COMPUTATIONAL ECONOMICS

Macroeconomic Modeling in GAMS

David A. Kendrick
Ruben P. Mercado
Hans M. Amman

1 Introduction

Macroeconomic models study the behavior of economic systems from an aggregate point of view. They try to capture the interdependence among consumption and investment expenditure, fiscal and monetary policy variables, the price level, the aggregate supply, and the level of employment. From a modeling perspective, we can say that there are three main classes of macroeconomic models: standard models, rational expectations models, and intertemporal optimization models.

Standard models like the one used in this chapter, also known as IS-LM models, specify aggregate relationships to explain the behavior of macroeconomic variables. They also usually assume that economic agents form expectations in an adaptive way. Rational expectations models also work with aggregate relationships, but they assume that the economic agents display forward-looking behavior. That is, in order to form expectations, those agents are assumed to make use of all the available information, including the model of the economy that policy makers use to model their behavior. Finally, intertemporal optimization models share with rational expectations models the same assumptions in connection with expectation formation, but try to base their modeling of macroeconomic behavior on more explicit *micro foundations*.

This chapter draws extensively on both the verbal and the mathematical development in Mercado et al. (1998). Kluwer Academic Publishers have kindly granted us permission to reuse here substantial materials from our previously published paper.

IS-LM models are the backbone of almost all introductory and intermediate macroeconomics textbooks and have long been the main workhorses in empirical macroeconomics, as is the case, for example, of the Fair model at `http://fairmodel.econ.yale.edu/`.[1] An example of a well-known rational expectations model is the Taylor (1993) model. Intertemporal optimization models are still relatively small and are not used very much in large-scale empirical applications or policy analyses, but are they used for teaching at the graduate level, for experimental purposes, and for policy analysis exercises on a relatively small scale. One of the most influential models of this type is the one by Rotemberg and Woodford (1997).

The solution methods of the foregoing models critically depend on the assump-

[1] The antecedents of these models go back to the work of Keynes (1936) and Hicks (1937).

tion regarding expectations formation. For example, models with backward-looking expectations, like those in the standard-type IS-LM model to be presented in this chapter, are solved using a given set of initial conditions for the lagged variables and paths for policy and exogenous variables. As we will see later in the book, this is not the case for rational expectations and intertemporal optimization models, since they share the assumption of forward-looking behavior and present what is technically known as *two-point boundary value problems*. Solving them requires both initial and terminal conditions or specific iterative procedures.

2 The Hall and Taylor Model

In this chapter we introduce the Hall and Taylor (citeyearhall1997) model, a well-known textbook standard, and illustrate how to represent and simulate it in GAMS. This is a twelve-equation nonlinear dynamic model for an open economy with flexible exchange rates and is well suited for teaching simulation and policy analysis at the undergraduate level. The core of this model can be seen as a standard IS-LM-OPEN ECONOMY SUBMODEL for the aggregate demand of the economy together with an *expectations augmented* Phillips curve, that is, the aggregate supply. The Hall and Taylor model contains the following equations, variables, and parameters

Equations of the Hall and Taylor model

- IS-LM

$$\begin{aligned}
\text{GDP identity:} \quad Y &= C + I + G + X & (1)\\
\text{Disposable income:} \quad Y^d &= (1-t)\,Y & (2)\\
\text{Consumption:} \quad C &= a + bY^d & (3)\\
\text{Investment:} \quad I &= e - dR & (4)\\
\text{Money demand:} \quad M/P &= kY - hR & (5)
\end{aligned}$$

- Expectations Augmented Phillips Curve

$$\begin{aligned}
\text{Expected inflation:} \quad \pi^e &= \alpha\pi_{-1} + \beta\pi_{-2} & (6)\\
\text{Inflation rate:} \quad \pi &= \pi^e + f\left(\frac{Y - Y_N}{Y_N}\right) & (7)\\
\text{Price level:} \quad P &= P_{-1}\,(1+\pi) & (8)
\end{aligned}$$

- Foreign Account

$$\begin{aligned}
\text{Real exchange rate:} \quad EP/P_W &= q + vR & (9)\\
\text{Net exports:} \quad X &= g - mY - nEP/P_W & (10)
\end{aligned}$$

- Government Deficit and Unemployment

$$\begin{aligned}
\text{Government deficit:} \quad G_d &= G - tY & (11)\\
\text{Unemployment rate:} \quad U &= U_N - \mu\left(\frac{Y - Y_N}{Y_N}\right) & (12)
\end{aligned}$$

Variables of the Hall and Taylor model

- Endogenous Variables (b) Policy Variables

$$\begin{aligned} C &= \text{Consumption} \\ G &= \text{Government expenditure} \\ E &= \text{Nominal exchange rate} \\ M &= \text{Money stock} \end{aligned}$$

- foreign currency and domestic currency

$$\begin{aligned} G_d &= \text{Government deficit} \\ I &= \text{Investment} \\ P &= \text{Domestic price level} \\ R &= \text{Real interest rate} \end{aligned}$$

- Exogenous Variables

$$\begin{aligned} U &= \text{Unemployment rate} \\ P_w &= \text{Foreign price level} \\ X &= \text{Net exports} \\ U_N &= \text{Natural rate of unemployment} \\ Y &= \text{GDP} \\ Y_N &= \text{Potential GDP} \\ Y^d &= \text{Disposable income} \end{aligned}$$

- prices

$$\begin{aligned} \pi &= \text{Inflation rate} \\ \pi^e &= \text{Expected inflation} \end{aligned}$$

Parameters of the Hall and Taylor model

a = 220; b = 0.7754; d = 2000; e = 1000; f = 0.8; g = 600; h = 1000; k = 0.1583; m = 0.1; n = 100; q = 0.75; t = 0.1875; v = 5; α =0.4; β =0.2; μ = 0.33

The model is dynamic—all variables without subscripts correspond to time t, those with –1 subscripts correspond to $t - 1$, and so on. In addition, the model is nonlinear—nonlinearities appearing in equations (5), (8), (9), and (10). As we will see later, its dynamic behavior displays the *natural rate* property: nominal shocks may affect real variables in the short run, but not in the long run.

Equations (1) to (5) are standard in most macroeconomics textbooks. Equation. (1) is an identity that states that GDP always equals the sum of its main components: consumption, investment, government spending, and net exports (exports minus imports); equation (2) determines disposable income as equal to GDP net of taxes; equation (3) is a standard consumption function in which current consumption depends on current income; equation (4) determines investment as an inverse function of the real interest rate; and equation (5) defines real money balances as a positive function of income (money demand for transaction purposes) and a negative function of the interest rate (the opportunity cost of holding money instead of interest-bearing assets).

Equations (6) to (8) correspond to an expectations augmented Phillips curve: equation (6) gives the expected inflation as a function of the past inflation in the last two periods (years), and equation (7) determines the inflation rate as a positive function of the expected inflation rate and the GDP gap (the difference between actual and the potential GDP in the previous year). A positive gap means an overheated economy, thus inflationary pressure. A negative gap means recession, thus deflationary pressure. Equation (8) just defines the price level as a function of the price level the previous year and the inflation rate.

