Building and Using a Small Macroeconometric Model: Klein Model I as an Example

Renfro, Charles G

1 January 2009

1/16/2009

Building and Using a Small Macroeconometric Model: Klein Model I as an Example

A MODLER Workbook

Charles G. Renfro

MODLER Information Technologies Press
Philadelphia and Cambridge

LIMITED WARRANTY

Neither the manufacturer nor the distributors of the MODLER software shall have any liability or responsibility to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by this product, including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of this product. This product will be exchanged within twelve months from the date of purchase if it is found to be defective in manufacture, labeling, or packaging; but except for such replacement the license of this software is without warranty or liability.

The above is a limited warranty and the only warranty made by the manufacturer, publisher, or distributors of the MODLER software. Any and all warranties for merchantability or fitness for a particular purpose are hereby excluded.

Trademarks Acknowledged

MODLER, MODLER BLUE, MODLER MBA, DATAVIEW and their derivatives are trademarks of C.G. Renfro & Associates. LOTUS 1-2-3 and WordPro are trademarks of Lotus Development Corporation. Quattro Pro and WordPerfect are trademarks of Corel Systems. Excel, MS-DOS, and Word are trademarks of Microsoft Corporation. All other product names in this publication are trademarks or registered trademarks of their respective owners.

Servicemark

Software for people whose time is valuable is a service mark of the Alphametrics Corporation.

Contents

Introduction

The small econometric model known as Klein Model I, or sometimes the Klein Interwar Model, was created by Lawrence R. Klein in the mid to late 1940s and first published in 1950 in Cowles Commission monograph No. 11, *Economic Fluctuations in the United States 1921-1941 [19]*. This model has since been a favorite of economists, essentially because of its tractability, particularly when parameter estimation methods are considered. There are a number of textbooks that describe it or provide parameter estimates for it, including those by Christ [6], Desai [7], Goldberger [13], Intriligator et al [17], and Theil [31]. There have also been one or two attempts to use the model to study the policies pursued during the depression years of the 1930s. A characteristic of this model is that it is one of the simplest possible examples of a reasonably complete, closed economic model, straightforward enough to be described in a few pages, yet containing both important accounting identities — particularly the embryonic GNP identity — and familiar behavioral equations — specifically those for Consumption and Investment.

Employing Klein Model I as an example, this workbook describes the process of building and using an econometric model in the context of the MODLER software. As this document illustrates, quite a lot can be learned from this application about the general process of estimating, forming, and even solving a structural macroeconometric model, notwithstanding Model I's simplicity. But the Klein model also carries with it some flavor of an earlier time, which in certain respects adds interest. One aspect is the time period itself: the fact that the model refers to the now almost fabled period of the 1920s and 1930s adds interest from that circumstance alone. However, there is also interplay between the creation of the database and the model. Any macroeconometric model presupposes the existence of an economic database, but this model dates from the relatively early days of national income accounting. The data used with it had to be collected and organized as an integral part of the model building process, which the 1950 monograph describes in considerable detail. The extent of this discussion reflects that the creation of the Klein model, which occurred mainly during the years 1944-47, was contemporaneous with the initial organizational development of the US National Income and Product Accounts.

Today, in contrast, macroeconomic data are obtained comparatively easily: anyone wishing to create a model of the US referring to the past 50 to 60 years can simply download the data from the Internet, drawing upon a multiplicity of governmental statistical agency and other sites. However, the absence of a requirement to collect these data observation by observation does not mean that it is no longer necessary to understand their characteristics. Quite to the contrary: the greater separation of the database development from the tasks of model construction both implies a need for the model builder to study the work of the economic statistician, now a specialty in itself, and raises the question of the degree to which the currently more available data base provides the most appropriate data for the purpose of the model [14, 27]. In 1950, Klein characterized the then existing economic statistics as having "been prepared on the basis of intuitive concepts without regard to specific models of the system from which the data are derived; consequently, there is a serious lack of coordination between the econometrician and the national income statistician. The readily available economic time series are almost never in a form suitable for immediate use in econometric studies" [19, p. 123]. Although it is now usual to build econometric models essentially to fit the estimates produced by statistical agencies, reflecting ostensibly greater coordination between the econometrician and the statistician, these data still do not necessarily provide the best foundation for model building.

In any case, the process of building an econometric model, once the data are obtained, involves specifying and defining the equations of the model, which immediately requires a classification of the variables, and thus the data. A model's equations can of course generally be classified into two categories, at least in the case of simple models: identities and behavioral equations. Broadly speaking, the identities express the accounting relationships that define the framework of the model. For instance, Gross Domestic Product, in usage more modern than GNP, is the sum of Consumption, Investment, and Government Expenditures, at least in the absence of foreign trade. The current capital stock is equal to the existing capital stock at the end of last year, plus the investment in capital stock that has occurred this year, less the depreciation that occurs. Notice that such a description of these identities conveys immediately both that the model involved is a closed model and that it abstracts from certain realities. All models at best approximate the object they represent, raising instantly the question how good they are as approximations.

In turn, as we will consider, the behavioral equations are intended to express the behavior of both consumers of and investors in the goods and services produced, among other behaviors that could be represented. Because, in these cases, such equations directly or indirectly involve taxes and government purchases, they express this behavior as taking place in a policy affected world, notwithstanding that in the Klein model such items as interest rates, money supply, and financial markets are absent. Recalling that money and financial markets were center stage in the interwar period, a question that immediately occurs is: how can a model that excludes these tell us much about those times? It is to be expected that any historically early attempt to model the economy might leave out important descriptive elements, a fact fully recognized by Klein, so that this comment should not be interpreted as being critical of Model I. Instead, it should be interpreted as simply posing a question—that can be asked at the outset, but not yet answered—about the necessary degree of approximation before it is possible to begin to describe a model as a good, or even adequate representation of the real world.

There are a number of issues that such qualitative judgments involve, including the range of variables present in a model, the degree of their aggregation or disaggregation, and how the behavioral equations should be specified. This latter topic is intimately related to the estimation of the parameters of the model, inasmuch as behavioral equations by their nature exhibit numeric parameters, the values of which are unknown and must be estimated using statistical methods. The process of estimating a model's parameters generally begins with regression, a broadly applied statistical technique, but it does not end there. The study of the methods of parameter estimation has of course become its own disciplinary specialty over the past more than 70 years, known as econometrics, and now involves making a choice from a considerable variety of estimation techniques and related statistical tests [30].

Part of the explanation for this continuing concentration on parameter estimation, even until the present day, is the historical computational environment. Klein Model I effectively predates the electronic computer. Both ENIAC, the first (programmable) electronic computer and EDVAC, the first stored-program computer, were created in the 1940s, but the earliest use of any such computer by economists (the EDSAC, a sibling of the EDVAC) did not occur until the early 1950s [28]. Furthermore, even until the mid-1980s, well after computers became commonplace, the process of building an econometric model of any size involved considerable time and hands-on computational effort, apart from conceptual design issues. Descriptions of model building projects undertaken during the 1960s and well into the 1970s usually characterized this process as necessarily involving 10 to 15 economists and research assistants for time periods of as much as a year or more, in order to construct a model of 100 or 200 equations [22]. In the 1960s and 1970s, one of the reasons was the lack of easily accessible data, and in addition the lack of adequate computing facilities [28]. At the birth of the microcomputer in the late 1970s, users of even mainframe computers ordinarily had access to less than 640K of RAM (Random Access Memory). In contrast, today, "ultraportable" notebook computers commonly offer 100 times this amount; some provide more. And throughout those years, more often than not, it was necessary to keypunch the data from printed documents; it was only in the 1990s that economic data became widely available in machine-readable form and, in particular, easily downloaded from the Internet.

Only since the 1980s has it been possible for someone working alone, with a computer, and a data base, to construct and use an econometric model of any size within a time period of much less than a year [26]. In contrast, in the late 1940s, the process of computing regression parameters for a single equation took essentially an entire day [9]. In the middle 1960s, it could take less than an hour, but might involve three separate computer jobs, respectively to update the database, make any transformations, and then run the regressions. By 1968, it might take as little as a minute or two to perform these tasks, except that the fact that mainframes were then shared facilities meant that as much as 1-8 hours or more might pass for the results to be distributed by the computer operator to the individual user. During these years it was also generally necessary for a user to be a relatively experienced computer programmer, due to a general lack of readily available software. Consequently, as an ordinary experience, it was not until 1985, or even later, with the widespread diffusion of the microcomputer, that the *typical* user of a computer could expect to run a regression and obtain the results in less than a minute. Your experience will be even better: if you perform the examples as you read this document, you should find that nothing you do will take more than few seconds to perform, possibly only nanoseconds. This will be true whether your computer uses a (now "ancient") Pentium, the latest Intel Core Duo, or one of the quad core CPUs just on the verge of becoming available.

The MODLER software described in this document has played an active part in this productivity gain. It has been used interactively on networked computers since the first months of 1970. Since the early 1980s, it has been used worldwide, dating from the time it was first "ported" to the original IBM Personal Computer, or PC, as this type of microcomputer has come to be known [1]. The initial development of this software occurred in 1968 on an IBM 7040 mainframe computer [26]. The first interactive version was developed in 1969 and, at the beginning of 1970, was mounted on a Digital Equipment Corporation PDP-10 time-sharing mainframe computer at the Brookings Institution in Washington, DC; unusually for the time, terminals hardwire-linked to this machine were installed in offices and spaces throughout the Brookings and adjacent buildings. The specific impetus to develop this version of MODLER was to provide software to estimate the Brookings Quarterly Econometric Model of the United States on that new machine. However, once installed, it began to be used generally by Brookings economists and others. In 1970, MODLER thus became one of the first production examples of an interactive, multi-user computer program. It is the first interactive statistical or econometric software package, being also one of the first computer programs to offer free-format interpretive commands [29].

During the early 1970s, MODLER evolved into an integrated and comprehensive econometric modeling "language," designed to support the development and use of econometric models generally [29]. Its components became the nucleus of one of the earliest online economic database information systems [2, 3, 25]. Subsequently, once converted for use with the IBM PC ten years later, MODLER was the first econometric software package to be offered to users of this type of machine that was capable of estimating, building, and solving large-scale macroeconometric models. As a consequence, it was immediately adopted almost universally by firms in the econometric consulting and forecasting business, as well as more broadly. During 1982-84, it was used for the first estimation and solution of an econometric model on the microcomputer, which included not only Klein Model I, but also small to medium size models of the British and World economies. Beginning in September 1984, it was the basis for the first general use of large-scale econometric models in a microcomputer context, in the form of the Wharton PC-Mark7 Quarterly Econometric Model of the United States, a 250+ equation model, becoming a component of the first microcomputer-based economic forecasting service. In early 1985, it was used to estimate and solve the Wharton Mark VII 600+ equation model, which at the time many doubted could be estimated and solved on a microcomputer. Since then, versions of the software have been used to solve models of up to approximately 3500 equations in size, although the standard version is limited to 1000 equations.

In addition to Wharton Econometric Forecasting Associates, MODLER was adopted in the 1980s by such well-known US economic consulting firms as Chase Econometric Associates, Data Resources Inc, Lawrence Meyer and Associates, and Townsend Greenspan & Company, as well as by clients of these firms and many others worldwide. By the late 1980s, MODLER was in use in such European countries as Belgium, Denmark, France, Germany, Greece, Italy, the Netherlands, Norway, Spain, Sweden, and the United Kingdom, as well as in Africa, Australia, Canada, Central and South America, Mexico, the Middle East, Japan and Singapore. Among the types of users, it is now used in brokerage firms, central banks, corporations, and universities, as well as by local, national, and international organizations and governmental agencies.

However, this description does not entirely convey the essential characteristic of the microcomputer computer revolution of the past 30 years, which is the degree of replicated use it supports. Prior to 1980, mainframe computers were located in many places around the world and some of these could be accessed from a dial-up telephone link from virtually anywhere, providing in principle worldwide use. However, the significance of the personal computer is that with the advent of this technology, it became commonplace for the first time to transfer copies of software and models from one computer to another, thus allowing the creation of a true worldwide community of users. As early as 1970, or before, it was possible to move a model from one machine to another, but not easily; often this conversion involved considerable additional programming and was a rare event. The truth is, other than Klein Model I, very few econometric models were ever mounted on more than a single computer prior to October 1984 [18].

This modern ability to share easily both large and small models across many machines constitutes a major breakthrough. Yet it is only now that this change is being fully realized. For many years, the process of estimating and constructing models was sufficiently arduous that, in practice, there was much estimation and building, but very little general use. For one thing, macroeconometric models have often been considered to be simply forecasting devices, rather than as being potentially created for the more general purpose of simulating, evaluating, and characterizing the behavior of economies more broadly. Part of the reason for this lack of broad use has been the complexity of the large models. However, an additional very important reason has been a general absence of sufficient documentation and explanation to permit economists in general to be able to comprehend and work with these models.

A further reason, perhaps, is the relatively uninteresting operational characteristics of many of these models historically, as compared to those ostensibly offered by the inoperative theoretical models that inhabit the literature. In order to help to bridge this conceptual gap, a document parallel to this one has been created that describes the characteristics and use of an interesting series of operative theoretical structuralist economic models that are described in a new book by Wynne Godley and Marc Lavoie, *Monetary Economics* [10, 11]. These particular models are *not* econometric models and are *not* estimated; the values of their behavioral parameters are simply assumed. However, once an operative theoretical model of this type has been created, the process of using it is actually quite similar to using a macroeconometric model. *Monetary Economics* provides the logic and details of the construction of these models. Consequently, once you have worked through the present document, you might find it interesting as a next step to look at the companion workbook, *Building and Using Theoretically Defined Operative Economic Models*: *The Godley-Lavoie Models as Examples*, which can be downloaded from the Learning Tools page of the www.modler.com website. As you will see from the Preface of *Monetary Economics*, MODLER was from the first used by its authors to create these models.

Incidentally, all these MODLER workbooks are designed to be printed using a color printer. Certain black and white printers are not capable of producing readable tables and figures. However, if you have a copy of Word 2000 or later, Word documents are also available that can be used to print in black and white.

Technical Note

This workbook is designed to be used with all recent versions of the MODLER software, beginning with version 10.7, build 1. The software is sometimes distributed together with a CD-ROM that contains a collection of MODLER compatible files. These are contained in a self-extracting file called KMODEL.EXE. Alternatively, this self-extracting file can instead be downloaded directly from the *Learning Tools* section of the www.modler.com Internet website.

The files included in the file KMODEL.EXE are:

```
KLEINBNK.BNK  03-07-03  12046
KLEIN1.MOD    03-07-03  38138

KIDENT.MAC    02-24-03    136
ESTMOD.MAC    02-24-03    202
TSKLEIN.MAC   02-24-03    330

ESTMOD.TXT    03-07-03   5891
```

The date of creation of each of these files is here given for reference in US date format (mm/dd/yy); the number to the right of the date is the size of the file in bytes.

The first of these files, KLEINBNK.BNK, is a MODLER compatible data bank that contains the data necessary to create and/or use Klein Model I with MODLER. These data are annual in observation frequency and span the period from 1920 to 1941. The values are replicated from Lawrence Klein's Cowles Commission monograph No. 11, *Economic Fluctuations in the United States 1921-1941* [19], page 135. The second file, KLEIN1.MOD, is the complete model, contained in a MODLER model file, colloquially known as a MOD file from its extent. These two files together are sufficient to create a working version of Klein Model I; in fact, if you wish to skip the estimation of the model and to consider it beginning at the model-use stage, then go directly to Chapter II of this workbook.

However, if you wish to proceed step-by-step, start with Chapter I: the remaining four files will permit you to create the model from scratch, although all that you actually need is KLEINBNK.BNK and this workbook. The three "macro" files (distinguished by the extent .MAC) simply permit you to avoid typing the equations. The first of these, KIDENT.MAC, contains the identities for the model. The second, ESTMOD.MAC, contains the code necessary to estimate the behavioral equations of the model using Ordinary Least Squares as the estimation method. The third, TSKLEIN.MAC, contains the code necessary to estimate these equations using Two Stage Least Squares. Finally, the fourth file, ESTMOD.TXT, is simply a text file that briefly describes the process of creating Klein Model I using these other files; it is included for self-documentary completeness and is redundant when you use the present workbook.

Generally this workbook presumes that you already are familiar with the MODLER software, to at least a degree. If you are not, the *MODLER Getting Started Guide*, the main *MODLER User Guide*, and the online MODLER helpfile, accessible while using the program – all of which can be downloaded from the www.modler.com website – are together intended to be used as a means of becoming more generally familiar with the MODLER software prior to reading the present workbook. This workbook is ***not*** a substitute for these other documents; it complements them. Furthermore, you are best advised to have read at least the *MODLER Getting Started Guide*, inasmuch as it describes not only the website-based automatic updating of the program, as improvements are incorporated, but also the MODLER facilities that permit the ancillary use of word-processing, spreadsheet, Internet browser, and other programs likely to be found on your machine.

Chapter 1 Building the Model: First Steps

Klein Model I consists of three identities, expressing the accounting framework of the model, and three estimated equations, generally behavioral in type. This chapter introduces the model and describes its specific structural characteristics, but after a brief introductory discussion the chapter's predominant focus is actually upon the step-by-step process of creating a working macroeconomic model using the MODLER software; it is not so much concerned with the conceptual whys and wherefores of the model's particular structure as it is about this process. The Klein model is described in the Cowles Commission monograph No. 11, *Economic Fluctuations in the United States 1921-1941* [19], as mentioned earlier. More modern treatments and evaluations include that in Berndt [4, chapter 10], Intriligator, Bodkin and Hsiao [17] and Desai [7]. The model is also described in Bodkin, Klein, and Marwah [5] in the context of an evaluation of the historical development of macroeconometric models generally.

Using MODLER's syntax for mathematical expressions, Klein's original statement of the estimated model [19, p. 67ff] can be presented in terms of the following 6 equations:

CE = 0.80*(W1+W2) + 0.02*PI + 0.23*PI(-1) + 16.78
I = 0.23*PI + 0.55*PI(-1) - 0.15*K(-1) + 17.79
W1 = 0.42*(Y+TAX-W2) + 0.16*(Y(-1)+TAX(-1)-W2(-1)) + 0.13*(T-1931) + 1.60
Y = C + I + G - TAX
PI = Y – W1 – W2
K = I + K(-1)

where the endogenous variables are:

Y = income and net output (private net national product at market prices, net of business taxes)
CE = consumption expenditures

I = net investment
W1 = private wages
PI = profits
K = capital stock, at year end

and the exogenous variables are:

G = government nonwage expenditure
W2 = public wages
TAX = business taxes
T = time in years (that is, 1920, 1921, …)

Considering this model first simply as a set of simultaneous equations, the endogenous variables are mathematically the "dependent" variables, particular values of which can be generated by a "solution" of the model. The exogenous variables are the "independent" variables, the values of which must be determined in advance, before any model solution can be made. As to these variables' characteristics as economic measurements, except K and T, all are flows per year, measured in billions of dollars of 1934 purchasing power. K, a stock, is also measured in billions of 1934 dollars. Time, which appears only in the term T-1931, is therefore measured in deviations from 1931. Notice also that, because the equations are provided in vector notation, lagged values are expressed simply as negative offsets, -k, where k is the order of the lag, rather than by using the notation t-k.

Although the equations are written in a slightly different form than in a printed economics text, including the absence of Greek letters, subscripts, superscripts, and other such familiar embellishments, MODLER's syntax is nevertheless sufficiently close to standard mathematical notation that you should be able to decipher it immediately. However, there are a few important interpretative differences involved, the most significant of which is the possibly less-than-obvious fact that the equals sign here is an *operator*, not just a symbol expressing equivalence. This operator quality of = is why the first and second identities (equations four and five) are stated as they are, rather than as did Klein (p. 68), who, in a non-operative written context, expressed them as:

Y + TAX = C + I + G

and

Y = PI + W1 + W2.

A fundamental design characteristic of MODLER since 1968-69 is the ability for the user to write equations in a manner that is very close to standard written notation, but with the critical difference that what is written has an operational quality: a mathematical statement is not just a declarative statement, it is generally a *command* to the program to perform some operation. However, in order for this type of operation to be performed by a program, it must first be able to interpret such expressions syntactically, applying mathematical logic. Second, it must be able to retrieve each of the original variable values from some data source and, once the computations are made, store these values as well until they can be displayed or used otherwise.

