complete data log-likelihood** for this case would be

$(1-y_{it})$

$$\begin{aligned}\ln L_C &= \sum_{i=1}^{n}\ln L_{i,C}\\ &= \sum_{i=1}^{n}\ln\left\{\sum_{j=1}^{J}D_{ij}\prod_{t=1}^{T_i}\left(\frac{1}{1+\exp(\mathbf{x}_{it}'\boldsymbol{\beta}_j)}\right)\left(\frac{\exp(\mathbf{x}_{it}'\boldsymbol{\beta}_j)}{1+\exp(\mathbf{x}_{it}'\boldsymbol{\beta}_j)}\right)^{y_{it}}\right\},\end{aligned}\tag{E-29}$$

where D_{ij} is an observed dummy variable that equals one if individual i is from class j, and zero otherwise. With this specification, the log-likelihood breaks into J separate log-likelihoods, one for each (now known) class. The maximum likelihood estimates of $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J$ would be obtained simply by separating the sample into the respective subgroups and estimating the appropriate model for each group using maximum likelihood. The method we have used to estimate the parameters of the full model is to replace the D_{ij} variables with their unconditional espectations, Prob($class_i = j|\mathbf{z}_i$), then maximize the resulting log-likelihood function. This is the essential logic of the **EM** (expectation–maximization) **algorithm** [Dempster et al. (1977)]; however, the method uses the conditional (posterior) class probabilities instead of the unconditional probabilities. The iterative steps of the EM algorithm are

(E step) Form the expectation of the missing data log-likelihood, conditional on the previous parameter estimates and the data in the sample;

(M step) Maximize the expected log-likelihood function. Then either return to the E step or exit if the estimates have converged.

The EM algorithm can be used in a variety of settings. [See McLachlan and Krishnan (1997).] It has a particularly appealing form for estimating latent class models. The iterative steps for the latent class model are as follows:

(E step) Form the conditional (posterior) class probabilities, $\pi_{ij}|\mathbf{z}_i$, based on the current estimates. These are based on the likelihood function.

(M step) For each class, estimate the class-specific parameters by maximizing a weighted log-likelihood,

$$\ln L_{M\,step,j} = \sum_{i=1}^{n_c} \pi_{ij} \ln L_i \mid class = j.$$

The parameters of the class probability model are also reestimated, as shown later, when there are variables in z_i other than a constant term.

This amounts to a simple weighted estimation. For example, in the latent class linear regression model, the M step would amount to nothing more than weighted least squares. For nonlinear models such as the geometric model above, the M step involves maximizing a weighted log-likelihood function.

For the preceding geometric model, the precise steps are as follows: First, obtain starting values for $\beta_1, \ldots, \beta_J, \alpha_1, \ldots, \alpha_J$. Recall, $\alpha_J = 0$. Then:

1. Form the contributions to the likelihood function using (E-28),

$$L_i = \sum_{j=1}^{J} \pi_{ij} \prod_{t=1}^{T_i} f(y_{it} \mid \mathbf{x}_{it}, \beta_j, class_i = j)$$

$$= \sum_{j=1}^{J} L_i \mid class = j. \tag{E-30}$$

2. Form the conditional probabilities, $w_{ij} = \dfrac{L_i \mid class = j}{\sum_{m=1}^{J} L_i \mid class = m}$. **(E-31)**

3. For each j, now maximize the weighted log likelihood functions (one at a time),

$$\ln L_{j,M}(\beta_j) = \sum_{i=1}^{n} w_{ij} \ln \prod_{t=1}^{T_i} \left(\frac{1}{1+\exp(\mathbf{x}_{it}'\beta_j)}\right)^{(1-y_{it})} \left(\frac{\exp(\mathbf{x}_{it}'\beta_j)}{1+\exp(\mathbf{x}_{it}'\beta_j)}\right)^{y_{it}} \tag{E-32}$$

4. To update the α_j parameters, maximize the following log-likelihood function

$$\ln L(\alpha_1, \ldots, \alpha_J) = \sum_{i=1}^{n} \sum_{j=1}^{J} w_{ij} \ln \frac{\exp(\mathbf{z}_i'\alpha_j)}{\sum_{j=1}^{J} \exp(\mathbf{z}_i'\alpha_j)}, \quad \alpha_J = 0. \tag{E-33}$$