Equations (9) and (10) are foreign account equations. Note that the nominal exchange rate E is defined as foreign currency/domestic currency. Thus an increase (decrease) in E is a nominal appreciation (depreciation) of the domestic currency. Equation (9) determines the real exchange rate (the nominal exchange rate times the domestic price level divided by the foreign price level) as a positive function of the interest rate. Thus, for example, an increase in the U.S. interest rate (implicitly assuming that the interest rate in the rest of the world remains the same) would cause capital inflows and an appreciation of the dollar. Equation (10) gives net exports as a function of GDP and the real exchange rate. Changes in GDP affect the demand for imports, whereas exports do not change as much. Thus net exports change. The real exchange rate is the relative price between domestic and foreign products, so its changes affect imports and exports. Finally, equations (11) and (12) give the government deficit and the unemployment rate, and they have no feedback on the rest of the model.

It is usual to develop a compact graphical representation of a model like this in two graphs: and IS-LM graph and an aggregate demand–aggregate supply graph. To derive the IS schedule, we substitute equation (2) into equation (3), equation(9) into equation (10), and then equations (3), (4), and (10) into equation (1). Solving the resulting equation for the interest rate, we obtain

$$R = \frac{a+e+g-nq}{d+nv} - \frac{1-(b-t)+m}{d+nv}Y + \frac{1}{d+nv}G \tag{13}$$

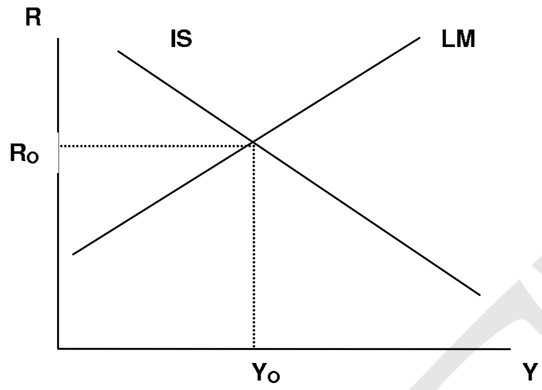
Equation (13) shows R as a function of Y(given G) and represents all the combinations of interest rate and income for which spending balances. To derive the LM schedule we just solve for R in equation (5) to obtain

$$R = \frac{k}{h}Y - \frac{1}{h}\frac{M}{P} \tag{14}$$

Equation (12) also shows R as a function of Y (given M and P) and represents all the combinations of interest rate and income for which the money market is in equilibrium. Finally, the graphical representation of both schedules in the (R, Y) space is shown in Figure 1. Given the model coefficient values, the IS curve is downward sloping and the LM curve is upward sloping. The intersection of the two schedules determines the equilibrium interest rate and income.

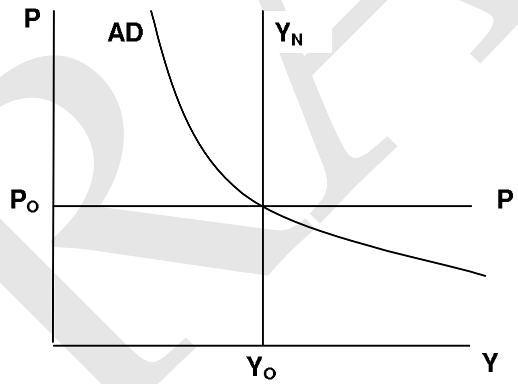
The aggregate demand (AD) schedule represents the IS-LM part of the model in a different space, the price level (P) and income (Y) space, and shows how much people demand at a given level of prices. It can be obtained combining equation (13) and (14), and the result, given the values of the model coefficients, is a downward-sloping nonlinear schedule with P as a function of Y, as shown in Figure 13.2. The aggregate supply is an expectations augmented Phillips curve embodied in equation (6), (7), and (8). To capture its behavior, we represent it in the (P, Y) space by means of two lines. The Y_N vertical line represents the long-run aggregate supply that is the potential or *natural* income level, which is assumed to be constant in the short run. Finally, the horizontal line or *price line* (P) represents the short-run aggregate supply,

Figure 1: IS-LM graph



which is supposed to be perfectly elastic, though in other textbook presentations it is assumed to be upward sloping. Figure 2 shows the graphical representation of aggregate demand and supply.

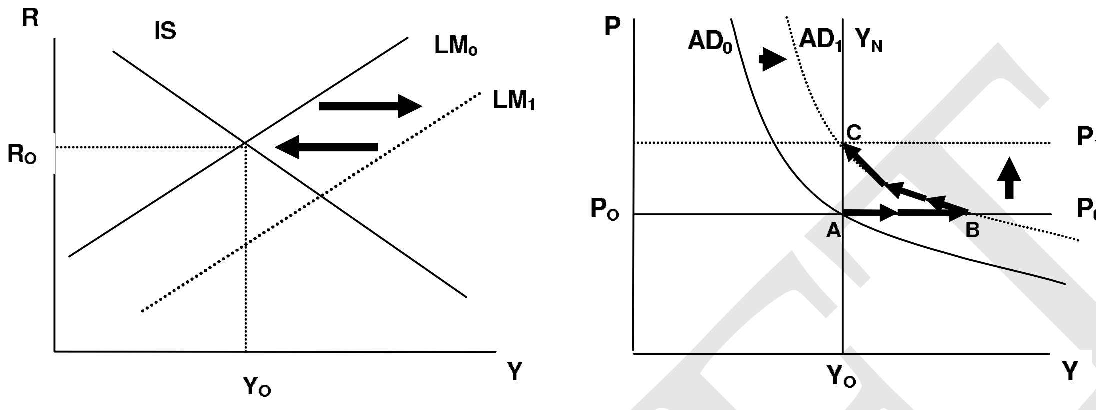
Figure 2: Aggregate demand – aggregate supply



The analysis of the effects of an increase in the money supply (M) helps us to understand the workings of the model in qualitative terms. An increase in the money supply brings about disequilibrium in the money market, shifting the LM schedule to the right, thus bringing down R and increasing Y. This implies that the AD schedule also shifts to the right, as is shown in Figure 3.

In the short run prices are sticky, so the economy moves from point A to point B. However, in the medium run, since there is a positive GDP gap, the

Figure 3: Qualitative effects of an increase in the money supply.



inflation rate becomes positive and prices begin to increase, as can be seen in the equation for the inflation rate

$$\pi = \pi^e + f\left(\frac{Y_{-1} - Y_N}{Y_N}\right) \qquad \text{[equation (7)]}$$

This process continues given that agents' expectations change owing to past changes in the inflation rate, as shown in the equation for expected inflation:

$$\pi^e = \alpha\pi_{-1} + \beta\pi_{-2} \qquad \text{[equation (6)]}$$

As prices increase, real money balances decrease (see the following equation for money demand) shifting the LM schedule to the left:

$$M/P = kY - hR \qquad \text{[equation (5)]}$$

Finally, the economy moves from point B to point C. We can see then that the increase in the money supply was neutral in the long run with respect to real variables, but not in the short run.