The idea that written expressions can stand for an evaluated set of numeric values is so familiar and well integrated into modern scientific thought that this particular aspect probably will not strike you as in any way new or novel, but what is actually relatively new — a result of the invention and development of the electronic computer — is the fact that these expressions can be operative, not just symbolic. The syntax in operative form takes a little getting used to: in particular, from the start you need to be aware that, implicitly or explicitly, the computer must be told *how* to operate with the information that you provide it, which may either require a little more from you, or else the prior establishment of a default convention. For example, when writing an equation, it is usually easiest, in the sense of being most parsimonious, to use the explicit (normalized) form, as above. However, as software, MODLER is sufficiently sophisticated that, *in the context of a model*, were you initially to write the first identity above in the form:

$$Y + TAX = C + I + G$$

the program could still interpret this expression properly *if* you then were to issue a second command that, in effect, tells MODLER how to deal with the left-hand-side sum Y + TAX; that is, a command that tells it which one of these two variables is "explained" by the equation. Each equation in a putatively solvable set of equations must finally contain a single unknown and equation-by-equation you must either explicitly or implicitly provide this information.

Observe that the model above is shown with its three behavioral equations listed first, followed by the three identities. As indicated earlier, one of the characteristics of the behavioral equations is the numerical parameter values that appear in them; in the context of an econometric model, these values need to be estimated in order to create an operative, solvable model and the process of doing this will be considered later in this chapter. In contrast, identities do not include any parametric values, at least those that are a priori *unknown*; often they will not contain any such values at all, but the stress is upon the idea of behavioral equation parameters being *unknown in advance*. This presence or not of parametric values here allows these two types of equations to be immediately distinguished, but more generally the absence of parametric values is not a necessary characteristic of identities; in some cases, they may contain explicit scalar numbers. The difference is that, in general, whenever they appear they will be given *a priori and therefore known*.

These subtle distinctions are less obvious presently, inasmuch as the behavioral equations are shown here with all parameter values already estimated – as originally estimated by Klein using the method of Limited Information Maximum Likelihood, which he calculated essentially by hand, using a desktop electromechanical calculating machine. In fact, as a procedural matter, the degree to which a model's parameters will need to be estimated locally using MODLER will depend upon the particular model, and how and from where it has been obtained. The software provides the *means* to estimate parameters as necessary, but does not *require* any parameters to be estimated locally: the above equations can be included in a MODLER-resident model exactly as they are, and that model can then be solved. That is, the software provides facilities, but the degree to which these are employed depends upon the situation. Insofar as the software is concerned, the constant values that appear in the equations of a given model can be determined in any of multiple ways: by assumption, by estimation, or any other method. Indeed, the interpretation of a given equation as an identity, a behavioral equation, or some other type is essentially in the mind of the model builder, not a feature of the software. As will be discussed as we proceed, this ambiguity provides flexibility in the use of the software and is a desirable feature, although initially it potentially can cause some confusion.

Going beyond generalities and focusing attention on the specific model illustrated above, several observations are pertinent. First, compared to more modern macroeconometric models [8, 15], Model I is unusual in expressing consumption expenditures as a function of wages, profits, and lagged profits. More modern specifications almost always treat consumption expenditures as a function of disposable personal income, among other variables, in the process subsuming (non-corporate) profits, with the effect thereby of preventing the possible portrayal of differential behavioral effects resulting from relative changes in wages and these profits. Personal Income by definition is a composite of wages, the net income of unincorporated businesses, dividend payments by corporations, rental income (which includes the imputed income of owner-occupiers of real property), government transfer payments, and a number of other, generally much smaller, component items. At the same time, modern models, particularly the larger ones, also tend to disaggregate consumption into its constituent types, including at minimum consumption expenditures on durables, nondurables, and services, in turn then expressing total consumption expenditures as the sum of these components. Consequently, if the equations are progressively considered, the behavioral relationships nowadays tend to be stated at a less aggregate level than in Klein Model I, but with individual variables *in the equations* sometimes treated more aggregatively.

A similar change has taken place in the treatment of investment: modern macroeconometric models not only also disaggregate total investment into particular subcomponents, such as business fixed investment, residential construction, and inventory investment, but their equation specifications also tend to remove the direct effects of changes in profits and capital stock per se from these equations.

Finally, although wage equations still appear in today's models, wages themselves seldom if ever appear as specific explanatory variables in aggregate demand equations. Instead, they tend to appear in employment (or manhours) equations and in personal income and related identities. Furthermore, to the extent that wages appear they usually appear at the disaggregated, more industry specific level.

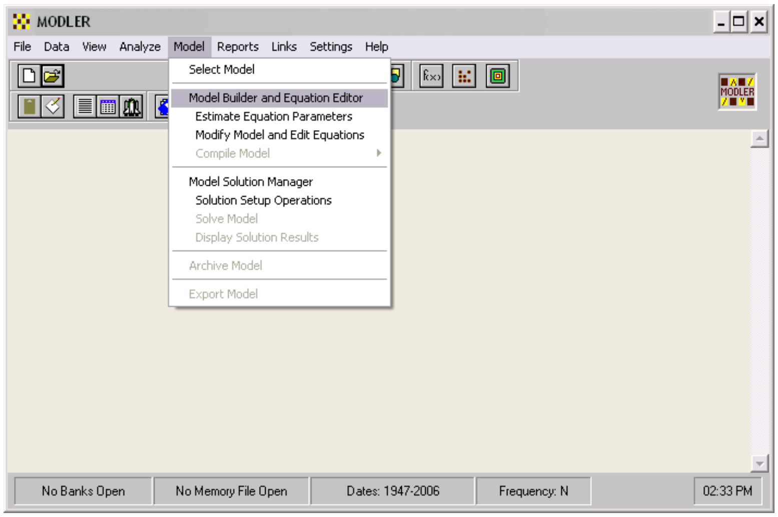
The differences just described reflect both circumstance and now more than 50 years of model building experience. The specific modern ability to disaggregate models is in large part the result of the fact that the variables that appear in them today are based upon National Income and Product Accounts, Employment, Prices, and other organized sets of measurements that are published in relatively detailed form by governmental statistical agencies, trade organizations, and a few other widely used sources. As mentioned in the Introduction, the day is gone that an individual economist might undertake the creation of a set of accounts for the purpose of building a macroeconometric model. The effect is thus to focus the attention of model builders on the explanation of standardized variables. And because these standardized variables may have familiar names (such as savings, consumption expenditures, personal income, and the like) but specific constructional meanings, there is if anything just as much need for the model builder to understand the nuances associated with these particular measurements. For instance, in this context Savings is a particular accounting residual and not necessarily the behavioral variable it is popularly taken to be, nor necessarily even a measure having any particular relevance to what an individual economic agent might consider as affecting his or her well being — or even expressing current period net addition to wealth. For instance, changes in the value of home equity and mutual fund portfolios (including those designed to ultimately provide pensions) are not included in Savings, as defined in National Income and Product Accounts.

Other comments could be made, but the spirit of these remarks is that the best way to approach econometric model building and the MODLER software is with a relatively open mind. Insofar as the model equations are concerned, you might want to evaluate how these express the relationships among variables. Insofar as the software is concerned, you might want to consider what options it provides: there are certain restrictions on what you can do, but the design intention is to provide as flexible a tool as possible. As discussed in the Introduction, it is only relatively recently that software like MODLER has become widely available, so that its role should be seen as purposely facilitating what its users individually and collectively wish to do. Of course, to the degree that you can imagine ways in which it could be improved, feedback is always very welcome and appreciated.

Building a Model: Basic Facilities

As indicated earlier, you are presumed to have already become generally familiar with the MODLER software and to have previously established its various environmental settings, as described in the *MODLER Getting Started Guide* and the general *MODLER User Guide*. You are therefore assumed to be reasonably comfortable with the basic MODLER facilities and not to need specific help on the use of data banks, or setting the observation frequency and the date range. It is best if you have also previously executed at least a few regressions, printed some tables, and created a few graphs, as described in the *MODLER User Guide*. However, if you have not, you may still find that this workbook provides sufficient examples that you will feel comfortable immediately whatever your previous experience.

In the context of MODLER, the construction of a multi-equation model will usually begin at the Model Building and Editing Screen. The opening MODLER screen, called the Central Control Screen, is shown in Figure 1.



No Banks Open | No Memory File Open | Dates: 1947-2006 | Frequency: N | 02:33 PM

Figure 1. Central Control Screen

This screen has four principal elements. Considered from the top down, the first is a menu bar the individual elements of which, if "clicked" with a mouse, display drop down menus, an example of which is displayed. Next, the screen displays two rows of clickable icons. These icons are each linked to particular menu elements but are not intended to replicate the full set, only the most commonly used. In addition, the light yellow or pastel rectangle that takes up most of the screen, and is called a “blotter” can be used both as an entry point for commands and to display previously executed commands. This blotter is sufficiently like that of a word processing program, as well as those of other programs, that the idea of entering text into this space should not be too startling. Furthermore, as you might expect from previous use of a word processor, if you click your mouse anywhere on this blotter, the immediate effect is likely to be that a cursor starts blinking slowly, initially in the upper left hand corner. Just as with a word processor, this blink signals the location of the text entry point — in this case the potential entry point of a command. Finally, at the bottom of the screen is a status bar. The elements of this status bar are live, in the sense that if you click on them MODLER will respond, permitting you to discover additional relevant information.

In order to invoke the Model Building and Editing Screen, select the menu item **Model** in the middle of the Central Control Screen and then click on the line item **Model Builder and Equation Editor**, which is highlighted in Figure 1. In response, the Model building and Editing Screen should then immediately appear. It should look like the screen shown in Figure 2, but initally without any drop down menu exposed. At first, this second Screen will be minimally descriptive, containing as menu items only **File**, **Links**, and **Help**. This lack of detail indicates that, as yet, there is no model attached. In fact, at this point the presumption is that the model has yet to be created; for this reason, we have ignored **Select Model**, the topmost menu element of the drop down menu displayed in Figure 1, which will be considered later.

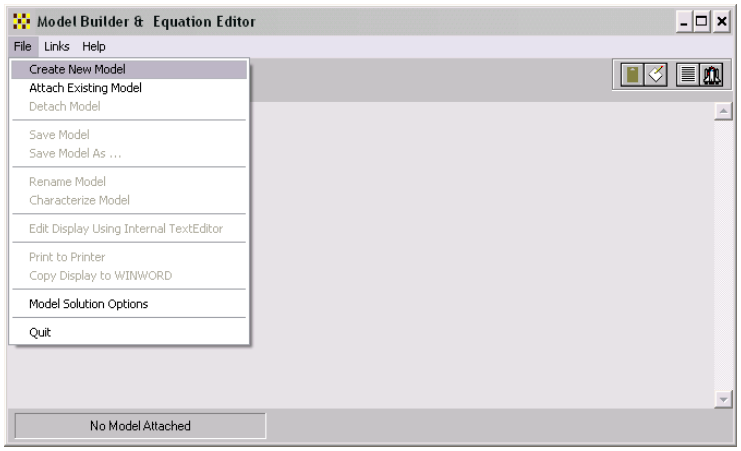
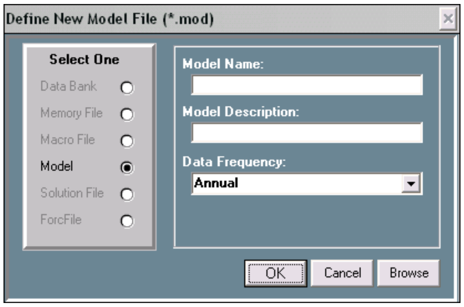


Figure 2. Model Building and Editing Screen

If at any time, you have questions related to the creation or formation of a model, select the menu item **Help**, which will provide you with context specific assistance. If you explore these help facilities, you will discover not only that they are relatively extensive, but in addition that you can print portions in order to supplement the possibly more cursory descriptions in this workbook. Furthermore, as you move from screen to screen, at the top level, if you press the F1 key on your keyboard as you arrive at each in turn, an outline description of that screen should immediately appear.

In order to begin to create a model, now select the menu item **File** on the Model Builder and Equation Editor screen, shown in Figure 2. Then click on the line item **Create New Model**, which is highlighted. In response, a subsidiary window, shown in Figure 3, should then appear. This form obviously allows you to specify the name of a model, its description, and the observation frequency of the data to be used with it. The model name must be between 1 and 8 characters in length, *must begin with an alphabetic character*, and should contain only alphabetic characters and numbers. For present purposes, the name *Klein1* might be appropriate. So enter it. The description is optional, but you might wish to put there the words "Klein Model I." The observation frequency of the model should be specified as Annual, as illustrated. The data for this model, contained in the data bank Kleinbnk, are annual; they span the dates from 1920 to 1941.



Define New Model File (*.mod)

Select One

○ Data Bank
○ Memory File
○ Macro File
● Model
○ Solution File
○ ForcFile

Model Name:

Model Description:

Data Frequency: Annual

Figure 3. Define New Model form

Once you have filled in at least the model name, click the **OK** button on this form and you should next see the screen displayed in Figure 4; this is the same screen as previously shown in Figure 2, but it is now displayed without the dropdown file menu. Observe, in particular, that the name of the model just created now appears on this screen's status bar. If you should click on this element of the status bar, you will then see a small screen that will describe the characteristics of this model. At the moment, the information it provides is only minimally informative, inasmuch as all we have done so far is to create the file to hold the equations of the model; we have not yet provided any equations. Essentially, we have an empty file, a fact that you need to be aware of, inasmuch as empty files can be confusing to a computer program should you at this point begin to ask the program to display or solve the model. Such files should not be left hanging around your hard drive, lest in the end they confuse you too. It is best to go ahead and start filling this empty box, without further ado.

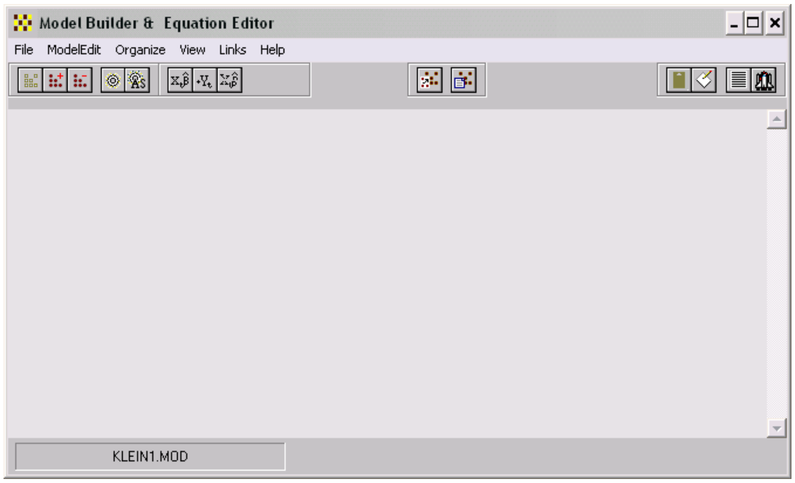


Figure 4. Model Builder and Editing Screen: New Model Attached

At this point we have several options, for the next step is not rigidly defined. In fact, in terms of the model building process itself, you presently have a number of possible choices, and which of these you should make is in part simply a matter of what you wish to do. For instance, you can begin by entering the model's identities. Alternatively, you can estimate the parameters of the behavioral equations then include the identities. Or you can include the various equations in some mixed order. As a piece of software, MODLER does not care which of these options you choose. Furthermore, so far as the ultimate model is concerned, provided that the equations are properly entered, each of these choices can lead to the creation of exactly the same model – in terms of its functional properties. The difference will be the initial ordering of its equations, which you may want to evaluate simply in terms of display considerations.

In order to make a good selection among these choices, it might be helpful initially to survey the Model Building and Editing Screen shown in Figure 4. Consider first the **File** menu element. If you click on it, you will see that it allows you to create new models, attach existing models, or detach the model, and other similar actions; these choices were shown in Figure 2, although there some were grayed out that will not be now. This **File** menu also allows you to save the model or to copy it to another file

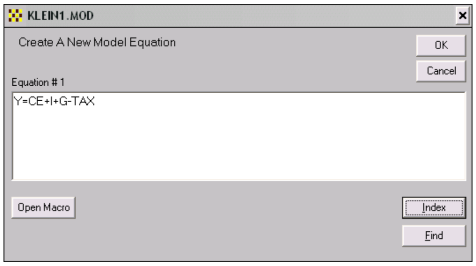
(**Save Model As…**), or else to rename it or discover its characteristics. Generally, these operations refer to the model broadly and, at the same time, more specifically to various file operations pertaining to the model file.

Next click on the menu item **ModelEdit**. At this stage, reflecting that the model so far hardly exists, many of the choices are still grayed out. However, observe in particular the three dropdown choices, **Add New Equation**, **Include Another Model's Equations**, and **Import Text Equations By Macro**. These represent three different ways to enter equations into the model. The first of these, which we will consider presently, provides the means to add equations one at a time. The second provides the means to add one or more equations contained in another MODLER model. Finally, the third allows you to import text equations from any other source, using a text file, generally in the form of a macro file; in this context, such a file can be presumed to contain model equations, which are in effect commands. Of course, these text equations must each obey the MODLER syntactic conventions for equations, and you may therefore need to edit any equations you obtain externally from other software contexts, among them AREMOS, EViews, MicroFit, PCGive, TROLL, TSP or other such packages. However, in most cases, so long as the package produces text representations of estimated equations not a lot of editing will be required, inasmuch as other packages commonly implement MODLER's syntactic conventions.

The one caveat that bears mention is that imported text models can contain redundant variables and equations. A common characteristic of other programs is that they may require the creation of extra variables as a prerequisite, in order to express regression and other commands using only variable mnemonic names. MODLER, in contrast, encourages the embedding of transformations in almost all commands, which as a consequence means that natively created models will normally contain the *minimum* number of variables and equations. In particular, MODLER models need not contain identities the sole purpose of which is to define specific transformations. After all, such extra variables do nothing but add to the bookkeeping requirement, as well as to make the models somewhat harder for you to interpret visually. Thus when importing text equations into MODLER, as a prior editing step — using either the inbuilt MODLER text editor or a notes editor such as Notepad or Wordpad — it is generally a good idea to minimize the number of variables by introducing into the imported equations all relevant embedded expressions. Among other benefits, this step will free you later from repeatedly having to update redundant data series that may be nothing more than the sum, log, ratio, or other such transformation of the essential model variables.

Entering New Equations: the Identities

Focusing attention now on the item **Add New Equation**, if you click on this menu term you will be presented with two choices. The first is **Key In Equation**. The second is **Estimate Equation**. Let us begin by adding an *identity* as the first model equation: if you click on **Key In Equation**, you will next see a form like that shown in Figure 5. In this figure, the identity has already been inserted, as you can see. Key in this identity yourself, then once you have checked to make sure that you keyed in the same equation as shown here, press the Enter key. By performing these steps, you will have specified the first model equation.

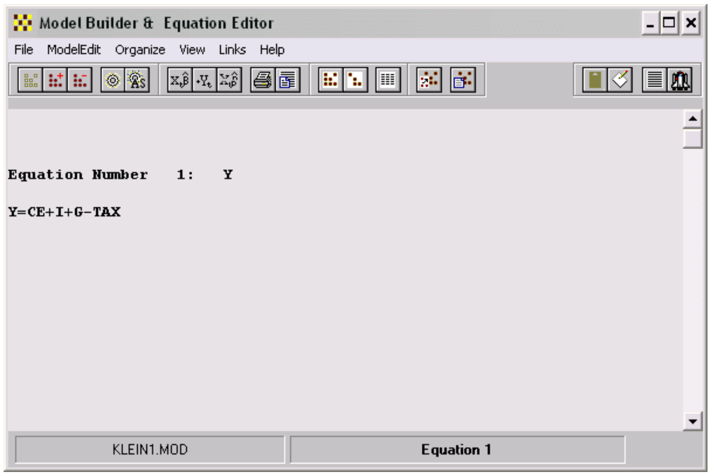


Equation # 1

Y=CE+I+G-TAX

Figure 5. Adding a New Equation

Actually, the fact that you have succeeded in creating a model equation will not be obvious at first, for at this point you will simply be returned to the Model Building and Editing Screen, almost as if nothing has happened. However, if you next select the **View** menu item, and then click on **Complete Model**, you will see the screen shown in Figure 6. This display confirms that the model now contains one equation.