Step 4 defines a multinomial logit model (with "grouped") data. If the class probability model does not contain any variables in z_i, other than a constant, then the solutions to this optimization will be

$$\hat{\pi}_j = \frac{\sum_{i=1}^{n} w_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{J} w_{ij}}, \text{ then } \hat{\alpha}_j = \ln \frac{\hat{\pi}_j}{\hat{\pi}_J}. \tag{E-34}$$

(Note that this preserves the restriction $\hat{\alpha}_J = 0$.) With these in hand, we return to steps 1 and 2 to rebuild the weights, then perform steps 3 and 4. The process is iterated until the estimates of $\beta_1, \ldots, \beta_J$ converge. Step 1 is constructed in a generic form. For a different model, it is necessary only to change the density that appears at the end of the expresssion in (E-32). For a cross section instead of a panel, the product term in step 1 becomes simply the log of the single term.

The EM algorithm has an intuitive appeal in this (and other) settings. In practical terms, it is often found to be a very slow algorithm. It can take many iterations to converge. (The estimates in Example 16.16 were computed using a gradient method, not the EM algorithm.) In its favor,

AU: Confirm x-ref to Example 16.16

the EM method is very stable. It has been shown [Dempster, Laird, and Rubin (1977)] that the algorithm always climbs uphill. The log-likelihood improves with each iteration. Applications differ widely in the methods used to estimate latent class models. Adding to the variety are the very many Bayesian applications, none of which use either of the methods discussed here.

E.4 EXAMPLES

To illustrate the use of gradient methods, we consider some simple problems.

E.4.1 FUNCTION OF ONE PARAMETER

FIG E.4

First, consider maximizing a function of a single variable, $f(\theta) = \ln(\theta) - 0.1\theta^2$. The function is shown in Figure E.4. The first and second derivatives are

$$f'(\theta) = \frac{1}{\theta} - 0.2\,\theta,$$

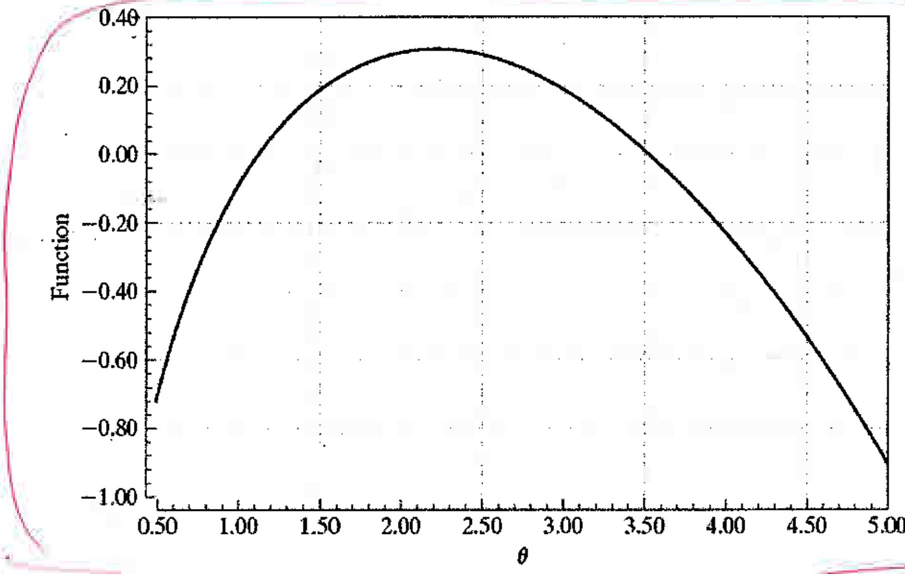
$$f''(\theta) = \frac{-1}{\theta^2} - 0.2.$$

FN 24

Equating f' to zero yields the solution $\theta = \sqrt{5} = 2.236$. At the solution, $f'' = -0.4$, so this solution is indeed a maximum. To demonstrate the use of an iterative method, we solve this problem using Newton's method. Observe, first, that the second derivative is always negative for any admissible (positive) θ.[24] Therefore, it should not matter where we start the iterations; we shall eventually find the maximum. For a single parameter, Newton's method is

$$\theta_{t+1} = \theta_t - [f_t'/f_t''].$$

FIGURE E.4 Function of One Variable Parameter.