3 The Hall and Taylor Model in GAMS

Different strategies can be followed when one is confronted with the problem of solving and performing policy experiments with a model like Hall and Taylor's. In the following, we review some of them.

Usually, the first step in the analysis of this kind of model is to find the steady-state values of the endogenous variables for a given set of constant values of the policy and exogenous variables. This requires transformation of the model from dynamic to static. Solving a nonlinear system of equations, even when it is static, is not easy. In general, we have to rely on numerical techniques that may or may not deliver a solution, even if it exists, depending on the initial conditions provided. However, the model we are interested in does not contain many or very strong nonlinearities, making the task of finding a solution relatively easy.

To solve for the steady state, we have to eliminate all the time subscripts and solve the resulting static nonlinear model. This does not present any challenge to GAMS users, even for beginners. Since this model is relatively straightforward we do not discuss it further here, but rather turn our attention to the dynamic nonlinear model that is of greater interest. The file for this model is htsim.gms on the web site. It is also contained in Appendix A at the end of this chapter. We discuss here in the body of the chapter two unusual aspects of the GAMS representations of this model. However, before doing so it is useful to look at the main SET specification of the model, namely

```
SETS T EXTENDED HORIZON / 0*15 /
```

Thus the model includes sixteen time periods: zero, one, two through fifteen. Also keep in mind that GAMS is not case specific and one finds the set of time periods specified in the GAMS statement at times as T and at other times as t; however, they are the same.

Next we consider the way the dynamic variables and equations of the Hall and Taylor model are represented in GAMS. This is shown below. Note that to avoid notational conflicts in the GAMS statement, the mathematical parameters e, g, m, and t have been renamed as ee, gg, mm, and tax, respectively. Variables and parameters with names denoted with Greek symbols in the mathematical statement of the model are also renamed in the GAMS statement, since GAMS cannot handle such notation. Finally, the following listing does not include all the variable names or equations names that are in the GAMS version of the model, but rather only a few; it does, however, contain all the equations:

```
VARIABLES
Y(t) gdp
Yd(t) disposable income

...
```

```
EQUATIONS

eq1(t) gdp identity
eq2(t) disposable income

...;

eq1(t+2).. Y(t+2) =E= C(t+2) + I(t+2) + G(t+2) + X(t+2) ;
eq2(t+2).. Yd(t+2) =E= (1 - tax) * Y(t+2) ;
eq3(t+2).. C(t+2) =E= a + b * Yd(t+2) ;
eq4(t+2).. I(t+2) =E= ee - d * R(t+2) ;
eq5(t+2).. M(t+2) / P(t+2) =E= k * Y(t+2) - h * R(t+2) ;
eq6(t+2).. piex(t+2)=E= alpha * pi(t+1) + beta * pi(t) ;
eq7(t+2).. pi(t+2) =E= piex(t+2) + f*(Y(t+1)-Yn(t+2))/Yn(t+2) ;
eq8(t+2).. P(t+2) =E= P(t+1) * (1 + pi(t+2)) ;
eq9(t+2).. E(t+2) * P(t+2) / Pw(t+2) =E= q + v * R(t+2) ;
eq10(t+2).. X(t+2) =E= gg - mm*Y(t+2) - n*(E(t+2)*P(t+2)/Pw(t+2));
eq11(t+2).. Gd(t+2) =E= G(t+2) - tax * Y(t+2) ;
eq12(t+2).. U(t+2) =E= Un(t+2) - mu*(Y(t+2)-Yn(t+2))/Yn(t+2) ;\\
```

Note that all the variables and equations are defined over the set `t`. However, the model equations are specified over the set $t+2$ and contain variables defined over the sets $t+2$, $t+1$, and t, instead of following the corresponding original indices `t`, `t-1`, and `t-2`, respectively. This is due to the way in which GAMS handles the assignment of values to lagged variables.

For example, we could define the set t as

```
SETS t /0,1,2,3/
```

and then write equation (6) with time subscripts as in its original formulation:

```
eq6(t).. piex(t) =E= alpha * pi(t-1) + beta * pi(t-2) ;
```

Then, when solving the model, GAMS would assign the default value zero to expressions like `pi(t-1)` and `pi(t-2)`, since -1 and -2 do not belong to the set `t`. Therefore, we would not be able to assign initial values other than zero to the inflation rate, even if we wished to do so.

Thus, when dealing with models containing lagged variables in GAMS, we go according to the following rule of thumb: for a solution horizon of duration `t`, specify equations starting from the longest lag. In Hall and Taylor's model, the longest lag is equal to 2. Note how we wrote the model equations containing lags—equation (6), (7), and (8)—where we have variables with subscripts equal to `t`, `t+1`, and `t+2`. At the same time, in equations containing no lags, all variables have subscripts equal to `t+2`. By operating in this way we *keep* the first two time periods (t and t+1) free to assign initial values and let GAMS find a solution for the remaining periods. More details on this are provided in Appendix B at the of the book.

To complete the GAMS specification of Hall and Taylor's model, apart from defining—as we did earlier—the extended horizon for simulations, we have to provide initial conditions for output and inflation:

```
SETS t EXTENDED HORIZON / 0*15 /
t0(t) PERIOD ZERO
t1(t) PERIOD ONE;
t0(t) = YES{\$}(ORD(t) EQ 1);
t1(t) = YES{\$}(ORD(t) EQ 2);
```

With this specification, we are defining a fifteen-period time index as the set `t`. Then, we declare and define the subsets `t0` and `t1` and assign to them, respectively, the first and second elements of the t set, that is, the elements in the ordinal `1` and ordinal `2` places. Thus the GAMS statement

```
t0(t) = YES$(ORD(t) EQ 1);
```

can be read as *assign to the set t0 the elements of the set t such that the ordinal position of element t is equal to one*. The $ operator in GAMS can be read as a *such that* operator in this context.

The specification for the sets t0 and t1 used earlier is useful in case one decides to change the extension of the simulation horizon, since we would not have to change the definition of the initial conditions subsets.

In the same way, we can also define terminal condition subsets. These conditions become necessary in models containing rational expectations, as we will see later in the book. For instance, terminal conditions for the last and the previous-to-the-last period can be written by defining two new subsets—for example, `tf(t)` and `tf1(t)`—of the set t and then adding the following two expressions:

```
tf(t) = YES$(ORD(t) EQ CARD(t));
tf1(t)= YES$(ORD(t) EQ (CARD(t) - 1));
```

where, as before, `ORD(t)` means ordinal and `CARD(t)` means the cardinality, that is, the number of elements in the set.