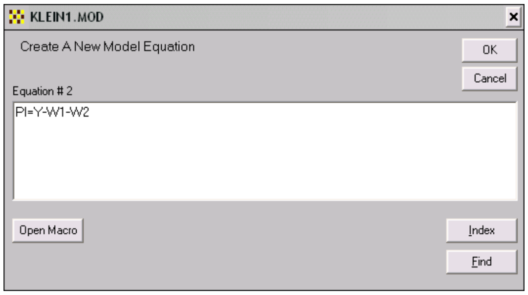
```
Equation Number   1:   Y

Y=CE+I+G-TAX
```

Figure 6. Displaying the first model equation, an identity

Observe two things in particular about this display. First, that the equation will not be displayed until you click successively on **View** and then **Complete Model**. Second, once further equations have been added, the display will *not* change until you *again* click successively on **View** and **Complete Model**, or on the relevant icon, which will be quicker. This display does not automatically refresh itself as you change the model. At this stage, you might wish that it did. However, once you get used to MODLER, you would then increasingly find any such automatic display a nuisance, since it would ultimately slow you down to have to look at everything repeatedly as you work. Once you have become proficient with the software, you will want to be able to work rapidly, rather than to be forced to look at everything as you go. MODLER is intended to be permissive, rather than dictatorial. It is also intended to be software for people whose time is valuable.

A third aspect is that if you now click also on the **File** menu item, you will see that no longer are there any grayed out items in this list. A particular option now newly available to you is to copy the display contents to your word processor. Another is to print the display to your printer. These particular display options reflect that an important aspect of MODLER is the facilities it provides in order to progressively document what you have done at virtually every stage of the process — if you so wish. But do not wish for the moon: you cannot expect the software to display everything automatically.

We have created the first equation. In order to add a second identity, choose again the menu item **ModelEdit** and then **Add New Equation**. Now, click on **Key In Equation** and type the next equation, as shown in Figure 7.



KLEIN1.MOD

Create A New Model Equation

Equation # 2

PI=Y-W1-W2

Figure 7. Entering the second identity

Of course, in the present case, each of these equations are given, so that our focus is simply replication, but it is worth noticing in passing that the form shown in Figures 5 and 7 displays several other, generally useful buttons, in particular **Open Macro**, **In-**

dex, and **Find**. The first of these will allow you to open a macro—which might contain model equations—and, if so, to copy one of these to the blotter of this form simply by double clicking on it. The second, **Index**, permits you to view the index of any open data bank; for example in case you cannot remember a particular variable name. Similarly, double clicking on an indexed name will move it to the blotter of this form. In turn, **Find** permits you to make a keyword search of the databank documentation of series, in order to locate variables, either to verify their existence or their names.

Once you have verified your typing of the second identity and have pressed the **OK** button, you should next see the form shown in Figure 8, which allows you to control where in the model the equation will be put. This particular form did not appear during the process of entering the first model equation, for the obvious reason that locating the equation did not require any human judgment. Here, however, there is some ambiguity, even if only a little. Evidently, you have the choice of putting the new equation either before or after the single, existing model equation. By default, MODLER assumes that you wish to put the equation *after*, at the end of your model equation(s). Click **OK** to accept this choice.

Incidentally, it is generally a bad idea to use this form to attempt to locate model equations at "future" locations, such as 3,4,5, or 6. At the moment, the model has only 2 equations and you can confuse both MODLER and yourself if you attempt to manipulate and display a model that contains separated equations interspersed with blank placeholders. Generally, the program only "knows" the number of equations in a model from the highest number equation inserted; it has no way of determining that a blank equation is simply a place holder and will attempt to print blank equations as well as given equations. In addition, as you add further equations, MODLER will tend to increment the number of model equations even if the last added equation is placed in a previously "blank" location within the range of model equations. Furthermore, until you are an experienced user of the program, you are likely to find it difficult, after the fact, to delete "blank" equations.

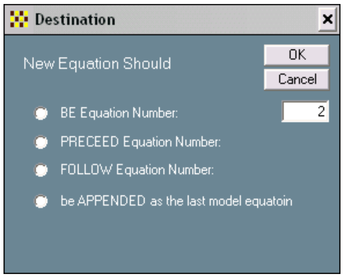
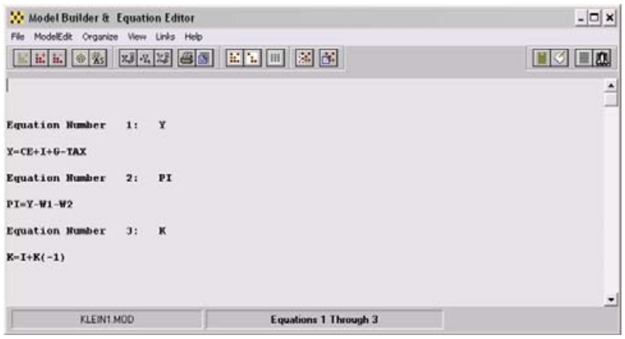

Destination
New Equation Should
OK
Cancel
○ BE Equation Number:
2
○ PRECEED Equation Number:
○ FOLLOW Equation Number:
○ be APPENDED as the last model equatoin

Figure 8. Selecting the equation location

The final identity of the model is K=I+K(-1). If you repeat the last step, but this time enter this capital stock identity into the form shown in Figure 7, instead of the prior one, and then click **OK**, you will be shown once again the form displayed in Figure 8; the default this time will be to add the final identity as equation number 3. To do this, simply click **OK**.

We have now created a three-equation model, consisting entirely of identities. At this point, if you once again click on the **View** menu item and then **Complete Model**, you should then see the model display shown in Figure 9. Notice in passing that the mnemonics shown immediately after each equation number — Y, PI, and K, respectively — specify the equation variable that that equation "explains" and therefore on which that equation is "normalized." We will consider this aspect of the model display in more detail later.

However, before we move on, notice also one other thing, which is that, so far, we have not needed to use any data. The identities have been added to the model simply as coded equations, without any need to retrieve observations from a data bank or other data source.



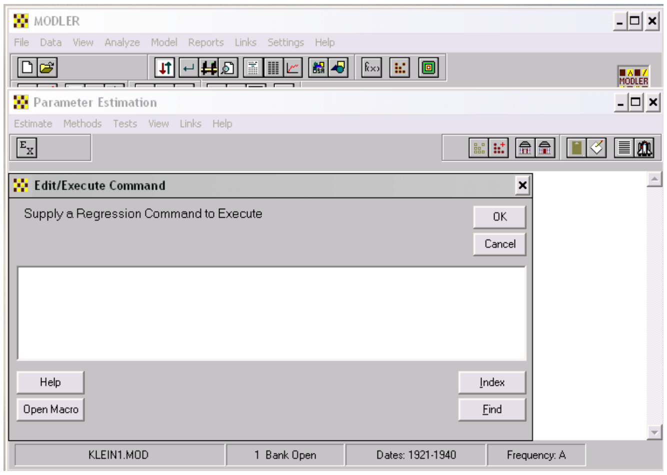
```
Equation Number   1:   Y

Y=CE+I+G-TAX

Equation Number   2:   PI

PI=Y-W1-W2

Equation Number   3:   K

K=I+K(-1)
```

Figure 9. Identities for Klein Model I

Estimating the Behavioral Equations

The next step is to estimate the model's behavioral equations and then, almost automatically, insert these into the model. As we will see, one of MODLER's more powerful features is its ability not only to estimate the parameters of equations, but also to *autocode* those equations, in the process placing them in the model fully formed. This autocoding performs at least 3 distinct operations. First, it takes the regression command and transforms it automatically into a mathematical expression of the estimated model equation. Second, it embeds in that expression the numeric parameter values computed during the estimation process. Third, it adds the equation to the model, in the location you choose, either explicitly or by default. Simultaneously, it also creates a machine language analogue of the equations, and performs various behind-the-scenes housekeeping chores. Inasmuch as these operations are performed no matter how complex the estimated equation, this is a MODLER feature that required a substantial amount of time to design and code, notwithstanding that it is now performed daily in nanoseconds. It is an operation that is not analytically performed by any other existing econometric software package.

The estimation of equation parameters is invoked by clicking again on **ModelEdit**, then **Add Equation**. However, this time choose **Estimate Equation**, rather than **Key In Equation**. Immediately, your screen should look much like the display shown in Figure 10. Notice, in the center of the screen, the form entitled Edit/Execute Command, which is superimposed on the form labeled Parameter Estimation, which is itself superimposed on the Model Builder & Equation Editor screen. Focus first on this Edit/Execute form. Should you be unfamiliar with the syntactical conventions of MODLER regression commands, press its **Help** button for information, including descriptions of the logical, relational, and mathematical operators and statistical functions that can be used in regression commands, either on their own or in combination. Notice also that you can directly open a macro from this form, by pressing the **Open Macro** button, should you wish to copy a regression command, whole or in part, that is contained in some macro file. The **Index** and **Find** buttons can be used to discover the names of databank variables that you might imperfectly remember. All these facilities are likely to be familiar from your earlier use of MODLER; the important point is that all these options are also immediately available at this stage.



Supply a Regression Command to Execute

KLEIN1.MOD | 1 Bank Open | Dates: 1921-1940 | Frequency: A

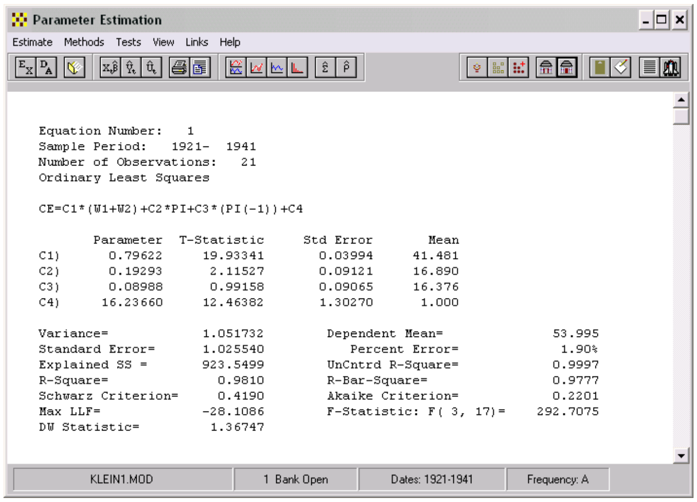
Figure 10. Parameter Estimation

Type into the Edit/Execute Command Form the regression command:

```
CE=f(W1+W2,PI,PI(-1))
```

If you should make a mistake, you can of course use the normal text editing facilities in this context, invoked by a click of your right mouse button, including the ability to cut, copy, and paste. Unless you have changed standard MODLER defaults, any variable names or other text typed into this command form will be case insensitive, so that it does not matter whether or not you capitalize what you type. But be sure to include the double parentheses at the end of the command, or else MODLER will tell you that your parentheses do not match; in that case it will also provide you the ability to correct this error.

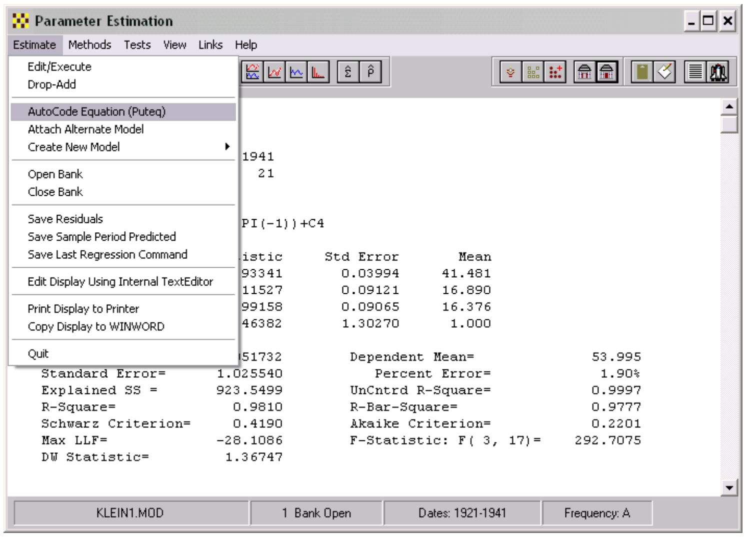
When you have correctly entered the regression command and pressed the **OK** button, you should immediately see the parameter estimation screen displayed in Figure 11. Observe that the estimated parameter values are not exactly those given earlier, reflecting that here they have been estimated using Ordinary Least Squares, rather than maximum likelihood [19, p. 67-68], however they do match the OLS estimates given by Klein (Page 75), as well as in Berndt and other relevant texts [4, Chapter 10]. As a separate exercise, you can compute a variety of different estimates, if you wish. For example, as mentioned earlier, the macro files that complement this workbook also include one that permits you to compute Two Stage Least Squares estimates, and with some work you can also compute the maximum likelihood estimates calculated by Klein. These options will be considered in a later chapter.



```
Equation Number:   1
Sample Period:   1921-  1941
Number of Observations:   21
Ordinary Least Squares

CE=C1*(W1+W2)+C2*PI+C3*(PI(-1))+C4

       Parameter  T-Statistic     Std Error       Mean
C1)      0.79622     19.93341       0.03994     41.481
C2)      0.19293      2.11527       0.09121     16.890
C3)      0.08988      0.99158       0.09065     16.376
C4)     16.23660     12.46382       1.30270      1.000

Variance=             1.051732       Dependent Mean=            53.995
Standard Error=       1.025540          Percent Error=           1.90%
Explained SS =        923.5499       UnCntrd R-Square=          0.9997
R-Square=               0.9810       R-Bar-Square=              0.9777
Schwarz Criterion=      0.4190       Akaike Criterion=          0.2201
Max LLF=              -28.1086       F-Statistic: F( 3, 17)=   292.7075
DW Statistic=          1.36747
```

Figure 11. First Estimated Equation

Considering the Parameter Estimation screen shown in Figure 11, notice, in particular, that there are many supplementary options available to you at this point. For instance, you can view graphs of actuals and predicted, and/or the regression residuals. As just indicated, you can select alternative estimation methods, instead of the default Ordinary Least Squares. You can in addition capture this regression display and save it as a file, copy it to your word processor, or print it on your printer. Or you can copy it to the clipboard. The general *MODLER User Guide* provides a more detailed description of all these various options. Our interests at the moment are specific: please select **Estimate** on the menu bar. You should then immediately see the dropdown list shown at the left of Figure 12.

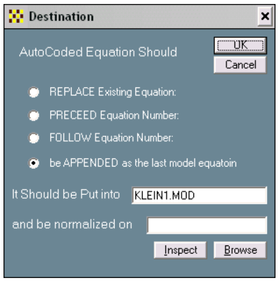


```
                             1941
                               21


                             PI(-1))+C4

                             istic     Std Error       Mean
                             93341       0.03994     41.481
                             11527       0.09121     16.890
                             99158       0.09065     16.376
                             46382       1.30270      1.000

                             51732       Dependent Mean=               53.995
   Standard Error=        1.025540          Percent Error=              1.90%
   Explained SS =         923.5499       UnCntrd R-Square=             0.9997
   R-Square=                0.9810       R-Bar-Square=                 0.9777
   Schwarz Criterion=       0.4190       Akaike Criterion=             0.2201
   Max LLF=               -28.1086       F-Statistic: F( 3, 17)=     292.7075
   DW Statistic=           1.36747
```

Figure 12. AutoCoding and Other Options

Focusing on the dropdown list, look specifically at the third item, **AutoCode Equation (Puteq),** which is highlighted in the figure. If you click on this item, one of two things will occur. First, you may see the form displayed in Figure 13, which is called a *Destination Form*. Alternatively, you may see an error message telling you that the program cannot find the regression results.

This error message, should you experience it, might strike you as strange, even perverse, inasmuch as you will probably be looking directly at a screen displaying the regression results. However, this screen displays a *past* calculation, and you should also bear in mind that, to carry out any particular operation, MODLER always needs to "know" that it has everything it needs to have. Autocoding is a complex operation, as noted earlier, so that if you have displayed one or more regression graphs, or looked at something else to do with the regression results, MODLER is programmed to check to make sure that absolutely nothing has occurred simultaneously that will adversely affect the autocoding operation. Actually, here, MODLER is programmed to err on the safe side: if anything *might* have occurred, MODLER will issue this error message. However, in this event, there is a simple solution: it is to re-select **Estimate** and then click on **Edit/Execute**, to re-estimate the equation. Once the regression results have been re-displayed, click once more on **AutoCode Equation (Puteq)**; this time you should see the form displayed in Figure 13.



AutoCoded Equation Should

○ REPLACE Existing Equation:

○ PRECEED Equation Number:

○ FOLLOW Equation Number:

● be APPENDED as the last model equatoin

It Should be Put into KLEIN1.MOD

and be normalized on

Figure 13. AutoCode Equation: Location of Model Equations

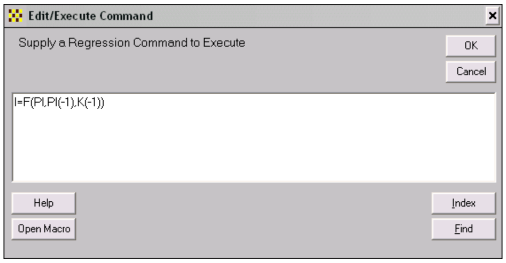
It should be immediately evident from Figure 13 that estimated equations that are Autocoded and inserted into a model are, by default, appended as the last model equation. Moreover, by default, they are loaded into the currently attached model. However, you can also see from Figure 13 both that estimated equations can be used not only to replace existing model equations, but instead can be put into some other, specified model. When you are estimating a model the first time, most of these choices are superfluous; however, if you re-estimate, or if you later wish to create alternative versions of a model, they may not be.

At present, the *normalization* option is also redundant, and can be left blank: whenever the dependent variable of an estimated equation is a simple variable, which is the case for all the Klein Model I behavioral equations, AutoCoded equations are, by default, normalized on the dependent variable. However, alternatively, such equations can be normalized on any contemporaneous variable in the equation. In a general sense, *normalization* refers to stating the equation in such as way that a particular variable's values are "explained" by the equation. Normalization is intuitively easily understood when the dependent variable is the logarithm of a particular model variable; in this case, normalization simply involves exponentiating each side of the equation. In more complex cases, normalization involves in effect "solving" the equation for the individual variable it is said to be normalized on. This process can include inverting the equation so that a variable formerly on the right-hand-side occurs on the left-hand-side as the dependent variable. In all cases, equations can only be normalized on *contemporaneous* variables: inasmuch as lagged variables represent the past and lead variables the future, generally only contemporaneous variables can be "explained" by a given model equation.

Notice also that the **Inspect** button permits you to "inspect" the model before the current estimated equation is added. If this button is enabled and is pressed, the Model Building & Editing screen becomes the active "window," thus allowing you to view model equations, variables, and other properties of the model. You can, for example, verify that the equation you are about to add is not already represented in the model or, if it is, determine its equation number, so that you can replace the existing version with the just-estimated equation. However, there are instances (such as at present) in which the Model Building & Editing screen is already displayed on your desktop, although momentarily it may be behind another screen. In this case, either the Inspect button may be disabled or, if it is enabled, and you press it, MODLER may display an error message to the effect "Model Building & Editing Screen Already Displayed. Select Status Bar Icon." You can make this screen active simply by clicking on the appropriate desktop status bar icon; it is not necessary to display another copy of this screen.

Finally, observe the **Browse** button on the Destination Form shown in Figure 13. It is there to allow you to browse your hard disk or other storage device in order to determine the location of some other model into which you could "put" the current estimated equation. Bear in mind that whereas a model file is presumed to be a container for a "model," there is no reason why you cannot create one or more "extra" MOD files, should you wish, simply to use as repositories for sets of estimated equations. There is obviously no requirement that any given MOD file must at some point be used to hold a solvable model. Recall the earlier, brief discussion of the MODLER command that permits one or more equations to be copied from one MODLER-created model to another: it is perfectly permissible to use a given MOD file to hold any number (up to 1000) of alternative versions of an equation, and then at the end to copy your final choice from among them to another MOD file that then will be used as a "true" model file.

However, these are collateral issues: there is still work to be done building Klein Model I. Specifically, there are two further behavioral equations that are part of this model. To estimate these, and include them in the model, essentially requires simply replicating, with only a few variations, the process just described for the Consumption Expenditures equation. That is, click again on **ModelEdit**, then on **Add Equation** and finally **Estimate Equation.** The Edit/Execute Command form should once again appear, as displayed in Figure 14. Here the Net Investment equation has already been keyed in; on your computer, you will obviously have to do this yourself.