[24] In this problem, an inequality restriction, $\theta > 0$, is required. As is common, however, for our first attempt we shall neglect the constraint.

TABLE E.1 Iterations for Newton's Method

Iteration	θ	f	f'	f''
0	5.00000	−0.890562	−0.800000	−0.240000
1	1.66667	0.233048	0.266667	−0.560000
2	2.14286	0.302956	0.030952	−0.417778
3	2.23404	0.304718	0.000811	−0.400363
4	2.23607	0.304719	0.0000004	−0.400000

TB E, 1

The sequence of values that results when 5 is used as the starting value is given in Table E.1. The path of the iterations is also shown in the table.

E.4.2 FUNCTION OF TWO PARAMETERS: THE GAMMA DISTRIBUTION

For random sampling from the gamma distribution,

$$f(y_i, \beta, \rho) = \frac{\beta^{\rho}}{\Gamma(\rho)} e^{-\beta y_i} y_i^{\rho-1}.$$

The log-likelihood is $\ln L(\beta, \rho) = n\rho \ln \beta - n \ln \Gamma(\rho) - \beta \sum_{i=1}^{n} y_i + (\rho - 1) \sum_{i=1}^{n} \ln y_i$. (See Section 16.6.4 and Examples 15.5 and 15.7.) It is often convenient to scale the log-likelihood by the sample size. Suppose, as well, that we have a sample with $\bar{y} = 3$ and $\overline{\ln y} = 1$. Then the function to be maximized is $F(\beta, \rho) = \rho \ln \beta - \ln \Gamma(\rho) - 3\beta + \rho - 1$. The derivatives are

AQ: Confirm x-refs to Sec 16.6.4 & Examples 15.5 and 15.7

$$\frac{\partial F}{\partial \beta} = \frac{\rho}{\beta} - 3, \qquad \frac{\partial F}{\partial \rho} = \ln \beta - \frac{\Gamma'}{\Gamma} + 1 = \ln \beta - \Psi(\rho) + 1,$$

$$\frac{\partial^2 F}{\partial \beta^2} = \frac{-\rho}{\beta^2}, \qquad \frac{\partial^2 F}{\partial \rho^2} = \frac{-(\Gamma\Gamma'' - \Gamma'^2)}{\Gamma^2} = -\Psi'(\rho), \qquad \frac{\partial^2 F}{\partial \beta\, \partial \rho} = \frac{1}{\beta}.$$

FN 25

TB E.2

Finding a good set of starting values is often a difficult problem. Here we choose three starting points somewhat arbitrarily: $(\rho^0, \beta^0) = (4, 1)$, $(8, 3)$, and $(2, 7)$. The solution to the problem is $(5.233, 1.7438)$. We used Newton's method and DFP with a line search to maximize this function.[25] For Newton's method, $\lambda = 1$. The results are shown in Table E.2. The two methods were essentially the same when starting from a good starting point (trial 1), but they differed substantially when starting from a poorer one (trial 2). Note that DFP and Newton approached the solution from different directions in trial 2. The third starting point shows the value of a line search. At this

TABLE E.2 Iterative Solutions to Max$(\rho, \beta)\rho \ln \beta - \ln \Gamma(\rho) - 3\beta + \rho - 1$

	Trial 1				*Trial 2*				*Trial 3*			
	DFP		*Newton*		*DFP*		*Newton*		*DFP*		*Newton*	
Iter.	ρ	β	ρ	β	ρ	β	ρ	β	ρ	β	ρ	β
0	4.000	1.000	4.000	1.000	8.000	3.000	8.000	3.000	2.000	7.000	2.000	7.000
1	3.981	1.345	3.812	1.203	7.117	2.518	2.640	0.615	6.663	2.027	−47.7	−233.
2	4.005	1.324	4.795	1.577	7.144	2.372	3.203	0.931	6.195	2.075	—	—
3	5.217	1.743	5.190	1.728	7.045	2.389	4.257	1.357	5.239	1.731	—	—
4	5.233	1.744	5.231	1.744	5.114	1.710	5.011	1.656	5.251	1.754	—	—
5	—	—	—	—	5.239	1.747	5.219	1.740	5.233	1.744	—	—
6	—	—	—	—	5.233	1.744	5.233	1.744	—	—	—	—