Next we turn our attention from the specification of the dynamics of the model in GAMS to the specification of the policy variable time paths. This is unusual in that the policy variables are specified in percent deviations from base levels rather than in levels. This is accomplished by providing statements that set the percent difference. An example is the statements that are used for monetary policy,:

```
SETS
TS1(T) periods for shock 1 / 4*15 / ;
```

which create a set TS1 over which the policy change is defined, and

```
Mper(TS1) = 0.0 ;
```

which sets the percentage change. Thus to create a solution where the money supply is 3 percent above the base level in periods 4 through 15 one would modify the foregoing statement to:

```
Mper(TS1) = 0.03 ;
```

Alternatively, the user might want to have two periods in which the policies were above and then below the base level. This would be done by first creating the two sets of time periods with GAMS statements of the form

```
SETS
TSPER1(T) Quarters in period 1 / 5*8 /
TSPER2(T) Quarters in period 2 / 10*13 / ;
```

followed by statements to set the percent deviations:

```
Mper(TSPER1) = 0.03 ;
Mper(TSPER2) = -0.02 ;
```

Then the money supply would be 3 percent above the base level in quarters 5 through 8 and 2 percent below the base level in quarters 10 through 12. However, when doing this be careful not to use quarters beyond those included in the set `T`.

The initial conditions for output and inflation are defined as

```
Y.fx(t1) = ini1; Pi.fx(t0) = ini2; Pi.fx(t1) = ini3;
```

where `t0` and `t1` mean period 0 and period 1, respectively; `.fx` tells GAMS to keep the assigned values fixed during the execution of the program; and `ini1` to `ini3` are given initial values.

In this model, in order to solve a system of equations in GAMS, it is necessary to include an additional variable `(J)` and an additional equation `(JD)` and to maximize or minimize the added variable. Thus the `SOLVE` statement is

```
SOLVE NONLDYN MINIMIZING J USING NLP;
```

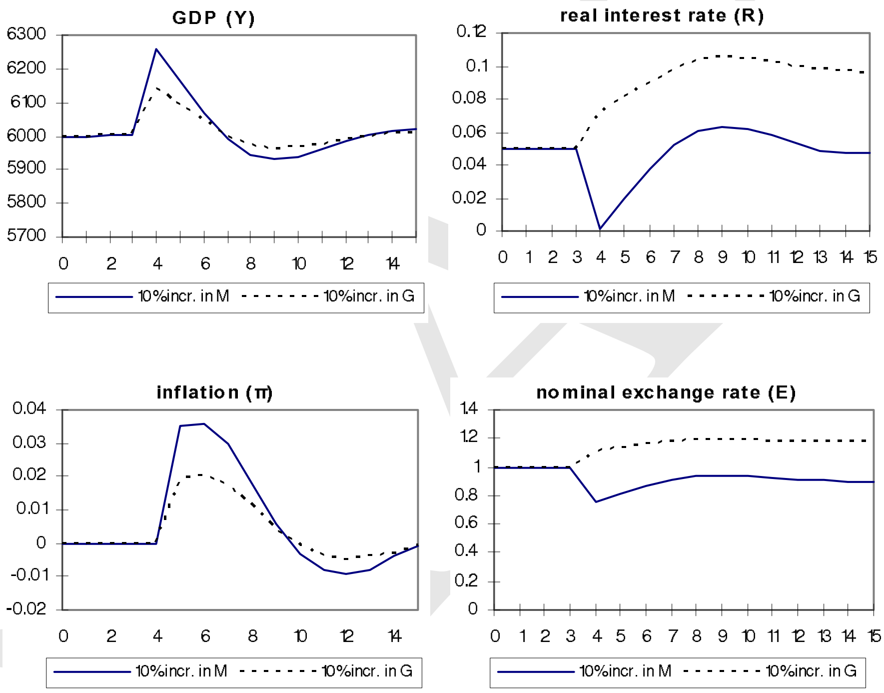
Moreover, since the model contains indexed equations, a stacking method is used in GAMS, as discussed in Appendix C. Finally, since the Hall and Taylor model is nonlinear, we have to invoke a nonlinear programming (NLP) solver. For an introduction to this type of solver see Appendix D.

To perform simulations with this model we change the values of the policy variables or the parameter values, as discussed previously, and compare the different dynamic solution paths obtained for the endogenous variables.

The graphical analysis we performed earlier gave us a useful representation of the qualitative behavior of the key variables of the economy. However, to deal with more variables and to obtain precise quantitative results, we have to simulate the model computationally. Figure 13.4 displays the results of two experiments: a first experiment where we start from an equilibrium position and then increase the money supply by 10 percent and a second where we start from equilibrium and increase government expenditure by 10 percent. We assume that both increases take place in period four and are permanent; that is, once they happen they are not reversed. Figure 13.4 shows the solution paths for income, the inflation rate, the interest rate, and the nominal exchange rate. The value of the variables between periods zero and three corresponds to the model's steady-state values. The continuous line corresponds to the money supply experiment, while the dotted line corresponds to the government expenditure experiment. GDP values are in billions of dollars. For the real interest rate and the inflation rate a value of 0.01 corresponds to 1 percent. The nominal exchange rate values correspond to an index value set equal to one in the steady state.

We can observe how, as expected, the change in money supply has short-run but no long-run real effects, whereas the change in government expenditure has short- and long-run real effects. We can also see how the trajectories to the new

Figure 4: Effetc of a 10 percent increase in the money supply and in government expenditure.



equilibrium positions are oscillatory, with temporary over- and undershooting of the final equilibrium positions.

The essential elements of the function of monetary policy can be seen in the results in Figure 13.4. Consider the case where money supply is increased by 10 percent (the solid lines). This has the effect at first of decreasing the interest rate, as in shown in the upper-right diagram. The decrease in the interest rate in turn causes an increase in investment and therefore GDP, as shown in the upper-left-hand graph. As GDP increases above potential, inflation increases, as shown in the bottom-left-hand graph. The increase in inflation raises the price level, which has the effect of decreasing the real money supply in the money demand equation,

$$M/P = kY - hR \tag{15}$$

This in turn causes an increase in the real interest rate beginning in period five as shown in the upper-right-hand graph. The rise in the interest rate then decreases investment and GDP begins to fall in period five as shown in the upper-left-hand graph. This oscillatory process continues until GDP returns to the potential GDP level and inflation returns to zero.

In the GAMS program htsim.gms there are also ways of changing more policy or exogenous variables to perform other experiments. For example, you can simulate a change in potential GDP or a change in the foreign price level.[2] You might also want to change the tax rate, which is defined in the program as a scalar, or any other model parameter.[3]

[2]If you change the foreign price level, you will note that the nominal exchange rate also changes in an opposite and neutralizing way so that nothing else happens. From equation (9) we know that the real exchange rate is determined by the interest rate. We also know that the domestic price level is sticky in the short run. Thus a change in the foreign price level has to be compensated by a change in the nominal exchange rate. You will observe a similar behavior, but in the long run, in the case of a change in the money supply. Since this change affects the domestic price level but not the real interest rate in the long run, the nominal exchange rate changes to compensate for the change in the domestic price level. Only in the case of a permanent change in the real interest rate (i.e., owing to a change in government spending) will the nominal exchange rate and the domestic price level not move in a compensatory way.