Supply a Regression Command to Execute

```
I=F(PI,PI(-1),K(-1))
```

Figure 14. Regression Command for Net Investment Equation

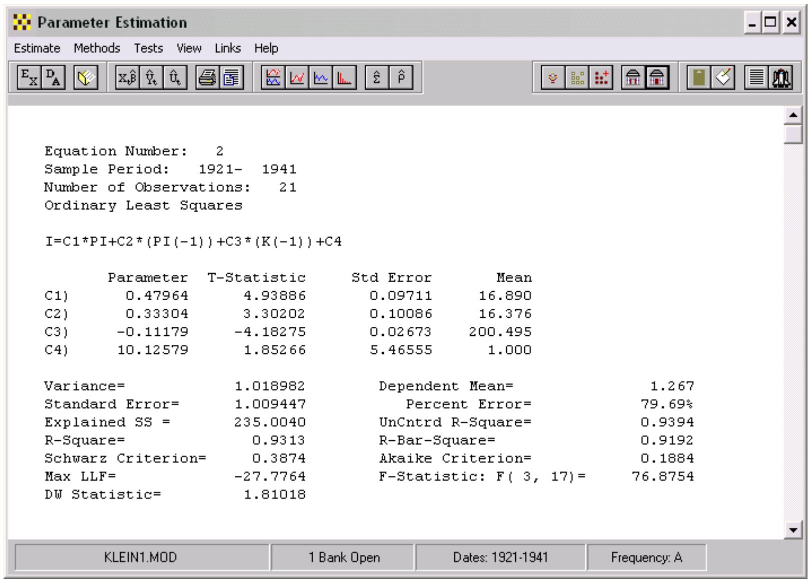
Recall that the text box (blotter) of this form permits you to edit its contents, should you make a mistake: to accomplish this, click on your right mouse button while pointing at this blotter and the standard editing options floating menu will appear. This feature is a MODLER capability in combination with Windows, generally requiring Windows 98 or later. However, provided this operating system criterion is met, it is a standard feature of MODLER: in almost every instance that you are called upon to insert text, you are able to edit, cut, copy, and paste.

Notice also that the form in Figure 14 exhibits a set of buttons, including in particular the **Help**, **Open Macro**, **Index**, and **Find** buttons. The first provides context specific help that describes the conventions, syntax, and options for regression commands specifically, including the set of operators and implicit functions that can be used. The **Open Macro** button permits you to open any macro, and if you highlight any part of its contents, to copy this highlighted text automatically. Double clicking on any line's contents will capture that line of macro text. For example, you can use this button to extract a regression command contained in that macro; of course, this operation leaves the macro undisturbed. As always, the **Index** and **Find** buttons permit you to discover the names and characteristics of variables in any open data bank or Memory File. These buttons are described in much greater detail in the *MODLER User Guide*; they are mentioned here once more just to remind you of their utility when building a model.

Now that you have entered this Investment equation, and verified that it is correct, click **OK**. Make sure, before clicking **OK**, that you have entered the ending, right double parentheses, as shown. If you omit one or both of these, MODLER will issue an error message to the effect that "parentheses do not match," and then allow you to correct this error.

You should obtain the results shown in Figure 15. You can easily confirm that these match the OLS estimates presented by Klein (Page 75), as well as by Berndt [4, Chapter 10] and others subsequently. At this stage, you might pause to note that the standard MODLER estimated equation display shows the equation more or less in the form that will be inserted into a model, once the AutoCode (PutEq) option is chosen.

The essential difference is that here the parameters are shown symbolically (C1, C2, … , CN). The rows just below this equation display, beginning with these symbols as labels, present the parameter estimates and related statistics.



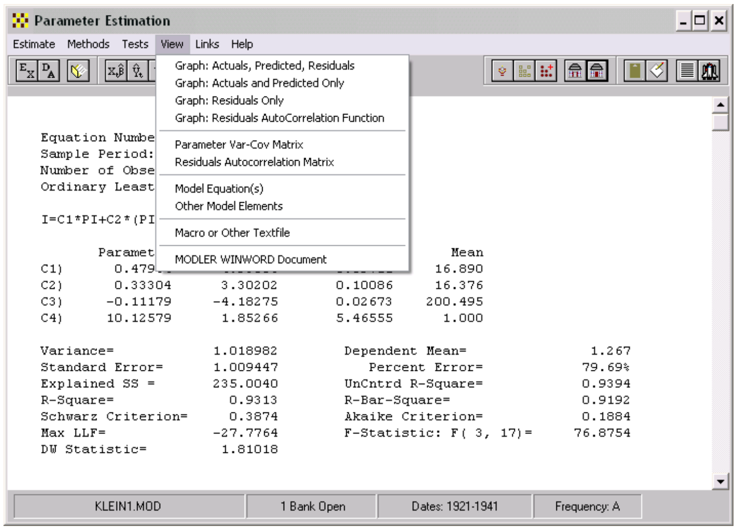
```
Equation Number:   2
Sample Period:   1921-  1941
Number of Observations:   21
Ordinary Least Squares

I=C1*PI+C2*(PI(-1))+C3*(K(-1))+C4

       Parameter  T-Statistic     Std Error       Mean
C1)      0.47964      4.93886       0.09711     16.890
C2)      0.33304      3.30202       0.10086     16.376
C3)     -0.11179     -4.18275       0.02673    200.495
C4)     10.12579      1.85266       5.46555      1.000

Variance=            1.018982       Dependent Mean=              1.267
Standard Error=      1.009447          Percent Error=           79.69%
Explained SS =       235.0040       UnCntrd R-Square=           0.9394
R-Square=              0.9313       R-Bar-Square=               0.9192
Schwarz Criterion=     0.3874       Akaike Criterion=           0.1884
Max LLF=             -27.7764       F-Statistic: F( 3, 17)=    76.8754
DW Statistic=         1.81018
```

Figure 15. Estimated Net Investment Equation

Once you have verified your results, then successively select the menu item **Estimate** and its dropdown element **AutoCode (Puteq)**. You will at this stage once again see a Destination form similar to that shown in Figure 13 above. Click **OK** in order to insert the estimated version of this equation into the model. If you accept the default, this operation will have the effect of appending this equation to the model. The model should now contain 5 equations: three identities and two estimated equations.

As you work through this example, you may notice that your screen, equivalent to that shown in Figure 15, can be extended downwards, using your mouse to click on the bottom edge and pull down. If you pull this screen down, using your mouse to drag the bottom edge, you will see that, in addition to the supporting statistics shown in the above figure, there are in fact a number of other test statistics generated by default. These include Residual Properties, Function Form, Heteroscedasticity, and Structural Stability test statistics. The default set shown is fairly comprehensive, as you will see, but if you click on the **Tests** menu element of this screen in addition and then on each of the associated drop down menu elements that appear, you will discover that you optionally have a great deal of control over the particular test statistics that are shown and their characteristics, as well as various additional test facilities, including Unit Root and related cointegration tests. These are not discussed here for the simple reason that the purpose of the present description is to describe how to form a model using the Klein Model I as a template. However, were you to use MODLER to create your own model, then these test statistics would be quite relevant and you would want to use them in order to make choices concerning which form of each equation you might wish to add to your model. Obviously, on your own you can consider all these tests in conjunction with the Klein model results; a recent book considers aspects of these statistics and their presentation [30].

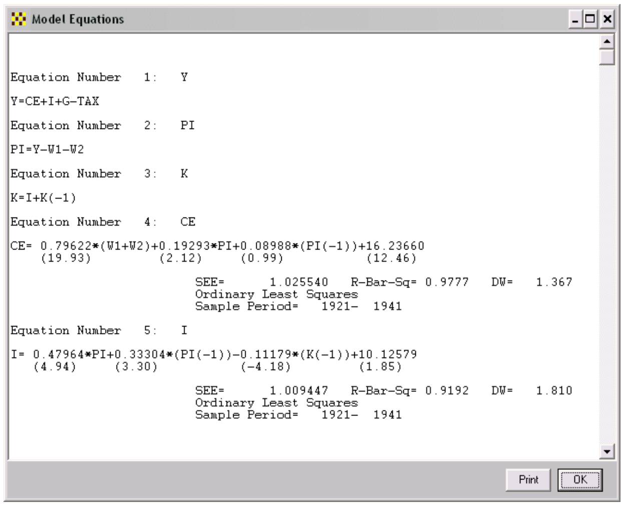
If you also then click on the **View** menu item shown in Figure 15, you will then see the drop down menu shown in Figure 16. Among the options available are those that can be accessed by clicking on **Model Equation(s)** and **Other Model Elements**. Your display may look slightly different, because of a double row of icon buttons, but will be essentially the same in function.



```
Equation Numbe
Sample Period:
Number of Obse
Ordinary Least

I=C1*PI+C2*(PI

       Paramet                                  Mean
C1)      0.479                                16.890
C2)      0.33304      3.30202      0.10086    16.376
C3)     -0.11179     -4.18275      0.02673   200.495
C4)     10.12579      1.85266      5.46555     1.000

Variance=             1.018982      Dependent Mean=             1.267
Standard Error=       1.009447         Percent Error=          79.69%
Explained SS =      235.0040      UnCntrd R-Square=           0.9394
R-Square=             0.9313      R-Bar-Square=               0.9192
Schwarz Criterion=    0.3874      Akaike Criterion=           0.1884
Max LLF=            -27.7764      F-Statistic: F( 3, 17)=    76.8754
DW Statistic=         1.81018
```

Figure 16. "View" Related Facilities

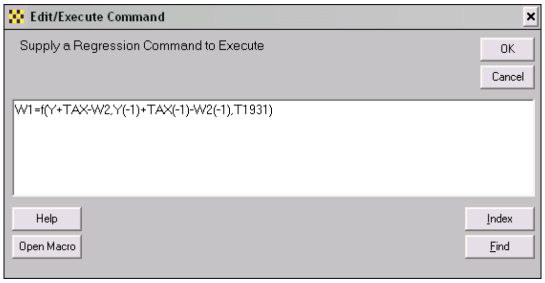
If you choose the first of these two choices, you should then see the supplementary screen shown in Figure 17, although you may need to pull down its bottom edge before the complete set of equations shown here is displayed. What this figure obviously illustrates is the complete set of equations for the Klein model that we have so far either keyed in or estimated. Alternatively, if you choose **Other Model Elements**, the Model Builder & Editor screen will be displayed, allowing you, if you wish, to examine the model in detail.



```
Equation Number   1:   Y

Y=CE+I+G-TAX

Equation Number   2:   PI

PI=Y-W1-W2

Equation Number   3:   K

K=I+K(-1)

Equation Number   4:   CE

CE= 0.79622*(W1+W2)+0.19293*PI+0.08988*(PI(-1))+16.23660
    (19.93)         (2.12)     (0.99)            (12.46)

                         SEE=      1.025540   R-Bar-Sq= 0.9777   DW=   1.367
                         Ordinary Least Squares
                         Sample Period=   1921-  1941

Equation Number   5:   I

I= 0.47964*PI+0.33304*(PI(-1))-0.11179*(K(-1))+10.12579
   (4.94)     (3.30)           (-4.18)          (1.85)

                         SEE=      1.009447   R-Bar-Sq= 0.9192   DW=   1.810
                         Ordinary Least Squares
                         Sample Period=   1921-  1941
```

Figure 17. Model Equations

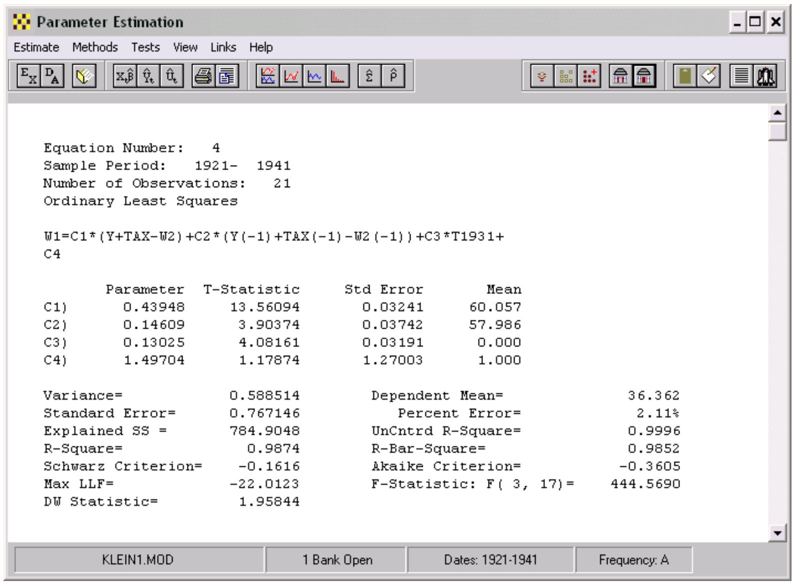
Alternatively, when you return to the Model Builder & Editor screen you can continue to estimate the equations of Klein Model I. The final equation can be estimated by first clicking once more on the **ModelEdit** menu item of this screen, then on **Add Equation**, and finally **Estimate Equation.** The Edit/Execute Command form should yet again appear, as displayed in Figure 18, which shows the Private Wage equation as a regression command. Notice that the equation shown here varies slightly from the wage equation given earlier in the context of Klein's statement of Model I. In particular, this command includes a variable T1931, rather than the expression T-1931. This variation involves a slight compression in the statement of the model; however, it is fundamentally cosmetic. Klein defines T as taking the successive values 1920, 1921, … Therefore the term T-1931 instead takes the values –11, -10,…,0,1,2, … Replacing this composite term with a simple variable (called T1931) obviously makes no essential difference to the model, or its use. However, should you wish, you can replace this term with the explicit expression, although in that case you would also need to create a new variable T. The possible virtue of using T, rather than T1931, is that the data set for the model would then more obviously include a time variable T=1920, 1921, … and the explicit statement of the model might be marginally more readable. Of course, the variable could also be called TIME, rather than T.



```
W1=f(Y+TAX-W2,Y(-1)+TAX(-1)-W2(-1),T1931)
```

Figure 18. Private Wage Equation Regression Command

Click **OK** to obtain the results shown in Figure 19 then select **Estimate**, **AutoCode (Puteq)**, and again click **OK** on the Destination form, in order to insert the estimated version of this equation into the model.



```
Equation Number:   4
Sample Period:   1921-  1941
Number of Observations:   21
Ordinary Least Squares

W1=C1*(Y+TAX-W2)+C2*(Y(-1)+TAX(-1)-W2(-1))+C3*T1931+
C4

       Parameter  T-Statistic     Std Error       Mean
C1)      0.43948     13.56094       0.03241     60.057
C2)      0.14609      3.90374       0.03742     57.986
C3)      0.13025      4.08161       0.03191      0.000
C4)      1.49704      1.17874       1.27003      1.000

Variance=             0.588514       Dependent Mean=             36.362
Standard Error=       0.767146          Percent Error=            2.11%
Explained SS =        784.9048       UnCntrd R-Square=           0.9996
R-Square=               0.9874       R-Bar-Square=               0.9852
Schwarz Criterion=     -0.1616       Akaike Criterion=          -0.3605
Max LLF=              -22.0123       F-Statistic: F( 3, 17)=   444.5690
DW Statistic=          1.95844
```

Figure 19. Estimated Private Wage Equation

One aspect of the equation displayed in Figure 19 that should be noted particularly is the inclusion of the expression Y+TAX-W2, in both contemporaneous and lagged form. Obviously, the ability to embed such expressions has the effect of making the estimated equation more instantly readable—as opposed to if this expression were to be replaced by a variable called XXX, or some other mnemonic name. However, as discussed previously, the fundamental virtue of being able to embed expressions directly is that this MODLER capability permits the model to be parsimoniously stated in its entirety, which is more than a matter of simple elegance. Extra variables essentially require extra definitional identities that litter the model and take up space in the model databank. More importantly, as a conceptual matter, they also make the model more difficult to read and understand. At the beginning of the next chapter, we will see in Figure 20 that MODLER's text presentation of the model is virtually identical to what might be published in a book describing the model, without requiring any specific editing. A fundamental reason is the software's automatic ability to interpret expressions within commands, as well as to display them.

A lot more could be said about the estimation process, especially in terms of the documentation of the model as a nearly automatic by-product. For example, at each stage, it is possible to capture the estimated equation display. As a general software feature, this display can be printed on your printer, copied to the clipboard, or directly transferred to your word processor, assuming of course that when you originally set up MODLER, you established all the proper settings, as described in the *MODLER Getting Started Guide*. The purpose of capturing the results at each stage might be to produce full documentation of each step of the model building process, either for record keeping or publication.

As you have been reading this workbook you have actually been watching this documentation occur, the only difference being that screens have been displayed here for the purpose of illustrating how your monitor will look at each step, rather than presenting the results in the form of text and graphics suitable for publication as a book or report. The presentation here has also subsumed other self-documenting MODLER features. For example, the Standard Index to the KLEINBNK data bank, when printed on your printer, or copied to your word processor, becomes in effect a glossary of model variables. Of course, the usefulness of such a glossary always depends on the person who creates and maintains the model databank, upon that person's willingness to describe the variables well.

In the case of the Klein model, all this documentation may be a little superfluous, inasmuch as the model is already published. However, should you later construct a new model, one that you create from scratch, these facilities are there for that model as well. Once you become accustomed to using them, they generally will involve only a few extra keystrokes, in return for which you obtain a more or less complete audit trail of your work – as you work, not as a separate, possibly tedious effort. Furthermore, not only can you save the regression results; you can also save the regression commands. Look once more under **Estimate** on the Parameter Estimation screen and you will see that a further option is to **Save Last Regression Command**. If you save these commands at each step, they will be inserted into one or more macro files. Subsequently, if you wish, you will later be able to re-estimate some or all of the equations you estimated simply by running (or re-running) these macros.

Chapter 2 Forming the Model

The model estimated in the first chapter is displayed below in Figure 20. The creation of this model, as described in the last chapter, involved estimating its unknown parameter values using Ordinary Least Squares. As was noted there, this method is not the only possible choice: the parameters could instead have been estimated in any of a variety of ways, including Two Stage Least Squares or Three Stage Least Squares, as well as Limited Information Maximum Likelihood, among other methods. Furthermore, the implications will not be the same: the particular estimation method used affects the specific parameter estimate values obtained and also the stochastic properties of the resulting model when it is solved. A useful supplementary and amplifying discussion is provided by Berndt [4], who considers the various methods that have been used by economists over the years to estimate Klein Model I, pulling together and referencing a number of previous studies. After considering the alternatives, it is possible, should you wish, to replicate at least certain of these methods using MODLER, either now or later. Furthermore, if you are willing to take the trouble to key in all the values, it is perfectly possible to form the model in the context of MODLER using *any* parameter estimates you have access to, not just those that can be produced by the program.

However, there is a sense in which the discussion in this chapter is unaffected by the particular estimation method used, for there is a distinction between estimating a model and solving it. In particular, once a model's parameters have been estimated, it can be regarded simply as a set of mathematical equations. This is not to say that the parameter estimates, and indeed the properties of the parameter estimators, do not affect the properties of a model when it is solved; it is only to assert that as a set of equations the model will have certain characteristics that need to be considered as a separate topic from those associated with parameter estimation. Furthermore, it is also important to recognize that, generally, this set of equations will form a nonlinear model, whether or not the parameters have been estimated using linear estimation methods. Of the historical macroeconometric models you are likely to come across, Klein Model I is one of the most linear. Other models commonly incorporate a variety of nonlinear elements, including both ratios — such as occur in the identity that specifies the formula for the unemployment rate — and logarithmic transformations —such as those that may be introduced if a model includes Cobb Douglas, CES, or other production relationships.

```
  Klein Model I

Equation Number   1:   Y

Y=CE+I+G-TAX

Equation Number   2:   PI

PI=Y-W1-W2

Equation Number   3:   K

K=I+K(-1)

Equation Number   4:   CE

CE= 0.79622*(W1+W2)+0.19293*PI+0.08988*(PI(-1))+16.23660
      (19.93)           (2.12)      (0.99)          (12.46)

             SEE=      1.025540   R-Bar-Sq= 0.9777   DW=   1.367
             Ordinary Least Squares
             Sample Period=   1921-  1941

Equation Number   5:   I

I= 0.47964*PI+0.33304*(PI(-1))-0.11179*(K(-1))+10.12579
     (4.94)       (3.30)         (-4.18)           (1.85)

             SEE=      1.009447   R-Bar-Sq= 0.9192   DW=   1.810
             Ordinary Least Squares
             Sample Period=   1921-  1941

Equation Number   6:   W1

W1= 0.43948*(Y+TAX-W2)+0.14609*(Y(-1)+TAX(-1)-W2(-1))+0.13025*T1931+1.49704
      (13.56)               (3.90)                         (4.08)        (1.18)

             SEE=      0.767146   R-Bar-Sq= 0.9852   DW=   1.958
             Ordinary Least Squares
             Sample Period=   1921-  1941
```

Figure 20. Complete Klein Model I

Further Elaboration of the Model

As just indicated, a model can be estimated using only linear parameter estimation methods, yet be a highly nonlinear system, once formed as a working model. Conversely, it is at least conceivable that a model could be estimated using non-linear estimation procedures, yet be a linear model when considered as a set of equations. However, before addressing these several models-as-sets-of-equations issues, it is useful to talk first about the inferences you can draw from the way in which models are displayed. Consider Figure 20 simply as a display. Observe that the behavioral equations are shown with a standard set of supporting statistics; this figure exactly replicates MODLER's standard model display. Furthermore, you need to be aware that, within MODLER, such statistics will be displayed for any particular equation only so long as you do not subsequently edit that equation, in the process changing its properties. Of course, this particular display, as you view it, has been copied to a word processing document and there it could of course have been changed without necessarily causing the associated statistics to vanish, but when considering such a model display entirely within MODLER, you have some assurance that you will know whether each equation is displayed as originally estimated. In particular, if supporting statistics are not present you know that the equation was not estimated and autocoded by MODLER — or that it has been changed in some essential way since estimation; as prosaic as this facility may initially seem, it is nevertheless a model validation feature.