[25] The one used is described in Joreskog and Gruvaeus (1970).

starting value, the Hessian is extremely large, and the second value for the parameter vector with Newton's method is (−47.671, −233.35), at which point F cannot be computed and this method must be abandoned. Beginning with $\mathbf{H} = \mathbf{I}$ and using a line search, DFP reaches the point (6.63, 2.03) at the first iteration, after which convergence occurs routinely in three more iterations. At the solution, the Hessian is $[(-1.72038, 0.191153)', (0.191153, -0.210579)']$. The diagonal elements of the Hessian are negative and its determinant is 0.32574, so it is negative definite. (The two characteristic roots are −1.7442 and −0.18675). Therefore, this result is indeed the maximum of the function.

E.4.3 A CONCENTRATED LOG-LIKELIHOOD FUNCTION

There is another way that the preceding problem might have been solved. The first of the necessary conditions implies that at the joint solution for (β, ρ), β will equal $\rho/3$. Suppose that we impose this requirement on the function we are maximizing. The **concentrated** (over β) **log-likelihood function** is then produced:

$$\begin{aligned} F_c(\rho) &= \rho\ln(\rho/3) - \ln\Gamma(\rho) - 3(\rho/3) + \rho - 1 \\ &= \rho\ln(\rho/3) - \ln\Gamma(\rho) - 1. \end{aligned}$$

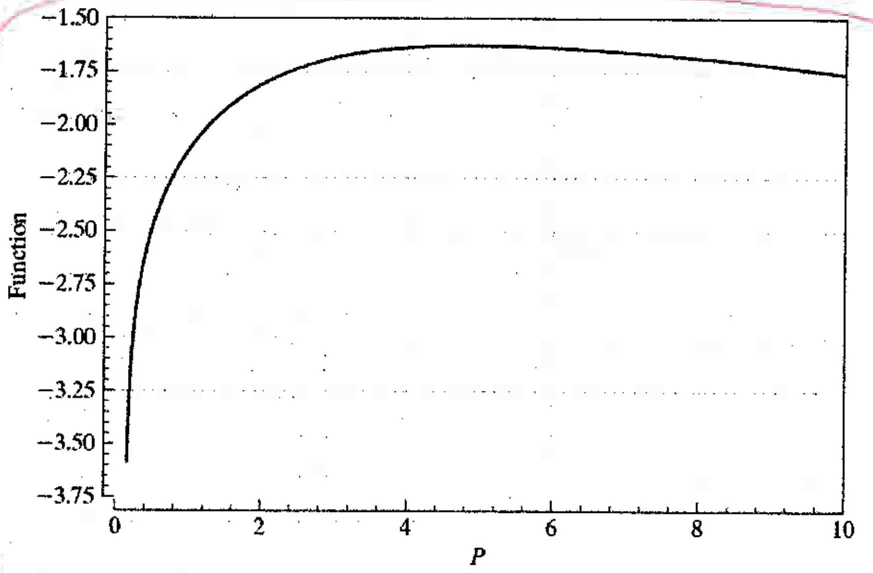
FIG E.5

This function could be maximized by an iterative search or by a simple one-dimensional grid search. Figure E.5 shows the behavior of the function. As expected, the maximum occurs at $\rho = 5.233$. The value of β is found as $5.23/3 = 1.743$.

The concentrated log-likelihood is a useful device in many problems. (See Section 16.9.6.c for an application.) Note the interpretation of the function plotted in Figure E.5. The original function of ρ and β is a surface in three dimensions. The curve in Figure E.5 is a projection of that function; it is a plot of the function values above the line $\beta = \rho/3$. By virtue of the first-order condition, we know that one of these points will be the maximizer of the function. Therefore, we may restrict our search for the overall maximum of $F(\beta, \rho)$ to the points on this line.

AU: Confirm x-ref to Sec 16.9.6c

FIGURE E.5 Concentrated Log-Likelihood.


Function
P