[3]Hall and Taylor's textbook comes with a black box software named Macrosolve that allows you to perform experiments with the model changing some policy or exogenous variables. The GAMS program presented in this chapter replicates many results from Macrosolve. A change in the tax rate, since it is a model parameter, changes the steady-state solution of the model, as would be the case with any other model parameter, such as, for example, the marginal propensity to consume.. However, for the particular experiment of changing the tax rate,

Having learned how to perform model simulations, we can now move to the realm of optimal policy analysis, which in a way is the reverse of simulation. Instead of determining the paths of the endogenous variables given values for the policy variables, we now want to determine the optimal path for the policy variables given target paths and relative weights for target variables. This can easily be done by adding a loss function as an extra equation to the model and redefining the policy variables of interest as endogenous variables. For example, in the earlier GAMS statement, we can substitute the following quadratic loss function for the previous JD equation and the Loss variable for the previous J variable, that is,

```
eqLoss.. Loss =E= 0.5 * sum(t, Wy * POWER((Y(t)-Ytar(t)), 2 )
               + Wp * POWER((P(t)-Ptar(t)), 2 );
```

where Ytar and Ptar are prespecified target values for output and the price level and Wy and Wp are weights on the deviations from target values of output and the price level, respectively.

Since the variables entering the loss function (GDP and the price level) are measured in different units, it is convenient to impose some normalization on the weights. For instance, if `Ytar` is 6000 and Ptar is 1, then to penalize deviations from target equally we could set Wy equal to 1 and then obtain the corresponding normalized Wp as

$$\mathrm{Wp} = 6000^2/1^2 = 3600000$$

Then, if we decide to penalize deviations from Ytar twice as much as for deviations from Ptar, we choose `Wy` = 2 and `Wp` = 3600000, or `Wy` = 1 and `Wp` = 1800000, and so on. For a full discussion of weighting procedures see Park (1997).

If we now redefine, for example, the money supply $M(t)$ as an endogenous variable and ask GAMS to solve the model minimizing the variable Loss, we obtain the corresponding optimal path for $M(t)$. This is a typical and basic experiment in policy analysis, but it can be made more sophisticated in a variety of ways, for example, by introducing stochastic elements and learning mechanisms. To do so, it may be convenient to move from GAMS to a more specialized software such as Duali. We do that later in this book.

Macrosolve gives steady-state invariant results. Our GAMS program does not. Thus, for that particular experiment, in case you wish to compare results, you will find that they differ. Note that there is nothing wrong in one case or the other, just two different simulation methods.

4 Experiments

In this chapter we simulated the effects of permanent changes in the money supply and in government expenditure. You may want to simulate temporary changes, that is, changes that last for only a few periods. To become familiar with the dynamics of the Hall and Taylor model, you should continue performing simulations of shocks to the model's exogenous variables, that is, potential GDP or the foreign price level, asking yourself if the observed effects make economic sense.

You may want to expand the model allowing for shocks to the domestic price level. This price shock may have different sources: changes in the price of an input to the economy (e.g., oil), a wage increase passed on by firms in the form of increased prices, and so on. You can represent this as an exogenous variable Z added to equation (7) so that it becomes

$$\pi = \pi^e + f\left(\frac{Y_{-1} - Y_N}{Y_N}\right) + Z$$

Thus, this shock is a shift factor in the short-run aggregate supply or horizontal price line. Note that to introduce this new variable in the GAMS program properly you have to define it as a parameter in the same fashion as we did potential GDP or the foreign price level and add it to the corresponding equation. You may want to try experiments in which this variable changes only temporarily. Note also that this variable is implicitly defined in percentage changes and not in levels.

Further, you may try to introduce changes in the model policy variables in order to counteract shocks to exogenous variables to bring the economy back to the initial equilibrium position, particularly in connection with the values of real variables. This is a rudimentary but useful way of undertaking policy analysis. Finally, you may want to perform a more sophisticated policy analysis shocking the economy with diverse shocks and working with a loss function as suggested at the end of this chapter, or you may decide to move on to Chapter **??**, where that kind of analysis is performed with a more specialized software.

References

Hicks, J. R.: 1937, Mr. keynes and the classics: A suggested interpretation, *Econometrica* **5**, 147–159.

Judd, K. L.: 1998, *Numerical methods in economics*, MIT Press, Cambridge, Massachussets.

Keynes, J.: 1936, *The General Theory of Employment, Interest and Money*, Macmillan Press, London, United Kingdom.

Mercado, P. R., Kendrick, D. A. and Amman, H. M.: 1998, Teaching macroeconomics with gams, *Computational Economics* **12**, 125–149.

Miranda, M. J. and Fackler, P. L.: 2002, *Applied Computational Economics and Finance*, MIT Press, Cambridge, Massachussets.

Park, H. J.: 1997, *A Control Theory Analysis of Macroeconomic Policy Coordination by the US, Japan and Korea*, PhD thesis, Department of Economics, University of Texas, Austin, Texas, USA.

Rotemberg, J. and Woodford, M.: 1997, An optimization-based econometric model for the evaluation of monetary policy, *NBER Macroeconomic Annual* **12**, 297–346.

Taylor, J. B.: 1993, *Macroeconomic Policy in a World Economy*, W.W. Norton, New York.