Notice also that, for each equation, the mnemonic for the variable that is "explained" by that equation is shown immediately to the right of the equation number, as very briefly indicated earlier. It is characteristic of this type of MODLER display that the autocoded behavioral equations will *always* be shown in the form they were estimated. Thus, after the fact, you can always discover the estimation specification. However, what is also implied is that if the equation has subsequently been normalized or later re-normalized, this normalization will be indicated by the mnemonic that appears immediately after the equation number for each equation. This subtlety has the possible defect that, for a normalized equation, you are not shown the specific normalized form. The saving grace is that MODLER normalizes equations analytically, so that the normalization is unique; therefore, you are always able to derive it yourself if you wish. Actually, across software packages, there has never been an agreed standard established for such equation displays — only one or perhaps two other packages have ever permitted normalization, and then using nonanalytic methods, generally involving an initial linearization by Taylor's expansion, or by first requiring you to specify the normalization formulae manually.

Figure 20 additionally demonstrates that, so far as the variables of a model are concerned, what-you-see-is-what-you-get. As stated earlier, one of MODLER's defining characteristics is that it permits mathematical, logical, relational, and statistical expressions, alone or in combination, to be embedded in almost any commands. There are a few exceptions to this rule and certain specific restrictions in the case of a model—for instance, statistical functions such as MEAN, SDEV, and others defined on the domain of a particular estimation sample date range—cannot be used in models, for the obvious reason that in the context of a model solution, these are scalar constants and should be so represented. However, in general, you are able to form regression and other commands *that are stated in terms of the basic model variables*. As a result, models created by MODLER are easily understood, inasmuch as extraneous identities are therefore unnecessary. Furthermore, among the other implications is that usable models can be formed very quickly. The model shown in Figure 20 is ready to be solved, fresh from estimation — that is, if you wish to proceed directly, only three commands stand between it and its first sample solution.

However, the fact that MODLER permits an elegant statement of a model does *not* mean that parsimony is required. At a later stage, when you begin to create tables, graphs, and other such displays, in order to present model solution results, you may wish to create additional model variables specifically for display purposes. For instance, you may wish to monitor relationships between specific variables, for example in the form of ratios [20], even though these constructs play no specific part in the formulation of your model. In the case of Klein Model I, simple as it is, there are various ways to elaborate the model. For example, as Berndt implicitly points out [4, p. 551], this model inherently subsumes several potentially interesting additional identities, even if these are formally redundant. In particular, define Total Wages, W, as:

$$W = W1 + W2$$

and Income, Y, as:

$$Y = PI + W$$

Total Product can then be stated as:

$$Y + TAX = CE + I + G$$

and Private Product, E, as:

$$E = Y + TAX - W2$$

You will recognize each of these as having a role in the model.

Actually, as mentioned earlier, a compound variable, such as Y + TAX, cannot be used as the left-hand-side term of a MODLER identity, so that we cannot introduce the particular expression containing it into the model, which explains the particular (solved) identity that we used as equation number 1. It was briefly noted earlier that a distinction exists between writing an expression on a sheet of paper and coding it into a computer program, namely that in the former case the equal sign expresses equivalence whereas in the later case it is ordinarily an operator. However, in the context of MODLER, we can always create a variable named NP, where by construction:

$$NP = Y + TAX$$

permitting as a result the statement of the above identity as:

$$NP = CE + I + G$$

Among other things, what this particular exercise reveals is the nature of the model's identities as the element of the model that, because of the constructional accounting relationships, establishes the connections between the behavioral equations and as a result tie the model together. Notice in particular that the three identities shown in Figure 18 insure the satisfaction of all six identities, whether or not these additional identities (or any subset of the "redundant" ones) are included in the model.

However, a cautionary note that might be sounded at this point is that the choice and configuration of identities *can* affect the interpretation and also the operation of the model. From Figure 18, observe in particular that a consequence of setting up the model as we have is that demand side relationships predominate: Y is defined as the (demand side) sum of demand variables and profits will increase or decrease residually to suit the relationship between the values of Y and wages. Recall that in the modern national income accounting framework, Gross Domestic Product can be defined identically from either the demand side, the product side, or the income side, so that to the degree that these identities appear (or are left implicit) the model can express demand side, supply side, or income side characteristics when it is solved. This is too deep a subject to be considered in any further detail at the moment, but you should be aware that alternative configurations of the identities, even in the case of a model as simple and primitive as the Klein model, can affect aspects of its solution.

Nevertheless, so long as you do not, by introducing extra identities, thereby create mathematically conflicting relationships, you are free to add such redundant identities to the model. Procedurally, adding such identities involves the same steps as we took, in the last chapter, to insert the original identities into the model. A modern microcomputer is sufficiently powerful that the addition of redundant identities (so long as they do not create conflicts) will be barely if at all noticeable, so that there is no reason on this score not to add such equations. A special case, should you be interested, is described in the Appendix to this Chapter; the type of specification discussed there allows, for instance, a distinction to be made between *ex ante* and *ex post* identity relationships.

Of course, any extra identities added can affect the "readability" of the model. Therefore you should also be aware, as will be demonstrated in a later chapter, that it is not necessary to add extraneous equations if the motivating purpose is simply to permit the display of composite values in graphs and tables, or similar contexts. The ability to include expressions in MODLER commands extends even to the creation of table templates or graphical commands. The only persuasive reason why you might want to include redundant identities is if you wish to be able to display the values of the variables they define in any "dump" of model solution results. Later in this chapter, we will see that MODLER incorporates standard solution displays that do not require you to set up any templates in advance: they simply display (or "dump") the values of model variables. To the extent that a variable is included in the model, its values can be displayed by this means.

Selecting the Model

If you began with chapter 1 and have steadily worked your way to this point, using your computer to create and display Klein Model I, the model will now be attached. It will need to be attached in order for you to produce the display shown in Figure 20. However, alternatively, you can also easily select and attach the model now, provided that you previously worked through the first chapter, whether or not you exited from MODLER at the end of that chapter, then re-started MODLER just now, or else have done all the work in a single sitting.

This particular capability to select a model has not been mentioned before, but it results from the fact that, as you create and use a model, MODLER steadily monitors your progress. Initially, when you create a model, by giving it a name, MODLER will simultaneously record that act of creation, creating in addition what is called a model definition file (or DEF file). This DEF file will also be given the name of your model, so that if the model is called KLEIN1 (and its equations automatically stored in a file called KLEIN1.MOD), the DEF file will be called KLEIN1.DEF and will be used to store information about the model and files that are or will be used in connection with the model. It is too early to begin to discuss the particular nature of this DEF file, since we have so far only begun to consider the model use process, but you may have noticed a related menu element as early as Figure 1, in the form of the **Select Model** menu element. Although this would have been grayed out on your copy of MODLER if at that time you had yet to create a model, you might have noticed and wondered about this item. We will shortly begin to learn about the DEF file and the **Select Model** command, but it is best to consider the model compile operation first.

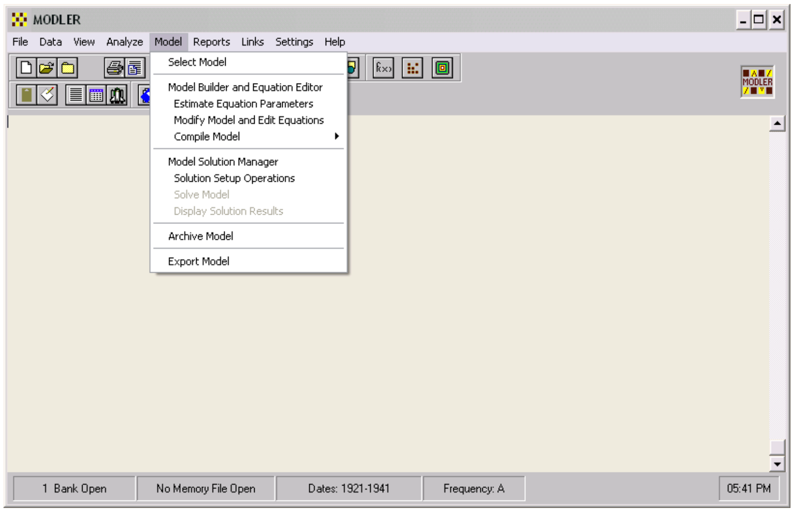
Compiling the Model

Reflecting the work that MODLER has done behind the scenes, the model displayed in Figure 20 is actually ready to solve, in the sense of being a complete, self-consistent set of equations. However, as displayed, the model is not yet in an *algorithmic* form that permits solution. It must first be *compiled*, which is essentially the process of creating a machine-language representation that MODLER can use. Although you cannot see it, standing behind the visual display of the model in this figure is an equation coding that, among other things, changes the everyday Infix notation that we are used to seeing on the printed page into what is called Reverse Polish notation. This behind-the-scenes conversion subsequently permits binary calculation and the proper management of what is called the "stack." You do not need to understand these rather technical terms and numerical operations in order to use MODLER. You need only to be generally aware that the program's display of the model as text in Figure 20 is simply an onscreen display to permit human understanding of the model; the equivalent operative form of the model is quite different, internal and proprietary to the MODLER computer code. Nevertheless, it may be helpful to know something about certain aspects of the internal model organization.

During estimation and initial model formation, MODLER gathers into a single file the individual equations of the model, treating them each as separate entities and holding them in separate records, a "record" technically being a subcomponent of a "file." This file is referred to as the MOD file, and will here take the name KLEIN1.MOD, on the assumption that the model has been named KLEIN1. Later, in response to a particular command, MODLER will then transform the separate equations into a system of simultaneous equations, by recognizing and organizing the linkages, the purpose being to create an interconnected, multi-equation "model" that can then be treated as a composite thing and "solved." Overall, this process is called "compiling," and a model once organized into this interconnected form is called a "compiled" model. This compiled set of equations, once created, has no separate visual representation, although the model's ordering of the equations for solution can be determined, as will be described. The model, once compiled, is then stored in another type of file, which will automatically be given the filename of the MOD file, plus the extent CMF, the filename part thus nominally pairing the two model files. In the present case, the compiled model file (CMF) will thus be known as KLEIN1.CMF. Generally, to solve and otherwise work with an existing model both its MOD and CMF files must be located in the models directory on your hard drive.

It should be clear from this brief description that a model's CMF file is generally a behind-the-scenes object. It will subsequently be managed by MODLER, without any instructions from you and you almost will be able to forget its existence. However, because it is necessary to the solution of a model, it is useful to be at least vaguely aware of it and, for example, you should never erase such a file from your hard disk without purpose. Furthermore, if you wish to copy a working model from one machine to another, you should always copy *both* the MOD and CMF files. Once a model has been compiled, these files *together* are the model. Actually, one of the purposes of the DEF file, referenced in the previous section of this chapter, is to manage all such model files, so that in practice you may not need ever to refer directly to the CMF file, but its general purpose should now be clear.

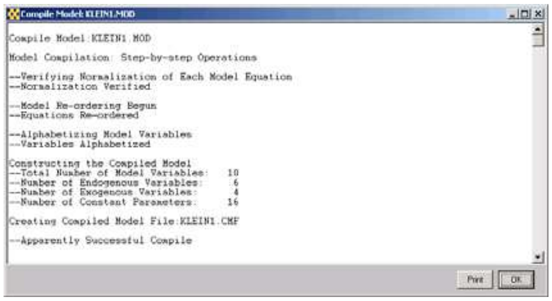
You do control when a model is compiled and compilation can be invoked in several different ways. If you inspect the menu elements of the several MODLER screens, you will discover that the word *Compile* appears in several different places, but it really does not matter from which of these menu elements compilation is actually instigated. However, it occasionally might be easiest to return to the Central Control screen from the Model Building and Editing Screen, and then to click on the menu item **Model**, thus causing the appearance of the drop down menu displayed above in Figure 21. Note the choice **Compile**. If you click on this choice and then click on **Execute**, Klein Model I will begin to compile. A requirement is that a model's MOD file be attached: the menu item **Compile** will be grayed out if a MOD file is not attached, so that a necessary precondition is that a model already be selected – but this will necessarily be the case if you just finished displaying the model, as was done in Figure 20.



1 Bank Open | No Memory File Open | Dates: 1921-1941 | Frequency: A | 05:41 PM

Figure 21. Compiling Klein Model I

The compilation process involves several logical stages, which are briefly indicated in Figure 22; whenever a model is compiled, this screen appears immediately afterwards. The purpose of the first indented operation mentioned in Figure 22 is to verify that each of the equations in the model is properly normalized. Inasmuch as this verification involves determining that each model equation can be associated with a unique model variable, this step results in the classification of the model's variables into the categories of endogenous and exogenous. In fact, this is a limited classification: in the context of preparing a model for solution, MODLER does *not* attempt to determine that a variable is truly exogenous or endogenous, in any ultimate statistical sense. Instead, MODLER merely determines that *for the given model* there are two classes of variable: those that are dependent or jointly dependent variables (the endogenous) and those that are independent (the exogenous). The number of endogenous must equal the number of model equations. The exogenous, in this context, are essentially the variables that have not been classified as endogenous.



```
Compile Model:KLEIN1.MOD

Model Compilation: Step-by-step Operations

--Verifying Normalization of Each Model Equation
--Normalization Verified

--Model Re-ordering Begun
--Equations Re-ordered

--Alphabetizing Model Variables
--Variables Alphabetized

Constructing the Compiled Model
--Total Number of Model Variables:   10
--Number of Endogenous Variables:     6
--Number of Exogenous Variables:      4
--Number of Constant Parameters:     16

Creating Compiled Model File:KLEIN1.CMF

--Apparently Successful Compile
```

Figure 22. Compilation Notification

Functionally, this classification process is important: provided that the set of equations is mathematically self-consistent and thus ostensibly solvable, MODLER organizes the model in such a way that it can be solved, all other things equal, without requiring you to say in advance which variables have which properties. Among its consequences, this convention allows a model to be created from any set of equations, including for example, a set of identities, and then solved, in order to determine, for instance, that these have been coded correctly. In the present case, the model that is the subject of this workbook is a well-known model, and if you have been careful coding it, it is likely to solve the first time you attempt a solution. However, in general, there are thousands of reasons why an arbitrary, or even very carefully developed, set of equations might not form a model capable of solution. In the general case, solving a newly created model is a touch-and-go operation, especially the larger the model. Very few large models solve on the first attempt, for reasons we will go into later.

The second stage of the compilation process, as documented in Figure 22, involves re-ordering the equations of the model so that these are placed in, more or less, as recursive an order as possible. In fact, this re-ordering process is optional. You can choose not to have MODLER re-order the model equations, allowing you instead to order the equations yourself. Alternatively, re-ordering can simply be omitted: not all solution methods require the model to be recursively ordered. It is even possible for you to organize your model so that it has functionally separate segments that may include an initial segment that is solved once each solution period, a second segment that iterates to solution, and a third segment that is simply solved once each period, after all or part of the rest of the model has iterated to solution. These complexities will not be considered further in this workbook, but it may be ultimately relevant to your work that you be aware that MODLER can provide a user with access to such options.

Incidentally, whenever MODLER re-orders the equations of your model it automatically creates a file, called ORDMOD.TMP, that has all the characteristics of the original MOD file, *except* that it contains the equations as an ordered set. If you rename ORDMOD.TMP to the original name of the MOD file (having first carefully saved the MOD file under some other name, or in another directory), you will be able to use it just as you would the original MOD file, including being able to print the model in ordered form, generally not the order of the equations originally.

The third operation, that of alphabetizing the model variables, strictly speaking has nothing to do with the process of solving the model. However, in order to keep track of the model's variables, when displaying the solution results, or at other times that you might wish to display values, it is generally helpful to be able to find them quickly, which is generally enhanced by this alphabetic ordering of the variable names. Therefore, alphabetic ordering of the variables is a part of organizing the model in terms of its use as a solvable model.

The final step involves creating a new model file, the *Compiled Model File* (or CMF). As indicated earlier, the CMF file has a specific proprietary structure, entirely machine-readable. It contains the model in a special *machine language* form and is used only during the model solution process, to include the creation of solution files and other operations associated with solving the model or displaying the solved model. In general, as mentioned earlier, MODLER manages the use of the CMF file, attaching it or detaching it as appropriate to a particular operation. For most purposes, it is sufficient that you simply know of the existence of this file. However, in certain cases knowing about this file can be important. In particular, as previously indicated, should you wish to copy a model from one computer to another, especially a model that has been compiled, you should copy both the MOD and CMF files; they will both have the same name, except that one will have the extent MOD and the other CMF. This file will (read *should*) be located in the same directory or folder on your hard drive. Incidentally, the process of *archiving* a model, using MODLER, will be described in greater detail later.

Creating a Solution File: Background Information

So far we have focused on the model code, the MOD and CMF files, and have not mentioned the data. The data associated with a solution are contained in one or more *solution files*. Each of these files may be given any filename that Windows and MODLER together accept as valid, but must have the extent SOL; obviously, among

SOL files located in a particular directory, this filename must be unique. The practical process of creating a solution file is described in the next section. However, you may first find it helpful to understand the relationships between this file, the CMF file and the MOD file. The CMF file, as just described, simply contains the algorithmic code of the model. It does not contain any observations on the variables. In contrast, the solution file is explicitly a data file. It contains data on the variables in the model. Furthermore, it contains data and nothing else: values of both the endogenous and exogenous variables plus the values of any constant or multiplicative adjustments. For the moment, the possibility of adjustments is a complication that can be ignored, but if you later work with larger models in a more sophisticated setting, making model solution adjustments may become interesting to you.

The creation of a solution file, or SolvFile, as it will often be called, must occur after a model is compiled. In the simplest case, once a CMF file has been created, the next step is to create a solution file, which is to say that we will need to load such a file with observations on the model variables. Initially, this file will be unique, but ultimately, there can be many solution files associated with any given pair of MOD and CMF files; however, each solution file will be associated with a particular set of values for all the model variables and it *can only be used in conjunction with one particular CMF file*.

The data source for the initial solution file is ordinarily the same as that for the values used for the model's estimation, plus any additional values associated with the model's identities. As you were reading the first chapter, you may have noticed that, in order to create the model identities, we did *not* need to access either a data bank or Memory File. We simply specified the identities, and MODLER happily took them in, allowing this set of equations to be inserted directly into the MOD file; this circumstance implies that, like the CMF file, the MOD file does not contain any values for the model variables — it simply contains equation code. However, in order to solve a model, certain values are required, namely historical values for all the variables in the model and values of the exogenous variables during any (out-of-sample) solution period for the model. We therefore need to load these values into the one or more model solution files. In special cases, we may also need to load certain endogenous variable values, but this possible complication can be ignored for the moment.

Exactly how many historical values are needed can be determined by applying a set of rules spelled out in the next section, but it may be helpful to consider first the logic of this matter as background. One perspective on the difference between estimating a model and solving it can be attained by considering what is unknown in each case and what is given. When estimating the model, the historical values of the model variables are the givens. In a profound and ultimate sense, these values are actually not known: we only have estimated values, and the statistical agencies that produce them can revise these values month to month, but for parameter estimation these values are ordinarily *treated* as given and used to compute the parameter estimates. In this limited sense, the historical values are "known" or given. In contrast, when solving a model, we do not know the values of the endogenous variables; at best we know the (estimated or projected) values of the exogenous variables, any given parametric values of the identities, and the (estimated) values of the parameters for the behavioral equations. In the limited sense of being available, these are the given or "known" values during model solution; that is, to solve the model, these values must be provided.