Appendices

A Hall and Taylor in GAMS

```
$TITLE   htsim:   HALL-TAYLOR SIMULATION
* Developed by Ruben Mercado

OPTION SYSOUT = OFF;
OPTION LIMROW = 7;
OPTION LIMCOL = 0;
OPTION SOLPRINT = OFF;
$OFFSYMXREF OFFSYMLIST
***********************************************************************
* SECTION 1 :  DEFINITION OF PARAMETER VALUES FOR THE ORIGINAL
*                           NONLINEAR  HALL-TAYLOR MODEL
***********************************************************************
SCALARS
a       minimum consumption             / 220 /
b       marg prop to consume            / 0.7754 /
d       interest elast of invest.       / 2000 /
ee      maximum  investment             / 1000 /
f       coeff. on excess aggr dem.      / 0.8 /
gg      maximum net exports             / 600 /
h       interest elast  of mon  dem.    / 1000 /
k       income elast  of money dem.     / 0.1583 /
mm      income elast  of net exp        / 0.1 /
n       real ex  rate elast  of net exp / 100 /
q       constant                        / 0.75 /
tax     tax rate                        / 0.1875 /
v       constant                        / 5 /
alpha   coeff. on 1 lagged inflation    / 0.4 /
beta    coeff. on 2 lagged inflation    / 0.2 /
mu      elast. of empl. wrt GDP         / 0.33 /  ;

***********************************************************************
*  SECTION 2:     DEFINITION OF TEMPORAL HORIZON FOR SIMILATION
***********************************************************************
* If you change the extension of the horizon, make the necessary
* adjustments in the section of shocks' definition  (Section 3)

SETS  T    EXTENDED HORIZON     / 0*15 /
       T0(T) PERIOD ZERO
       T1(T) PERIOD ONE ;

       T0(T) = YES$(ORD(T) EQ 1);
       T1(T) = YES$(ORD(T) EQ 2);
       DISPLAY T0, T1;
***********************************************************************
* SECTION  3 : DEFINITION OF CHANGES IN POLICY AND EXOGENOUS VARIABLES
***********************************************************************
PARAMETERS
*  definition of policy and exogenous variables (in percentage changes)
Mper(T)        money stock  (in % change)
Gper(T)        Gov. expenditure (in % change)
Ynper(T)       potential GDP (in % change)
Pwper(T)       foreign prices (in % change)
```

```
*  definition of policy and exogenous variables (in levels)
M(T)     money stock (in levels)
G(T)     Gov. expenditure (in levels)
Yn(T)    potential  GDP (in levels)
Pw(T)    foreign prices (in levels)  ;
*  default values for policy and exogenous variables
Mper(T) = 0.3 ;   Gper(T) = 0 ;  Ynper(T) = 0 ;  Pwper(T) = 0 ;
M(T) = 900 ;    G(T) = 1200 ;  Yn(T) = 6000 ;  Pw(T) = 1 ;
******************************
* CHANGE IN MONEY SUPPLY
******************************
SETS
TS1(T) periods for shock 1    / 4*15 / ;
Mper(TS1) = 0.0 ;
****************************************
* CHANGE IN GOVERNMENT EXPENDITURE
****************************************
SETS
TS2(T) periods for shock 2    / 4*15 / ;
Gper(TS2) = 0.0;
***********************************************************
*  CHANGE IN POTENTIAL GNP (notice that the natural rate of
*                          unemployment remains the same)
***********************************************************
SETS
TS3(T) periods for shock 3    / 4*15 / ;
Ynper(TS3) = 0.0;
****************************
*  CHANGE IN FOREIGN PRICES
****************************
SETS
TS4(T) periods for shock 4    / 4*15 / ;
Pwper(TS4) = 0.0;
*  Transformation of shocks in % changes into shocks in levels
M(TS1) =   900 * (1 + Mper(TS1)) ;
G(TS2) =  1200 * (1 + Gper(TS2)) ;
Yn(TS3) = 6000 * (1 + Ynper(TS3)) ;
Pw(TS4) =    1 * (1 + Pwper(TS4)) ;


* reporting policy and exogenous variables values
PARAMETER REPORTEX  POLICY AND EXOGENOUS VARIABLES VALUES;
  REPORTEX(T,"Money") = M(T);
  REPORTEX(T,"Gov. Exp.") = G(T);
  REPORTEX(T,"Pot. GDP") = Yn(T);
  REPORTEX(T,"Fgn Price") = Pw(T);

***********************************************************************
* SECTION 4:  COMPUTATION OF SOLUTION
***********************************************************************
PARAMETERS
Un(T)          natural rate of unemployment ;
Un(T) = 0.05 ;

VARIABLES
Y(T)          gdp
Yd(T)         disposable income
C(T)          consumption
I(T)          investment
R(T)          interest rate
P(T)          price level
pi(T)         inflation rate
```

```
piex(T)       expected inflation rate
E(T)          nominal exchange rate
X(T)          net exports
Gd(T)         government deficit
U(T)          unemployment rate
J             performance index

EQUATIONS
eq1(T)      gdp identity
eq2(T)      disposable income
eq3(T)      consumption
eq4(T)      investment
eq5(T)      money demand
eq6(T)      expected inflation
eq7(T)      inflation rate
eq8(T)      price level
eq9(T)      real exchange rate
eq10(T)     net exports
eq11(T)     government deficit
eq12(T)     unemployment rate
JD          performance index ;

JD..         J =E= 0 ;
eq1(t+2)..   Y(t+2)   =E=   C(t+2) + I(t+2) + G(t+2) + X(t+2) ;

eq2(t+2)..   Yd(t+2)  =E=    (1 - tax) * Y(t+2) ;

eq3(t+2)..   C(t+2)   =E=    a + b * Yd(t+2) ;

eq4(t+2)..   I(t+2)   =E=    ee - d * R(t+2) ;

eq5(t+2)..   M(t+2) / P(t+2)  =E=   k * Y(t+2) - h * R(t+2) ;

eq6(t+2)..   piex(t+2)=E=    alpha * pi(t+1) + beta * pi(t) ;

eq7(t+2)..   pi(t+2)  =E=    piex(t+2) + f*(Y(t+1)-Yn(t+2))/Yn(t+2) ;

eq8(t+2)..   P(t+2)   =E=    P(t+1) * (1 + pi(t+2)) ;

eq9(t+2)..   E(t+2) * P(t+2) / Pw(t+2)  =E=  q  + v * R(t+2) ;

eq10(t+2)..  X(t+2)   =E=   gg - mm*Y(t+2) - n*(E(t+2)*P(t+2)/Pw(t+2));

eq11(t+2)..  Gd(t+2)  =E=    G(t+2) - tax * Y(t+2) ;
eq12(t+2)..  U(t+2)   =E=    Un(t+2) - mu*(Y(t+2)-Yn(t+2))/Yn(t+2) ;

***********************************************************************
* In what follows, we assign initial variables' values and lower bounds
* WARNING: The order of declaration of assignments is very important
*     Successive assignments to a same variable undo the previous ones
***********************************************************************

*  Guess of initial values for the solution algorithm.
*  Without them, the problem may be declared "infeasible"
*  That is, the algorithm will converge to a solution from some initial
*  positions but not from others
*  This is common in nonlinear problems

 R.L(T+2) = 0.09 ; Y.L(T+2) = 6500 ; E.L(T+2) = 1.2; C.L(T+2) = 4500 ;
 I.L(T+2) = 900 ;  X.L(T+2) = -100 ; Gd.L(T+2) = 75 ; U.L(T+2) = 0.07 ;
 Yd.L(T+2)= 4875 ; pi.L(T+2) = 0.1 ; piex.L(T+2)=0.2 ; P.L(T+2) = 1.1 ;
```

```
* lower bound for p, to avoid  division by zero
 P.LO(T+2) = 0.0001 ;

* fixing initial steady-state values for lagged endogenous variables
 P.FX(T1) = 1 ;  pi.FX(T0) = 0 ;  pi.FX(T1) = 0 ; Y.FX(T1) = 6000 ;

MODEL NONLDYN  /eq1, eq2, eq3, eq4, eq5, eq6,
                eq7, eq8, eq9, eq10, eq11, eq12, JD / ;

SOLVE NONLDYN MINIMIZING J USING NLP;

* Reporting solution values
PARAMETER REPORTS SOLUTION VALUES IN LEVELS;
  REPORTS(T,"GDP")          = Y.L(T);
  REPORTS(T,"Inflation")    = pi.L(T);
  REPORTS(T,"Int.Rate")     = R.L(T);
  REPORTS(T,"Exch.Rate")    = E.L(T);
  REPORTS(T,"Gov.Def")      = Gd.L(T);
  REPORTS(T,"Unemploy")     = U.L(T);

* Showing final results
DISPLAY REPORTEX;
DISPLAY REPORTS;
```

B Ordered sets in GAMS

As we discussed Chapter , the definition of lagged indices for variables in GAMS may be somewhat problematic if it is not done with care. For example, if the variables w and z were defined over a set `t[i.e., w(t) and z(t)]` such as

$t = \{0,1,2,3\}$

then an expression like

```
eq(t).. w(t) =E= z(t 1)
```

result in the following equations being generated by GAMS:

```
eq(0).. w(0) = E = 0;
eq(ref{eq1}).. w(ref{eq1}) = E = z(0);
eq(ref{eq2}).. w(ref{eq2}) = E = z(ref{eq1});
eq(ref{eq3}).. w(ref{eq3}) =E= z(ref{eq2});
```

Thus, it causes GAMS to assign the value zero to the first element of `w(t)`, since the element `z(--1)` of the variable z is not defined. We do not want this to happen, since it would be a source of confusion at the time of assigning initial values for lagged variables and also for the interpretation of solution values corresponding to the initial periods of the solution horizon.

To be sure about the results of the dynamic specifications in GAMS, every time one writes a program involving dynamic variables it is advisable to set `OPTION LIMROW` equal to the maximum number of periods involved in the solution of the model. This tells GAMS to print a detailed equation-by-equation solution report that allows one to check period-by-period the evolution of the time indices for each variable within each equation. It is particularly important to check the specification of the equation for the first few and the last few time periods. For example, here is how the corresponding GAMS output looks for equation (6) in Chapter :

```
eq6(t+2).. piex(t+2) =E= alpha * pi(t+1) + beta * pi(t) ;
```

when Hall and Taylor's model is solved for a time horizon of 7 periods, that is, for a t set equal to $\{0,1,\ldots,5,6\}$.

```
---- EQ6 =E= expected inflation
EQ6(ref{eq2}).. -- 0.2*PI(0) -- 0.4*PI(ref{eq1}) + PIEX(ref{eq2}) =E= 0 ;
EQ6(ref{eq3}).. -- 0.2*PI(ref{eq1}) -- 0.4*PI(ref{eq2}) + PIEX(ref{eq3}) =E= 0 ;
EQ6(ref{eq4}).. -- 0.2*PI(ref{eq2}) -- 0.4*PI(ref{eq3}) + PIEX(ref{eq4}) =E= 0 ;
EQ6(ref{eq5}).. -- 0.2*PI(ref{eq3}) -- 0.4*PI(ref{eq4}) + PIEX(ref{eq5}) =E= 0 ;
EQ6(ref{eq6}).. -- 0.2*PI(ref{eq4}) -- 0.4*PI(ref{eq5}) + PIEX(ref{eq6}) =E= 0 ;
```

Note that `eq6(t+2)` goes from periods 2 to 6, while pi(t) goes from 0 to 4, `pi(t+1)` from 1 to 5 and `piex(t+2)` from 2 to 6. This means that the effective solution horizon for the model is equal to five periods, two fewer than the number of elements of the set `t`.

For further details see the chapter on *Set as Sequences: Ordered Sets* in the GAMS User's Guide at www.gams.com.

C The Stacking Method in GAMS

As a compact way of expressing a multiequation model, GAMS allows us to write indexed equations. As seen in several of the chapters in this book, those indices may represent commodities, locations, time periods, and so on.

For example, the equations corresponding to a problem such as

$$\max J = \sum_{i=0}^{2} w_1 x_i + w_2 y_i \tag{C-1}$$

subject to the constraints

$$a_{11} x_i + a_{12} y_i = b_1 \tag{C-2}$$
$$a_{21} x_i + a_{22} y_i = b_2 \tag{C-3}$$

can be represented in GAMS as

```
eqj..       j =e= sum(i, w1 * x(i) + w2 * y(i));
eq1(i)..    a11 * x(i) + a12 * y(i)) =e= b1;
eq2(i)..    a21 * x(i) + a22 * y(i)) =e=  b2;
```

When the index set is $i = \{0, 1, 2\}$ the model is expanded and stacked in the following way:

```
j =e= w1*x(0) + w2*y(0) + w1*x(1) + w2*y(1) + w1*x(2) + w2*y(2)
eq1(0)..    a11 * x(0) + a12 * y(0)) =e= b1;
eq2(0)..    a21 * x(0) + a22 * y(0)) =e= b2;
eq1(1)..    a11 * x(1) + a12 * y(1)) =e= b1;
eq2(1)..    a21 * x(1) + a22 * y(1)) =e= b2;
eq1(2)..    a11 * x(2) + a12 * y(2)) =e= b1;
eq2(2)..    a21 * x(2) + a22 * y(2)) =e= b2;
```

Note that previously we had a model with an objective function and two indexed equations and two variables [x(i) and y(i)] and now we have a model with one objective function, six equations, and six variables [x(0), x(refeq1), x(refeq2), y(0), y(refeq1) and y(refeq2)]. Thus, before solving the model, GAMS transforms a model of n indexed equations into one of *n x card* equations plus the objective function, where *card* indicates the number of elements in the index set. If the index denotes time periods, this is equivalent to transforming a dynamic model with n indexed equations and t time periods into an equivalent static model of $n \times t$ equations plus the objective function.

When, as in Chapters 8 and 13, we are interested in solving a system of equations and not an optimization problem, we just set the objective function equal to any constant value (i.e., j =e= 0;). Thus, when executing the corresponding solver statement,

```
solve model maximizing j using nlp;
```

GAMS expands and stacks the system of equations and it solves it as a by-product of a pseudo-optimization.

D Nonlinear Optimization Solvers

Solving nonlinear optimization problems usually requires the use of numerical methods. In general, those methods consist of a *smart* trial-and-error algorithm that is a finite sequence of computational steps designed to look for convergence to a solution. There is a variety of algorithms to solve nonlinear problems. Some of them are global methods, in the sense that they perform a parallel exploration of many regions of the optimization space, for example, the genetic algorithm. Other are local, as they tend to focus on the exploration of a particular region of the optimization space. We introduce here two of the most popular local methods—the gradient and the Newton methods—used by the solvers in Excel, **GAMS**, and **MATLAB**, but before introducing them, we give with a simple example.

Suppose that we are trying to find the maximum of a nonlinear function

$$y = f(x) \tag{D-1}$$

such as the one represented in Figure 5.