It was mentioned earlier that, in general, econometric models are nonlinear in form. Because of this characteristic, MODLER is designed to solve a model using algorithms that are based upon the *assumption* that the model is nonlinear. This assumption has important consequences, the first of which is that any solution is approximate: *it is in the nature of a computer algorithm that solves a nonlinear system of equations that the solution will not be exact*. You may be aware that if A is a square, nonsingular matrix, and x and y are vectors, such that:

$$Ax=y$$

the solution is:

$$x = A^{-1}y$$

In fact, in the context of a computer, even this solution computationally involves approximation — because of the way in which the inverse of A must be computed using finite mathematics — but *in principle* such a linear solution is exact. In contrast, a nonlinear solution is neither in principle exact, nor is it necessarily unique; one might even go so far as to say that *in principle* it is non-unique [23] [24]. This non-uniqueness derives from the combination of facts that a nonlinear solution can *in principle* involve multiple roots, and computationally involves numerical approximation and even irreducible error [30].

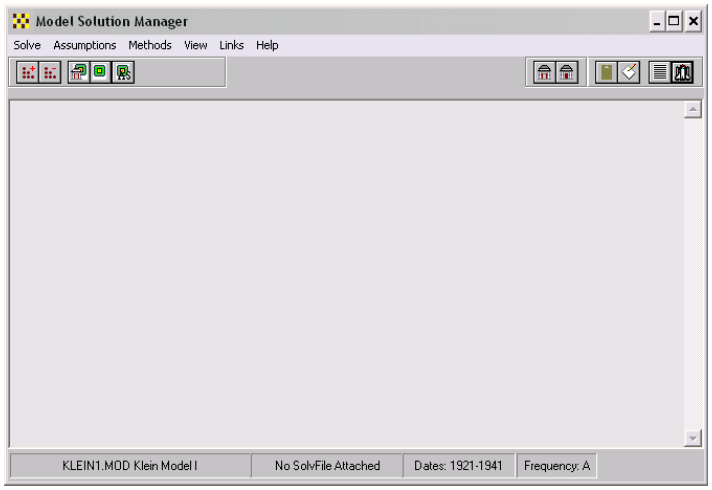
From a user's perspective, to fully comprehend the computational issues is not critical at present. However, the implications need to be taken into account. One implication is that, in order to solve a model, what is required each period of the solution is to start with a preliminary, or initial value for each endogenous variable value, in addition to contemporaneous values of the exogenous variables and the given values of the model equation parameters. This "initial" value, also known as a seed value, is usually the best value (in some sense) from a prior period, usually the period immediately prior to the time period of the solution. This fact establishes the requirement that, in order to solve a model using MODLER, you must ordinarily provide the values of all endogenous variables for the period of time immediately before the *first* solution period. Furthermore, if the model contains lagged values of variables, which Klein Model I does, then you must in addition provide values for all variables for as many prior (that is, historical) periods as *the maximum number of lags on any variable in the model*.

In addition to these computational requirements, it is generally true that you will be wise to provide historical data that can be used in the presentation of the model results. How many historical observation periods is your choice, but at least 3-4 years is not a bad recommendation, even in the absence of lagged variable values. Thus whether you regard the need for historical values as a computational requirement, or simply as presentational in origin, each solution file must contain a certain amount of history for all model variable values. The upper limit is normally not a binding constraint, for a MODLER solution file can contain a maximum of as many as 400 observations on each model variable.

Creating a Solution File: Practical Matters

In order to create a new solution file for a MODLER model, you must first attach a compiled model and open one or more data banks or a Memory File that contain values on each of the variables in the model. If you just estimated the model, compiled it, and are now about to create a solution file, these conditions will necessarily be fulfilled. However, if you are starting fresh with this Chapter, they may not be. There is no particular order in which these operations need to be performed, but for present purposes it might be best to begin by opening the KLEINBNK data bank, setting the observation frequency to annual, and the current date range from 1921 to 1941. All this can be done from the Central Control screen, which is MODLER's opening screen, last shown in Figure 21: in order to illustrate the compile step, this screen is displayed in that figure so that the dropdown **Model** menu is exhibited, with a bank open and the frequency and dates set appropriately.

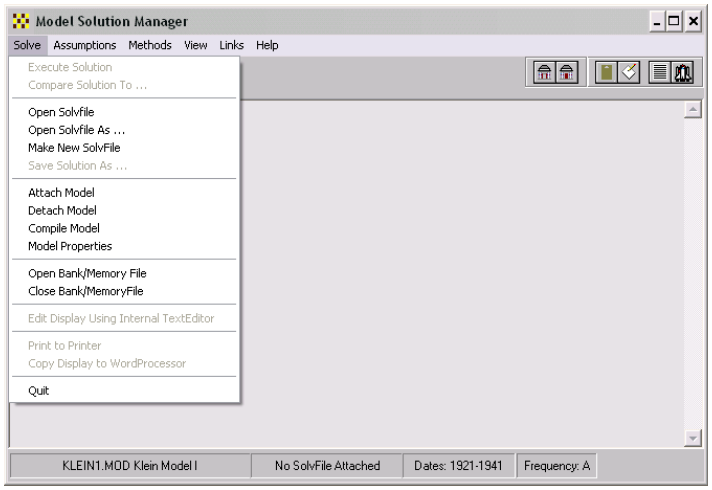
Referring back to this figure, notice the dropdown menu item **Solution Setup Operations** under the **Model Solution Manager** Option. Inasmuch as we created Klein Model I previously and compiled it, it is for present purposes the existing model, but observe also that the options **Solve Model** and **Display Solution Results** are both disabled, effectively implying the absence as yet of any solution file. If you then click on **Solution Setup Operations**, you should next see the screen that is displayed in Figure 23, which is the Model Solution Manager screen. The status bar for this screen indicates that Klein Model I is attached, but that a solution file is not. The dates and frequency status bar items also refer to the model and, once attached, to its solution file as well. At this stage, the databank, KLEINBNK, should be open for access, inasmuch as the process of creating a solution file specifically involves copying observation values on each of the model variables from this data bank to the solution file. If the data bank is not open, open it from the Model Solution Manager screen, either by clicking on the icon just below the **Links** and **Help** menu items, or by clicking on **Solve** and then the element **Open Bank/Memory File** of the dropdown menu that then appears.



KLEIN1.MOD Klein Model I | No SolvFile Attached | Dates: 1921-1941 | Frequency: A

Figure 23. Model Solution Manager Screen

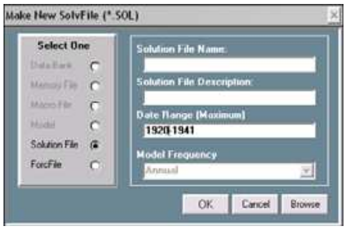
To actually make the new solution file, click on the menu entry **Solve**, which should cause the dropdown menu shown in Figure 24 to appear. Notice in particular the item **Make New Solvfile**. There is also an icon for this operation on the screen.



KLEIN1.MOD Klein Model I | No SolvFile Attached | Dates: 1921-1941 | Frequency: A

Figure 24. Making a New Solution File

When you click on **Make New Solvfile**, the form shown in Figure 25 should then appear. This form obviously can be used to specify the name of the solution file, give it a description, and set the extensive dates. Take care specifying the solution file name, for it is *both* the name by which the file will be known internally by MODLER and the name by which it will be known by the operating system. The name must begin with an alphabetic character, be from one to eight characters in length, and normally should contain only letters and numbers. It cannot have any embedded blank characters. MODLER will automatically give it the extent SOL, which is the required extent. The description is optional. Notice also that here the frequency textbox is grayed out, reflecting that the existing model mandates the observation frequency of the solution file data. Thus the only remaining choice is the date range.



Make New SolvFile (*.SOL)

Select One

Data Bank ○
Memory File ○
Macro File ○
Model ○
Solution File ●
ForcFile ○

Solution File Name:

Solution File Description:

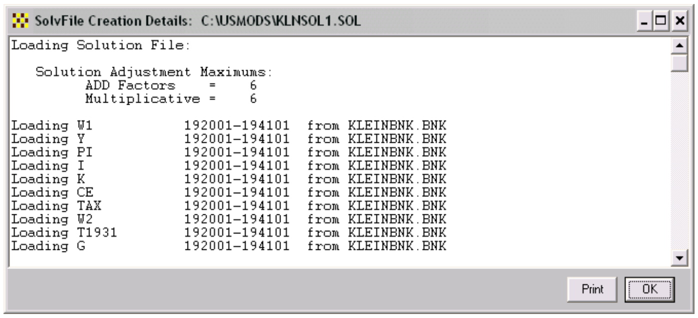
Date Range (Maximum): 1920-1941

Model Frequency: Annual

Figure 25. Solution File Definition Form

The Date Range must be set in conformity with the requirements discussed earlier. Since the longest lag of any variable in the model is a single period, there must be at least one year of history prior to the first period of any solution. However, you can choose to include more history. In particular, it does not matter if the initial date is left as 1920. The maximum number of observations per solution file variable is 400, so that it is unlikely that we will exceed that limit, even if the initial date is set to 1920.

For the choices available in Figure 25, choose the solution file name KLNSOL1, choose the date range as 1920-1941, and then click **OK.** The next screen you should see is that displayed in Figure 26. This display indicates the successful creation of a new solution file. Observe that you are told which model variables have been loaded and the implied quantity of data loaded: notice in particular that for each model variable, the date range of the observations *loaded* is shown. Because MODLER knows the name of every variable in the model, it responds by loading, if it can, data on each model variable, and this display evidently tells you variable-by-variable which observations have been loaded for what series. MODLER searches all currently open data banks, and any open Memory File, for these observations. Notice as well that the source of the data loaded for each variable is also indicated. At present, only a single data bank is open, but in other instances two or more data sources might be.



```
Loading Solution File:

   Solution Adjustment Maximums:
         ADD Factors    =    6
         Multiplicative =    6

Loading W1           192001-194101  from KLEINBNK.BNK
Loading Y            192001-194101  from KLEINBNK.BNK
Loading PI           192001-194101  from KLEINBNK.BNK
Loading I            192001-194101  from KLEINBNK.BNK
Loading K            192001-194101  from KLEINBNK.BNK
Loading CE           192001-194101  from KLEINBNK.BNK
Loading TAX          192001-194101  from KLEINBNK.BNK
Loading W2           192001-194101  from KLEINBNK.BNK
Loading T1931        192001-194101  from KLEINBNK.BNK
Loading G            192001-194101  from KLEINBNK.BNK
```

Figure 26. Solution File Creation Details Screen

The detail shown in Figure 26 is important. When you create a solution file you will need to confirm series by series that the right quantity of data has been loaded. The question is, what can you *actually* tell from this screen? Ostensibly you can tell that data has been retrieved for each series from KLEINBNK.BNK for the period from 1920 to 1941, and in the case of Klein Model I that is probably the correct inference. In contrast, if, for example, MODLER had been unable to find, in the open data bank (or in any open data bank or Memory File), *any* observations for one or more of the model variables, you would see the message "Undefined Variable(s) – Cannot Proceed with Solution." Such an event can occur if you happen to misspell a variable name in one of the identities, or if you were to open the wrong data bank – specifically one that did not contain the variable names used in the behavioral equations. Insofar as the behavioral equations are autocoded, if the right bank is chosen, the names of bank variables and the variable names used in these equations will obviously match; assuming of course that no other data source was used during model estimation. However, as mentioned earlier, when equations are manually coded, which is normally the case with identities, MODLER does not verify these variable names against the names in an open bank or Memory File; consequently, it is at this create-solution-file stage that any errors made specifying the variable names in the identities might begin to appear.

Now, you need to pay particularly close attention to what is about to be said: an important characteristic of the Make Solvfile operation is that, within the date range set, MODLER attempts to load observations variable-by-variable and series-by-series. If, for example, for particular model variables, data bank (or, if relevant, Memory File) observations do not exist for some of the date range, MODLER will retrieve a restricted set consisting of only the existing data source observations. Consequently, *so long as the missing observations are at either end of the date range*, any observations missing for certain variables will be indicated. In this case, for instance, you might see for a variable the date range given as 1920-1925, which would tell you that the data source only contained 6 observations for that series, from 1920 to 1925. One way to correct this problem, should it occur, would be to add new observations to the data bank and then to attempt again to create the solution file following the steps just described.

If such a message were to appear in the present case, since only one data bank is open, the obvious implication would be that the bank did not contain any data from 1926 to 1941 for that series. However, if multiple banks and/or a Memory File were to be open, then the implication might be that the data have been retrieved from the wrong data source. As usual with MODLER, the default search order is first the Memory File and then the data banks, in the order in which they have been opened, so that when there is more than a single bank as a data source, short series can occur as a result of MODLER retrieving data from the "wrong" data source. The inference to be drawn is that, when creating a solution file, you need to be careful not to have extraneous data banks or other data sources open, particularly if series names are not unique to a particular data source. In any case, you should, as a matter of course, review the data retrieval log and make sure that the data have been retrieved from the correct source for the date range you expect.

A further potential evaluation problem is that missing *intervening* observations will *not* be indicated by the screen display shown in Figure 26. Missing (or NA values) within the date range are not evident from this screen, and in and of themselves, such missing values are *not necessarily* an indication that an error has been made. As will be discussed further later, there are a number of instances in which missing intervening observations are likely to occur as a normal case. Thus the answer to the question posed above (what can you actually tell?), is that the screen displayed in Figure 26 is only an *initial screening device*, confirming *only* that some data has been loaded for each of the model variables, or advising you to the contrary.

Fortunately, additional evaluative facilities are available, even if possibly somewhat redundant in the present case. In general, at this stage of solution file creation, you would be wise always to click the **OK** button of the screen shown in Figure 26, and then click on the **View** menu item of the Model Solution Manager screen. As shown in Figure 27, when you click this item, you will see a dropdown menu that includes the option **Standard Solution Table**, and the subordinate options **All Variables**, **Endogenous** and **Exogenous**.

You may recall earlier discussion of the fact that MODLER incorporates certain standard display facilities, which you can use without previously making any particular provision. The *Standard Solution Table* is one example. Furthermore, notwithstanding that this phrase might be interpreted as implying that this table can be displayed once a model is solved, which it can, this table can also be displayed once a solution file has been either created or attached. One of the reasons for this convention is to allow you to look at a newly created or attached solution file. In the simplest case, the variables will be displayed in the order of the model equations in the Compiled Model File (CMF); that is, in the *solution order* of the model. The values will be displayed for the date range set at the time the solution file is either created or attached, and all the values for the chosen variables will be displayed for that time period. As you become more familiar with MODLER, you will also discover that you can define more restrictively the specific set of variables displayed, using the VList (Variable List) option; but at the moment this complexity can be ignored. What is relevant now is simply that a mechanism does exist to allow you to view the data in the solution file.

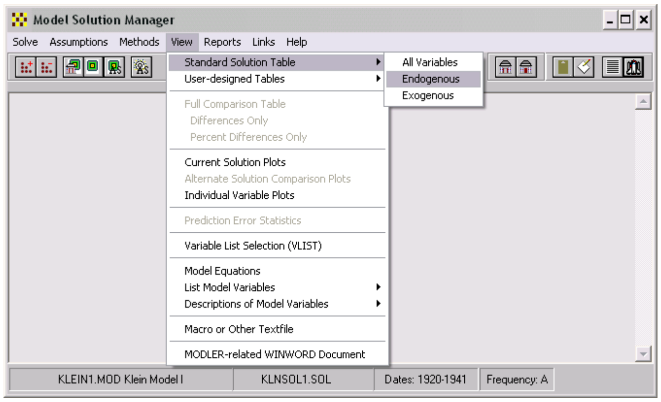
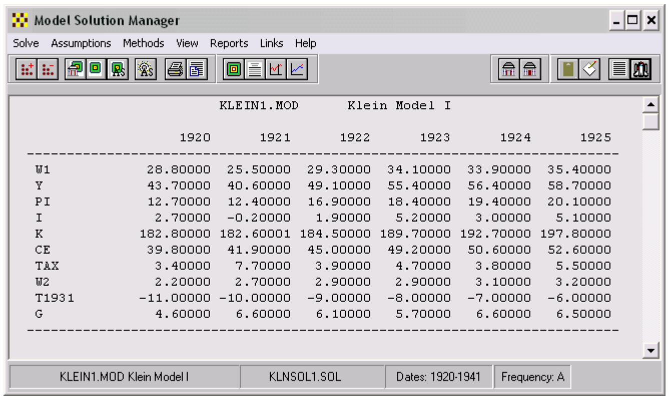


Figure 27. Model Solution Manager View Options

If you select **All Variables** from the options shown in Figure 27, you should immediately see the screen displayed in Figure 28.



```
                    KLEIN1.MOD      Klein Model I

                 1920       1921       1922       1923       1924       1925
----------------------------------------------------------------------------
 W1          28.80000   25.50000   29.30000   34.10000   33.90000   35.40000
 Y           43.70000   40.60000   49.10000   55.40000   56.40000   58.70000
 PI          12.70000   12.40000   16.90000   18.40000   19.40000   20.10000
 I            2.70000   -0.20000    1.90000    5.20000    3.00000    5.10000
 K          182.80000  182.60001  184.50000  189.70000  192.70000  197.80000
 CE          39.80000   41.90000   45.00000   49.20000   50.60000   52.60000
 TAX          3.40000    7.70000    3.90000    4.70000    3.80000    5.50000
 W2           2.20000    2.70000    2.90000    2.90000    3.10000    3.20000
 T1931      -11.00000  -10.00000   -9.00000   -8.00000   -7.00000   -6.00000
 G            4.60000    6.60000    6.10000    5.70000    6.60000    6.50000
----------------------------------------------------------------------------
```

Figure 28. First Display of Model Solution Data

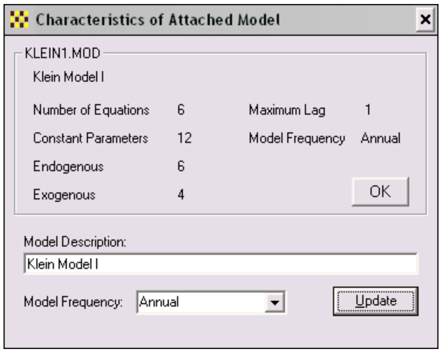
This table is scrollable, and by scrolling up and down you will be able to confirm that the solution file is fully populated with data. As just indicated, the table can be displayed at any time a solution file is attached, before or after a solution has been made. It shows the values of the solution file variables at that instant. Any missing values will be marked "NA" and will therefore be immediately obvious to the eye as you scroll through this table. Should you wish to view just the endogenous variable values or just the exogenous, then instead select the options **Endogenous** or **Exogenous** respectively. The ability to display solution data instantly, without going to the trouble of setting up intricate templates, is very helpful, particularly whenever you create a new model. For one thing, there is no assurance that you will keep the model in that form beyond the first solution attempt, so that you do not want to do a lot of extra work now, setting up some fancy display. Admittedly, the table is not beautiful, but it displays your model solution data in a way that you should have little trouble understanding what it says.

You should also note that an implication of the solution file creation procedure is that *whatever values* are in the data bank (or, more generally, your data sources) for the date range chosen will be copied to the solution file. For instance, should you have chosen to load "future" exogenous variable values from 1942 to 1950, and set the date range as 1920-1950, the result would be a solution file containing endogenous variable values from 1920 to 1941 and exogenous variable values for 1920 to 1950. Should you then wish to solve the model from 1942 to 1950 that could be done immediately. So long as observations are present for the period 1942 to 1950 for the exogenous variables, MODLER will quite happily attempt to solve the model for this period. A consequence of this convention is that the process of creating the solution file can serve *both* to load historical model variable values *and* the data needed to make an actual forecast with the model. There are also a number of other ways to specify the exogenous variable values, which will begin to be considered in the next chapter, but it is sometimes useful to load all the exogenous variable values at time you initially create a solution file.

Incidentally, if you look back at Figure 24, you will see in the dropdown menu a particular item, **Model Properties**. Click on this menu element and you will be presented with the form shown in Figure 29. Observe that this form states the Maximum Lag for the model, among other things. Earlier, we saw that the Klein model involved only a single lag on any variable, so that this form was not of particular interest when we created the first solution file. However, when you create other models, you may not recall instantly what its Maximum Lag is at the time you are about to create a solution file. Hence the relevance of this form in the more general case.

As an additional feature, this form also provides you with the means to change the description of your model, and even its observation frequency. Change the model description to your heart's content. However, be careful if you change the model frequency; there may be unintended consequences. In general, MODLER is designed to allow you to do a variety of things, some of which you should do only very carefully.

Changing the model frequency is one of those. The fundamental design assumption behind MODLER is that you know what you would like to do and that therefore the purpose of the software is not to set roadblocks in your way. But the assumption is also that you will understand all the implications of what you do.


Characteristics of Attached Model

KLEIN1.MOD

Klein Model I

Number of Equations 6

Constant Parameters 12

Endogenous 6

Exogenous 4

Maximum Lag 1

Model Frequency Annual

Model Description: Klein Model I

Model Frequency: Annual

Figure 29. Model Properties

Solution File Bookkeeping

Given the capabilities of the modern personal computer, including its operating speed, one way to proceed as you make various alternative forecasts and other model solutions is to create each solution file that you need just as we created the first one, using the **Make New Solvfile** option. However, an alternative is to create a base solution file and then to use this repeatedly to create successive solution files in order to then make a variety of alternative solutions. As a software use issue, it really does not matter which choice you make, but in the next chapter it happens that we will use a base Solvfile, partially because this will slightly simplify the presentation there. An additional, more general operational benefit of proceeding in this way, whenever you wish to compare alternative solutions to see the effect of doing one thing rather than another, is that by doing this it also may be easier to insure that the *only* differences between one solution and another are due to specific assumptions that you have made. If you create a new solution file each time, particularly over a number of different MODLER sessions that may span two or more working days, the data bank you use may change, or other things may happen to affect your results. In contrast, once you have created a base solution file, and if you use this file to create other solution files, your solution data set is frozen against all changes in your original data sources.

Whenever you create a new solution file, and have checked to see that it contains the observations you expect and has the characteristics you want, you can make a backup copy of that file, under a different name, simply by clicking on **Solve** and then **Save Solution As…** as illustrated in Figure 30. Once you have clicked on both of these menu elements you will be asked to provide a name for the saved solution file, which in its contents will be an exact copy of the currently attached file. Obviously, this operation is quite general and can be used whenever you wish to make a copy of the current solution file. But at the present time if we name the saved file KLN1BASE.SOL, the effect will be to create a base solution file that can be used to create a succession of other Solvfiles later.

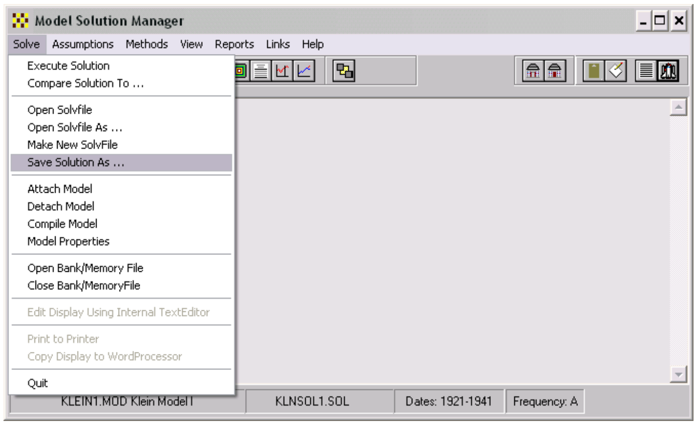
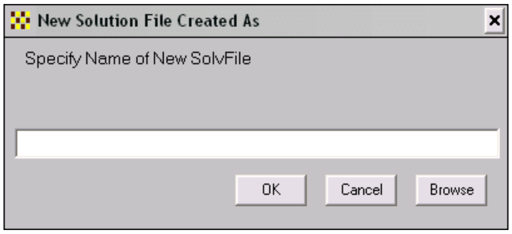


Figure 30. Making a Backup or Base Solution File

Actually, it is possible to go both ways. The reverse operation, creating a solution file to use in solving a model that is a copy of a solution file previously created, is invoked by clicking on the option **Open Solvfile As …** also displayed in figure 30. If you select this option, the next thing you will see is a form entitled *Select Existing Solution File*, which will list all the solution files in the Model Files directory of your hard disk, or other such storage area. Assuming that you created KLN1BASE.SOL previously, if you then select this file and click **OK**, you should next see the form shown in Figure 31, entitled *New Solution File Created As*. Provide a name, and the effect will be that MODLER will make an exact copy of KLN1BASE.SOL and give it the name you just specified, simultaneously attaching it as your current solution file. The consequence can be to overwrite any existing file with the name you specify, so be careful. This operation will leave unaffected KLN1BASE.SOL or other original file, unless of course you specify the same filename in each case, which is *not* recommended. Once again, what you should do and what you can do are not necessarily the same.



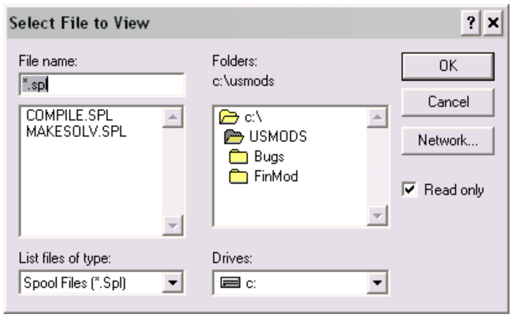
New Solution File Created As

Specify Name of New SolvFile

Figure 31. Open Solvfile As option

Additional Information

During the process of compiling the model and making the solution file, you might have noticed that a small yellow icon suddenly appeared on the toolbar of the Model Solution Manager screen. For purposes of identification, it can be seen in Figure 30 immediately below and to the right of the menu item **Help**, as two boxes each partially superimposed on a larger yellow box, or at least yellow is the color that will appear on your screen; it obviously will not be yellow in the context of this document if it has been printed in black and white. However, in any case, if you click on this screen icon at this stage, assuming that you have been working through this document using your computer, you will see a small file selector dialog box, as shown below in Figure 32. On your computer screen, the files shown in this figure should appear, but you may see the filename ESTIMATE.SPL in addition.



Select File to View

File name: *.spl

COMPILE.SPL
MAKESOLV.SPL

Folders: c:\usmods

c:\
USMODS
Bugs
FinMod

List files of type: Spool Files (*.Spl)

Drives: c:

☑ Read only

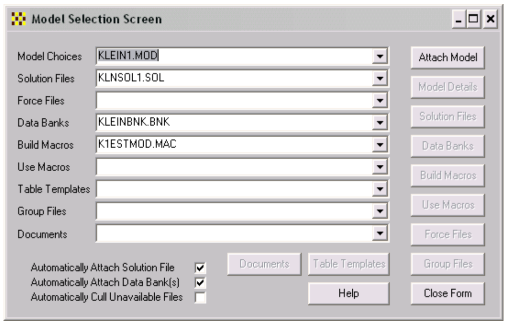
Figure 32. SpoolFile Display

Try selecting each of these files in turn. What you will discover is that COMPILE.SPL is a file that displays the compilation notification we previously saw in Figure 20. In turn, MAKESOLV.SPL displays the solution file creation details previously displayed in Figure 24. If you estimated the behavioral equations of the model using the macro file that is distributed with this document, ESTIMATE.SPL will also appear. It displays the set of estimated equations of the model. These files are generically called "spoolfiles" and they are automatically created by MODLER.

Spoolfiles are commonly created during each MODLER session and they are essentially temporary files, destroyed at the end of each session. Consequently, if you end a MODLER session and then subsequently begin a new session, spoolfiles created in the previous session will not be displayed. However, you can preserve them from session to session by renaming them, *before the end of a session*, so that under the new name they no longer have the extent SPL. MODLER only deletes files it recognizes as temporary, namely those with SPL and TMP extents, or having other special names.

Documenting a Model: The DEF File

You will have noticed as we have proceeded that we have progressively created files at each stage of the model building process. You might have wondered, in passing, how to keep track of them, especially if you ignored the earlier discussion of the model DEF file. If you go to the *Central Control Screen*, shown last in Figure 21, click on the **Model** menu item and then click on the first element, **Select Model**, a form quite similar to that shown below in Figure 33 should immediately appear. With some help from you, this form will manage the set of files that are associated with each model you build. Furthermore, not only does it organize them, but it will also assist you whenever you wish either to back up these files or copy them for export or to archive them.



Model Choices KLEIN1.MOD
Solution Files KLNSOL1.SOL
Force Files
Data Banks KLEINBNK.BNK
Build Macros K1ESTMOD.MAC
Use Macros
Table Templates
Group Files
Documents

Automatically Attach Solution File ☑
Automatically Attach Data Bank(s) ☑
Automatically Cull Unavailable Files ☐

Figure 33. Model Selection Screen

Notice first that the topmost text box of this form is entitled *Model Choices* and that the topmost button is labeled **Attach Model**. The items you will find in the *Model*

Choices drop-down text box, if you should click on it, are strictly determined by the set of MOD and CMF files that at that moment happen to be located in the *Model Files* directory, so specified in your directory settings as being where models are stored. You can confirm this directory choice, among all the directory settings currently in force, by going to the **Settings** menu item of the *Central Control Screen* and then clicking on **Directories**; for further details about these settings and their implications see Chapter I of the *MODLER User Guide* or else the *MODLER Getting Started Guide*.

Initially, when this form in Figure 33 appears, the model choice displayed will be the first model alphabetically by name. It is possible that, at the moment, you will have only a single model, but if you have more you can choose a particular model by selecting it from among those in the *Model Choices* dropdown text box. Whenever a particular model is selected in this way, the entries in the other list boxes on the Model Selection Screen will change conformably so as to catalogue the individual solution files, banks, macro files, document files, etc. that are associated with that model. However, at this point, this listing operation is a "read only" process: unless it was attached previously, the model identified in the *Model Choices* text box is *not* attached to MODLER. Its name and the files related to it are simply being listed for your information.

What permits this listing of both models and their related files is the existence of a DEF file (Definitions file) for each model, which was briefly described earlier. This is a file that is created automatically by MODLER when you first define the model. It is located in the same directory as the MOD and CMF files for that model. Furthermore, the filename is the same, so that a model's DEF file is easily identified. This file ideally contains a sectional listing of all the files associated with the individual model, each documentation section corresponding to one of the textboxes on the Model Selection screen. Should you be interested in the technical details, this file is formatted in the same way as any Windows INI file.

Incidentally, the word "ideally" is used in the last paragraph because of the fact that not all model-related files will necessarily be listed. Some of the sectional listings are created automatically as you develop each model; other listings must be created and maintained by you, inasmuch as you will be the final authority on what set of files actually comprise *all* those associated with each of your models. MODLER has only a limited ability to identify these: the MOD and the CMF files share a common name, and if you have created solution files using the same computer, MODLER obviously will be able to capture information about these, as well as the names of any data banks used in their creation. However, other files, such as the MAC files, are not immediately linked to a model. Furthermore, if you simply copy model-related files to the hard disk of a specific machine from some other machine, the program has no way of knowing that these are related to a specific model. Consequently, although MODLER attempts to make the requisite linkages to the best of its ability, at the end of the day you are responsible for identifying which files are associated with a given model, using facilities that will be described later.

For the moment, let us assume that each model is fully documented. In order to attach a particular model, once it has been selected and its related files are displayed, you must press the **Attach Model** button. In response, two things will happen: first, the model will be attached to MODLER. Second, the remaining buttons on the *Model Selection Screen* will become enabled. However, before you press the **Attach Model** button, look first at the bottom left-hand-corner of the Model Selection Screen and observe the check boxes there. Notice that these imply that you are able not only to attach the model, but simultaneously, at your option, also whichever solution file is currently shown in the text box labeled *Solution Files*, plus all the data banks that are listed in the drop down text box entitled *Data Banks*. Only one solution file can be attached at a time, but up to 15 banks can be simultaneously open for access. Be careful here: if you open banks at this point, it will usually be best to have previously closed all banks prior to displaying the *Model Selection Screen*: if other, unrelated banks are already open and contain data series with the same name as any used in the model, there is a risk that these other data series could be retrieved, instead of those in the model-related data banks.

Notice also the check box that, by default, is *not* checked; it is labelled “Automatically Cull Unavailable Files.” If this box is checked, whenever a model is attached using the *Model Selection Screen*, and by pressing the **Attach Model** button, MODLER will quickly review all the model-related files that are listed in the pertinent DEF file and will *delete from this file any file names for which an associated file cannot be found in whichever directory/folder it ostensibly should appear*. If a filename appears on its own, without a path specification, MODLER will infer its directory location on the basis of the default directory settings. If a filename appears complete with path, then the location of that file will be understood to be that explicitly specified by this path.

As a beneficial characteristic, what is implied by this culling convention is that if you delete an originally relevant file from one of the MODLER directories, it will automatically be culled from the model’s DEF file the next time you use the Model Selection Screen to attach that model; for example, you might independently consolidate several transformation macro files into a single file, deleting the extra files from the directory. These surplus files will then automatically be culled from the model’s DEF file the next time you attach the model. Of course, on the negative side, if you have moved a file to some other directory and MODLER cannot find it when attaching a related model, the program will cull its name from the DEF file for that model if this check box is checked. This culling will also occur if the *directory* cannot be found, as may happen if the drive originally used was removable and has since been removed. However, in this case, as a safety measure, the culling will take place in two steps: initially, an error message will appear, indicating that the directory is unknown, at which point the *directory* portion of the path will be deleted, leaving only the filename. The *next time* the model is attached, this file name will then be culled if you have not in the meantime put a file of that name in the relevant default directory for that file type.

An obvious question at this point is, how do you add related model files, in order to create and maintain the other listings? It was mentioned earlier that whenever a model is attached, using the **Attach Model** button, all the grayed-out buttons on the *Model Selection Screen* become enabled. Once these are enabled, you can use them to add files to the model DEF file for a particular (attached) model. With the exception of the **Help**, **Close Form**, **Attach Model** and **Model Details** buttons, pressing the buttons individually will in each case cause an appropriate file selector form to appear. This form will be keyed to the file type and will display the relevant files located in the relevant directory, thus allowing you to select a pertinent file. Alternatively, if you press the **Browse** button on that form, you will be able to search for additional files of that type in any directory and on any accessible drive. Files that are found in the default directory for files of that type will be displayed without any drive or directory information. However, if you browse for the filename, then the complete path plus filename will be shown, thus indicating that the file is not located in a default directory.

Should you later move the file to another directory, you may need to add it again to the DEF file, for if you actually *move* it, it cannot then subsequently be found in its original directory. In this case, the fully pathed file name may be culled, inasmuch as (with the exception of the MOD, DEF, and CMF files) there is no automatic general mechanism to associate a file name with a model. The exception to this rule, as mentioned earlier, is if you move the file to the specified directory for that filetype, in which case, once its original path details are removed, it will be found in the directory to which you have moved it whenever the model is next attached.

An additional, related facility is the capability to archive or export a model, options that can each be selected by first clicking the **Model** menu item of the *Central Control Screen*. Then choose either **Archive Model** or **Export Model** from the listed elements. Actually, the only significant difference between these alternatives is conceptual, for "export" in this context refers to copying model and related files for use at another MODLER installation, whereas "archiving" refers to backing up your files locally. Obviously, the difference fundamentally consists of the possible later use of these files in either the same location or at another location, although in practice different storage locations and even different types of storage media may be involved in the archiving versus export process. In contrast, exporting a model and related data for use in a foreign, non-MODLER software context will generally involve instead, or as well, the creation of a text representation of the model and various types of data exchange files, including perhaps ASCII or spreadsheet files.

Generally, when archiving or exporting a model in a MODLER-only context, there are two important rules that must be obeyed. First, a model cannot be archived or exported to a destination directory that already contains a model: you must first delete all model-related files from this directory and its subdirectories. In any case, attempting to copy a model overtop another model (or even another copy of itself) is not a good idea inasmuch as file version conflicts and other such problems can easily occur. Second, you should make sure that any files you are attempting to archive or export are not at that time being used by either MODLER or another program. In particular, you should close all banks and detach all model files before you attempt to archive or export a model.

Notice that, for archive or export, MODLER simply copies a model and its files, as determined by its DEF file. It does not attempt to "zip" or otherwise compress or locate these files in any type of library file. However, once the model and its related files are all copied to a particular directory, then it is possible to use WinZip, Pkzip or other file compression or "library" software to create some type of single file containing all these files, possibly in a compressed format. Obviously, before you take the step of compressing these files, or otherwise containing them, you should inspect the copied files to make sure that all the relevant files have been copied.

Chapter Appendix: Forming an Objective Function

As discussed earlier in this chapter, Klein Model I involves such constructive data relationships as:

$$PI + W + TAX = CE + I + G$$

Which means that the model involves redundant or possibly "hidden" identity equations. In a modeling context, these are actually functionally the same thing as the familiar *ex post* market clearing equality of Demand and Supply:

$$Q_D = Q_S$$

Not at present, but in the context of other models, out of equilibrium behavior may usefully be incorporated into a model. MODLER permits you to do this in a variety of ways, including the use of relational operators in the equations of a model. For example, if QD is Quantity Demanded and QS Quantity Supplied, then the logical relations:

```
(QD.GT.QS)
(QD.LT.QS)
(QD.EQ.QS)
```

will each evaluate in the context of a MODLER expression (or model equation) to the value one or zero, depending as the specific expression is true or false. It is therefore relatively straightforward to create expressions like:

```
X = (QD.GT.QS)*Z1  +  (QD.LT.QS)*Z2
```

which have the property that X = Z1, X = Z2, or X = 0, depending on the relative values of QD and QS.

Incidentally, you do need to be careful with the value 0 (zero) as the result of a *mathematical* expression (as opposed to logical or relational), for very seldom will two variables in a computer take exactly equal values in a meaningful way. There is also the problem that *each period* the solution is obtained iteratively, so that if the implied speed of response is less than infinite, and in particular if it takes more than a single period, special care will need to be taken to avoid within period, iterative solution problems that arise because a value is generated (temporarily) during a particular iteration that is not an “equilibrium” value, yet could be evaluated as if it were.

The ability to use logical and relational expressions to model out-of-equilibrium behavior, or to model *ex ante* and *ex post* distinctions is powerful, but as this is an almost unexplored aspect of possible model behavior you need to proceed carefully.

Chapter 3 Solving the Model: the Basics

A model such as Klein Model I can be regarded as a representation of an economic process, or tentatively it can even be considered to represent the structure and at least certain of the characteristics of a macroeconomy. Given both the dated quality of this model and its obvious limitedness as a representation, it is of course important not to get too carried away making such claims. Nevertheless, such qualifications aside, the point is that when the process of solving a model is considered, the difference between Klein Model I and some other model that objectively might be considered more adequate as an economic representation is a matter of degree, rather than kind. Furthermore, the very simplicity of this model will allow us to apply it as a prototype to explore the process of using a model without at the same time becoming entangled by the inherent complexities of a more detailed and realistic model.

The use of any model essentially implies solving it and there are several different classifications of model solutions. Using the model to *forecast* or *predict*, according to the conventional meaning of these terms, generally involves making a statement about the "future," which under ordinary circumstances can be understood to refer to the period of time subsequent to that for which the model was estimated. This logic is buttressed by the fact that, normally, a model will be estimated using all the recent historical data currently available, so that only the "true" future (or irrelevant past) remains. Klein Model I is, however, obviously a special case: its parameters are estimated for the period 1921-1941. Therefore, this model can today, by this definition of the future as post-estimation sample, be used to forecast the period, say, from 1942 to 1950, an idea of the future that possibly sounds strange from the vantage point of the twenty-first century. Consequently, we will obviously need to clarify further what might be meant by a forecast or a prediction.

A similar, but in certain respects more general idea is that the model can also be solved in order to *simulate* the behavior of the economy – that is, as a representation of an economy, the model can be made to generate a set of variable values that the economy possibly would or might, given a pertinent alternative reality. Such a simulation can take any of several forms, but the simulation process does not necessarily need to refer to a particular chronological time period. For example, one possibility is to use the model to consider how it behaves, relative to the observed historical behavior of the actual economy, during all or part of the estimation sample period. Such *in-sample* simulation is unambiguous for this model, but it is clearly also possible to use the model to simulate the effect of particular shocks to the economic system or the effect of particular economic policies, either during the 1920-1941 time period, or some other given time period, or even some period of time that has no actual relation to chronological time [12].

The justification for talking about the model in these (possibly lofty) terms is founded upon the recognition that it is essentially a set of simultaneous equations incorporating parameters that have been estimated using historical economic data relating to the US economy during the time period from 1920 to 1940. Just how much this estimation of the parameters thereby imbues the model with actual characteristics of the economy may be debatable, but there is clearly some relation. And one of the reasons to go to the trouble of exploring the behavior of the model in various in-sample and out-of-sample contexts might be in order to consider how to go beyond Klein Model I in order to create a modern successor that it is easier to justify as an adequate representation of an actual economy for whatever purpose that model might be used. It is in fact this last consideration that makes solving Klein Model I interesting, other than as a purely pedagogical example.