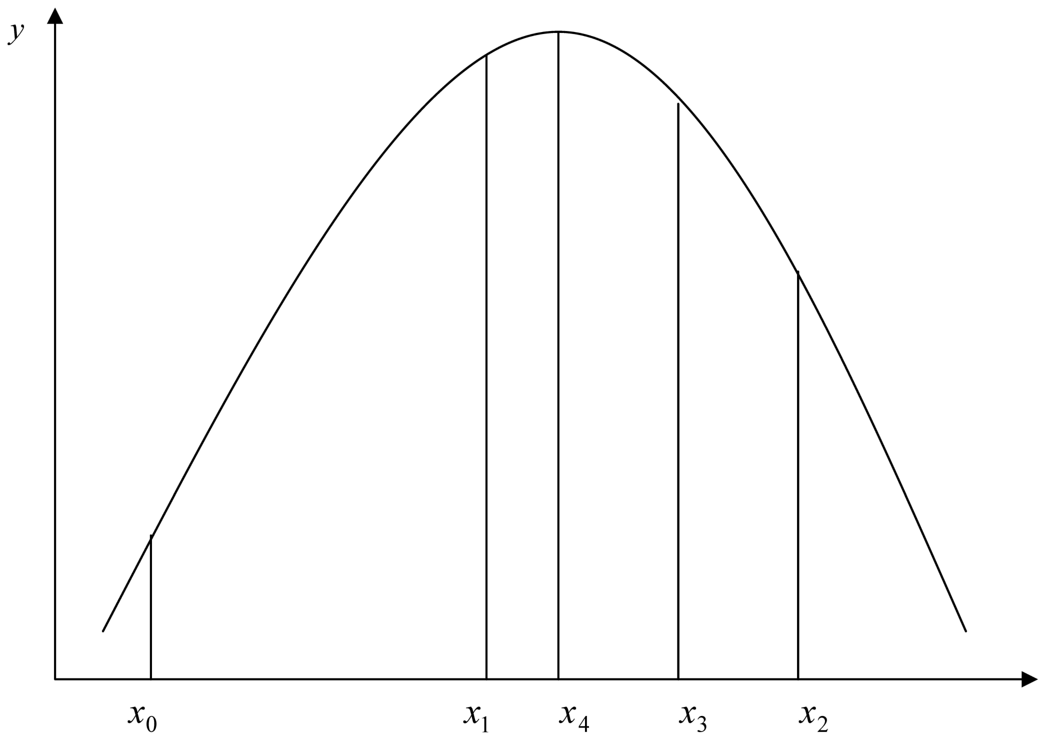
A simple and very rudimentary algorithm to find the solution might be as follows: We choose an arbitrary initial value for x, such as x_0 in Figure 5, and compute the corresponding $y_0 = f(x_0)$. We then increase that value by a constant magnitude h (we name this magnitude the *search step*) that we also choose in an arbitrary way. For the new value of x, that is x_1, we compute the corresponding value of

$$y_1 = f(x_1) = f(x_0 + h) \tag{D-2}$$

and compare this value to the one obtained in the previous step. We continue to do this as long as the differences between two successive values of y are positive (negative for a minimization problem). As soon as we compute a difference with a negative sign (in Figure F.1 this would correspond to x_2), we reverse the direction of the search. We begin to move in the opposite direction along x (i.e., subtracting h from x) and use a smaller value for h than the one we were using while we moved in the opposite direction. We continue like this until we again find a difference between two successive values of y that is negative, at which point we again reverse the direction of the search and further reduce the size of h, and so on. We stop when the difference between two successive values of y falls below a preestablished tolerance limit.

The gradient and the Newton methods are iterative like the one just presented.

Figure 5: A nonlinear function.



However, they exploit local information about the form of the function; that is, they use the function's derivatives. To illustrate this we change to a multivariate example. In this case we use the following equation to obtain each new value of the vector x:

$$x_{n+1} = x_n + h\Delta x \tag{D-3}$$

where h is the search step—now always a positive value—and Δx is the direction of change, which, as we will see, is determined by the function's derivatives.

The gradient method uses the first derivative or gradient, which gives us information about how the function changes in the neighborhood of a given point. Its basic framework is the well-known first-order Taylor approximation,

$$f(x_{n+1}) \cong f(x_n) + h\nabla f(x_n)\Delta x \tag{D-4}$$

where $\nabla f(x_n)$ is the gradient vector. Note that since h is supposed to be positive, the best direction of motion is

$$\Delta x = \nabla f(x_n) \tag{D-5}$$

for a maximization problem, since

$$f(x_{n+1}) \cong f(x_n) + h\nabla(f(x_n))^2 > f(x_n) \tag{D-6}$$

In addition, for a minimization problem

$$\Delta x = -\nabla f(x_0) \tag{D-7}$$

since

$$f(x_{n+1}) \cong f(x_n) - h\nabla(f(x_n))^2 < f(x_n) \tag{D-8}$$

The basic framework of the Newton method is the second-order Taylor approximation

$$f(x_{n+1}) \cong f(x_n) + h\nabla f(x_n)\Delta x + \frac{h}{2}\Delta x' H(x_n)\Delta x \tag{D-9}$$

where $H(x_0)$ is the second-order derivative or Hessian, which tells us how the slope of the function changes in a neighborhood of a given point.

Assuming the Taylor expansion of a function f is a good global approximation of that function, we approximate the optimum value of f by optimizing its

Taylor expansion. In our case, this is equivalent to saying that to determine the best direction of motion Δx we have to optimize the expression (f.9). Differentiating (f.9) with respect to Δx, setting the result equal to zero, and solving for Δx, we obtain

$$\Delta x = -\frac{\nabla f(x_n)}{H(x_n)} \tag{D-10}$$

which is the best direction of motion for the Newton method.

Sometimes iterative methods like the ones presented here do not converge to a solution after a finite number of iterations. This problem can be overcome by changing the maximum number of iterations, or the size of the search step, or the tolerance limit, or the initial value of the search. Most solvers allow you to change these parameters.

Note also, as is the general case for numerical methods dealing with nonlinear optimization problems, that if there is more than one local optimum we will find only one of them. Thus, we never know for sure if the optimum we reached was a local or a global one. A rough way of dealing with this difficulty is to solve the problem providing the algorithm with alternative initial values of the search.

In this appendix we presented three numerical methods of increasing complexity. Of course, the more complex ones make use of more information, thus reducing, in general, the number of steps needed to achieve convergence. However, those steps become more complex, since they require the computation of a gradient or a Hessian. Then there are trade-offs to be evaluated when choosing a solution method.

There are additional methods for solving nonlinear problems numerically—for example, the conjugate gradient method, the penalty function method, and the sequential quadratic programming—a number of which extend, combine, or mimic the ones introduced here. For a comprehensive presentation refer to Judd (1998) and Miranda and Fackler (2002). The Excel solver uses a conjugate gradient method or a Newton method. **GAMS** uses a variety of methods, depending on what you choose or have set up as the default nonlinear solver. The **MATLAB** solver used in Chapter 7 and invoked by the fmincon function uses a sequential quadratic programming method. For details on the specific methods used by **Excel**, **GAMS**, and **MATLAB** refer to their corresponding user's and solver's manuals.