Of course, the idea of using a representation, or model, of some existing, or even possible thing, is not unique to an economy. For example, ship and aircraft models are commonly created to consider aspects of the design of aircraft and ships before they are built. Alternatively, after a disaster, models are sometimes created in order to explore what went wrong. In each of these cases, there is an obvious distinction between the object represented and the model, with the model never intended to be an exact replica of or replacement for what has been modeled. Clearly, the issue is not exact replication per se, but rather a matter of creating a model that has at least some of the relevant characteristics of the object that is modeled – the aim when building a model is not to create a clone, but rather to develop a representation that is sufficiently adequate to exhibit particular behavioral characteristics. Judging that adequacy is part of the design problem.

We began earlier to consider the essential difference between estimation and solution. Parameter estimation involves using existing data on the model variables to estimate the unknown values of the behavioral equation parameters. In contrast, when the model is solved, the parameters of the model are given. As discussed earlier, the unknowns in this context are certainly the values of the endogenous variables, as generated by the model. But what about the exogenous variable values? These will certainly be unknown if a prediction is to be made for a time period after the present. That is, any solution made that refers to a future chronological time will necessarily involve the need to make some assumptions in advance about the values of the model exogenous variables that are to be used. Once these values are determined (or assumed) and are fed into the model appropriately, this set of equations will generate corresponding values for the endogenous variables.

As a matter of terminology, it is conventional to refer to a forecast or prediction made using assumed or otherwise projected values of the exogenous variables as an *ex* *ante* forecast or prediction [21]. Thus, any forecast made with the model for a period of time chronologically later than today, in the early part of the 21st century, will obviously be *ex ante*. In contrast, a prediction or forecast made using observed, historical values of the exogenous variables will ordinarily be called *ex post*. Usually, but not invariably, *ex post* will refer to historical model solutions made for periods of time outside the sample period, 1920-1941. Solutions made during the period 1920-1941 will normally be called *in sample* solutions, but clearly we may also need to distinguish between solutions made using the historical values of the exogenous variables and those made using other values, but intended to be used as observations pertaining to the time period 1920-1941. Often, in the latter case, such solutions will be called simulations.

Notice, incidentally, the use of the word *assumption* to refer to the exogenous variable values used to make a solution. The solution of a model based upon a particular set of exogenous variable values falls into the category of being a *conditional* prediction: the endogenous variable values are generated conditional upon the values specified for the exogenous variables, as well as the estimated parameters and other previously specified values. Of course, the exogenous variable values themselves may have been predicted, but if this is the case, these predictions are presumed to have been made *a priori* as a separate operation: they are assumptions in the sense of being predicate values, rather than being values generated by the model itself.

It is useful to have defined a few terms; for a more extensive discussion, see Klein's *Essay on the Theory of Econometric Prediction* [21].

The First Solution: An In-sample Simulation

We can now begin the practical demonstration by solving Klein Model I for the period of time from 1933 to 1941. For this exercise we will use the relevant historical values of the model's exogenous variables, the same values as were in part used to estimate the model parameters. As a matter of definition, this will be an *ex post in-sample simulation*, and it is almost self-evident that since the model's parameters already incorporate, by their estimation, information about the relationship between the endogenous and exogenous variables during this period, this type of solution of the model will not provide us with any substantive information about the model as an economic representation. Howrey and Kalejiaan [16] long ago presented results to this effect. Nevertheless, proceeding in this fashion does allow us to solve the model without the need to make assumptions about the exogenous variable values. Later, we can consider how to make and change such assumptions.

To make this forecast, you may wish to begin at the Model Solution Manager screen. In this case, if you are not there, you can get to it from the Central Control screen by selecting the menu item **Model** and then clicking on **Solution Setup Operations**, which opens the solution manager screen. Figure 34 illustrates the choices.

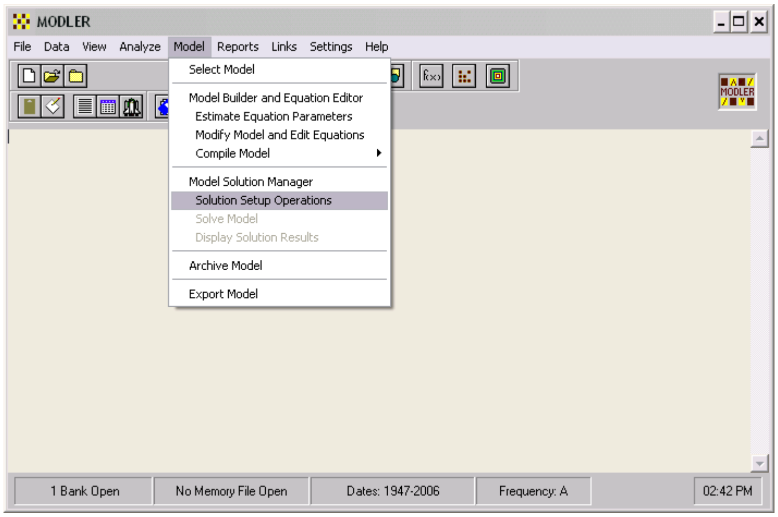
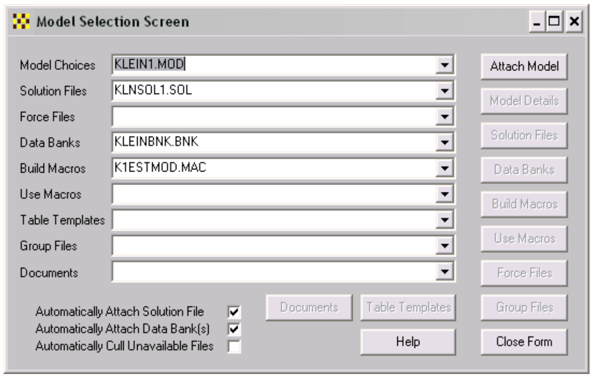


Figure 34. Model Solution Manager: the First Solution

Solution Setup Operations is highlighted in the figure, but before we consider the implications of this choice, notice also that **Solve Model**, just below it, is grayed out, which indicates that the Klein I model is not attached. There are actually two ways to attach this model for solution. As discussed at the end of the last chapter, one is to click on the topmost dropdown element **Select Model**, which opens the Model Selection Screen shown in Figure 35.


Model Selection Screen

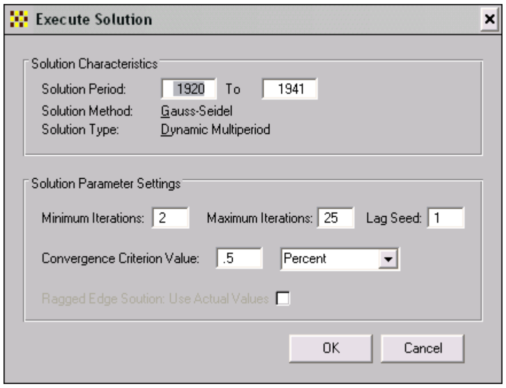
Model Choices	KLEIN1.MOD
Solution Files	KLNSOL1.SOL
Force Files	
Data Banks	KLEINBNK.BNK
Build Macros	K1ESTMOD.MAC
Use Macros	
Table Templates	
Group Files	
Documents	

Automatically Attach Solution File ☑
Automatically Attach Data Bank(s) ☑
Automatically Cull Unavailable Files ☐

Figure 35. Model Selection Screen

Considering this screen, notice in particular the check boxes in the lower left-hand corner. When you choose one of the models listed in the drop-down text box labeled *Model Choices*, and then press the **Attach Model** button, if the first two of these check boxes are checked, not only will you succeed in attaching the model, but you will also simultaneously attach whichever solution file the name of which is displayed in the *Solution Files* text box, as well as the data banks listed in the *Data Banks* drop-down text box. If one or both these check boxes are not checked, then the effect of attaching the model will be more limited. Further information about the various options can be obtained by pressing the **Help** button. At the moment the two checked boxes should be checked.

Once you have attached a model using the Model Selection screen, then closed the form, using the **Close Form** button at the bottom right of this form, **Solve Model** will be enabled. Click on this element and you will see the form shown in Figure 36. The effect is to permit you to solve the model immediately, essentially as a two-step operation: 1) Attach the model. 2) Solve it. If you have just read chapter 2 and have worked the examples, at this moment you are ready to solve the model.



Execute Solution

Solution Characteristics

Solution Period: 1920 To 1941

Solution Method: Gauss-Seidel

Solution Type: Dynamic Multiperiod

Solution Parameter Settings

Minimum Iterations: 2 Maximum Iterations: 25 Lag Seed: 1

Convergence Criterion Value: .5 Percent

Ragged Edge Soution: Use Actual Values □

Figure 36. Execute Solution Form

However, it may be helpful to approach the model solution process in a more deliberate way. If, instead, starting from Figure 34, you choose **Solution Setup Operations**, the Model Solution Manager Screen shown in Figure 37 will appear. This particular display shows this screen as it will look once you have selected the menu item **Solve**, and before any model has been attached, so that this choice has been made in lieu of attaching the model. Effectively, the screen looks as it will if you newly start MODLER and then choose **Solution Setup Operations**. At this point, your further choices are clearly rather limited, inasmuch as at this stage you really can only choose between attaching a model and opening a bank, the latter of which is not now a relevant choice. Therefore, if your screen looks like that shown in Figure 34, first attach the model using the dropdown menu choice **Attach Model**, and then select Klein1 from the list of files you will then be shown. Once you have attached the model, the option **Open Solvfile** will no longer be grayed out. Next, attach the Solution File, KlnSol1, by clicking on **Open Solvfile** and then selecting it. Obviously, we are now proceeding deliberately.

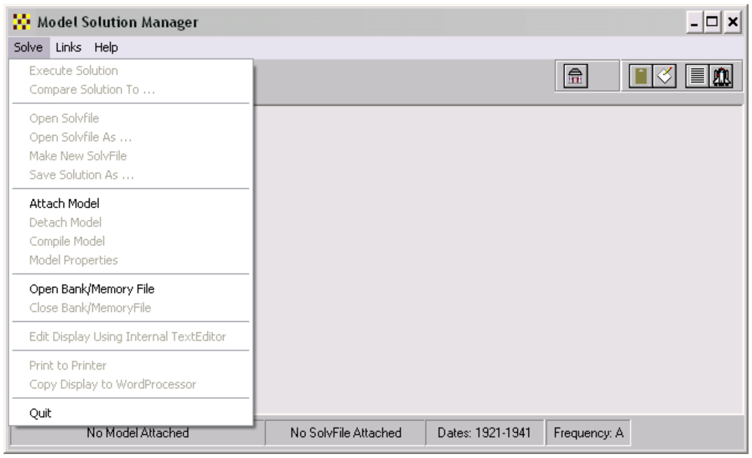
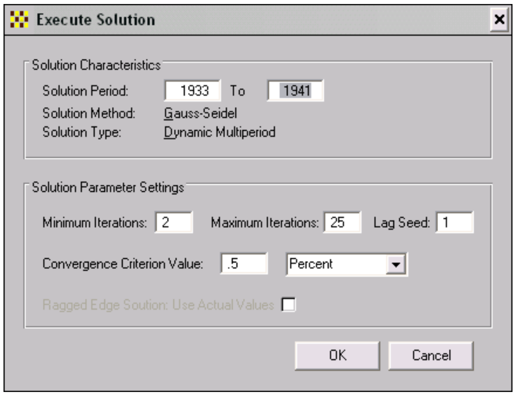


Figure 37. Model Solution Manager Screen

As you carry out these operations, consider the logic of your actions: first you chose a particular model, then you chose a particular solution file. The model consists of a particular representation of the characteristics of an economy. The solution file, because it contains the historical values of the model's variables, as well as particular assumed values of the exogenous variables specifically, represents a given set of circumstances. At this stage, we have taken the easy route of using as our assumed values the historical values of these variables, thus leaving to the side, for the moment, any consideration of how to go about making assumptions, or inserting particular assumptions in a given solution file. Ultimately, we will need to consider how to make and change the individual assumptions, but for the moment, that is a detail. Initially, what is important is the general concept. We are now in fact ready to make our first model solution, an historical simulation. Look once again at Figure 37 and notice that the topmost choice in the Solve dropdown menu is **Execute Solution**. Once the Solvfile has been attached, unlike in Figure 37, this choice will no longer be grayed out.

Therefore, click on **Execute Solution**. You should then see the form shown as Figure 38. This is of course the same form previously displayed in Figure 36, so that you can now see the two routes that can be taken to get to it. This form manages the model solution process and you should at this stage make one change only: change the first solution period date, 192101, to 1933, as shown, and press the Enter key. The various options on this form permit quite detailed control of the model solution process, ranging from the selection of the method of model solution (which now should be Dynamic Multiperiod) to control of the seed values and other solution parameter settings. If you would like to know to what each of these settings specifically refer, cancel out of this form, and click on the **Help** menu item of the Model Solution Manager screen, then choose **Model Solution Facilities Outline**. But if you are willing to go with the flow, instead click **OK**.

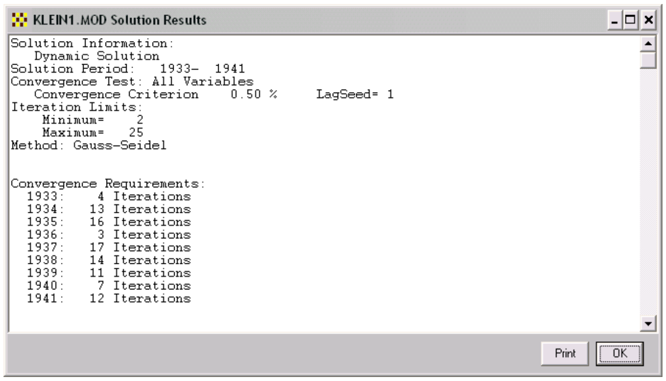


Solution Characteristics
Solution Period: 1933 To 1941
Solution Method: Gauss-Seidel
Solution Type: Dynamic Multiperiod

Solution Parameter Settings
Minimum Iterations: 2 Maximum Iterations: 25 Lag Seed: 1
Convergence Criterion Value: .5 Percent
Ragged Edge Soution: Use Actual Values □

Figure 38. Execute Solution Form

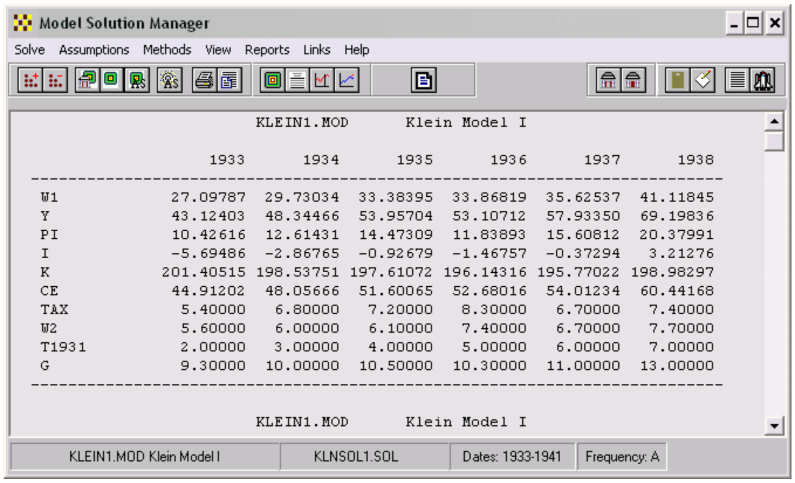
You should then see the display shown in Figure 39. You can print this display, if you wish, by clicking on **Print**. Alternatively, until you make another solution or leave MODLER, you can display it at any time by going to the Model Solution Manager screen, selecting **View** and then selecting **Solution Log**. This solution log specifies the characteristics of the last solution, ranging from the solution parameter settings to the number of iterations it takes each period to solve the model. The number of iterations is displayed because models are solved by a process of iteration. You should be aware of this fact, but for the moment this is a detail that you can otherwise ignore. We will consider the particular implications later.



```
Solution Information:
   Dynamic Solution
Solution Period:   1933-  1941
Convergence Test: All Variables
   Convergence Criterion    0.50 %     LagSeed= 1
Iteration Limits:
    Minimum=    2
    Maximum=   25
Method: Gauss-Seidel


Convergence Requirements:
  1933:    4 Iterations
  1934:   13 Iterations
  1935:   16 Iterations
  1936:    3 Iterations
  1937:   17 Iterations
  1938:   14 Iterations
  1939:   11 Iterations
  1940:    7 Iterations
  1941:   12 Iterations
```

Figure 39. Solution Log Display

The Model Solution Manger Screen will appear once more when you have clicked the **OK** button on the Solution Log Display. If, on this screen, you click on the menu item **View** and then choose **Standard Solution Table** and **All Variables** from the dropdown menu, the table shown in Figure 40 will appear on your screen. One of the characteristics of this solution that immediately stands out is that Net Investment is shown to be negative until 1938.

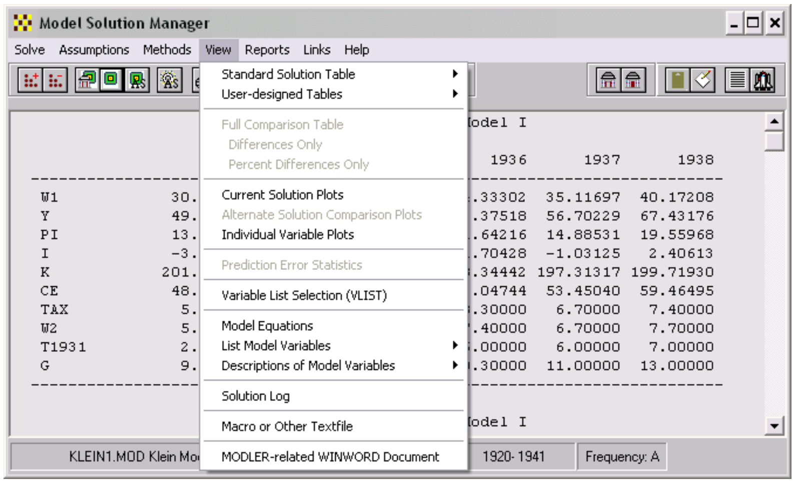


```
                         KLEIN1.MOD      Klein Model I

                    1933      1934      1935      1936      1937      1938
 ----------------------------------------------------------------------------
  W1            27.09787  29.73034  33.38395  33.86819  35.62537  41.11845
  Y             43.12403  48.34466  53.95704  53.10712  57.93350  69.19836
  PI            10.42616  12.61431  14.47309  11.83893  15.60812  20.37991
  I             -5.69486  -2.86765  -0.92679  -1.46757  -0.37294   3.21276
  K            201.40515 198.53751 197.61072 196.14316 195.77022 198.98297
  CE            44.91202  48.05666  51.60065  52.68016  54.01234  60.44168
  TAX            5.40000   6.80000   7.20000   8.30000   6.70000   7.40000
  W2             5.60000   6.00000   6.10000   7.40000   6.70000   7.70000
  T1931          2.00000   3.00000   4.00000   5.00000   6.00000   7.00000
  G              9.30000  10.00000  10.50000  10.30000  11.00000  13.00000
 ----------------------------------------------------------------------------


                         KLEIN1.MOD      Klein Model I
```

Figure 40. Standard Solution Table: All Variables

The table shown in Figure 40 is of course one of the standard solution displays. It is not a particularly attractive table and has no fancy formatting, but it has the virtue of being instantly produced once a solution is made. As you will already know, the model's variables are shown in a particular order, specifically the order that corresponds to the way in which MODLER organizes the model for solution; each of the variables is identified by its mnemonic name. Obviously, to be able to understand this table, you will need to know these names, so that this is certainly not the sort of display that you might wish to publish. Its purpose is simply to provide instant feedback to the person solving the model. As indicated earlier, the presumption is that anyone actively solving a model will know these names and will value the ability to look at a solution without first having to go to the trouble of creating a fancy table.

An alternative way to view a model solution is graphically. But, first, it may be helpful to consider the elements of the View dropdown menu, which is shown in Figure 41.



```
                                       odel I

                                          1936       1937       1938
  ----------------                     -----------------------------
   W1            30.                    .33302   35.11697   40.17208
   Y             49.                    .37518   56.70229   67.43176
   PI            13.                    .64216   14.88531   19.55968
   I             -3.                    .70428   -1.03125    2.40613
   K            201.                    .34442  197.31317  199.71930
   CE            48.                    .04744   53.45040   59.46495
   TAX            5.                    .30000    6.70000    7.40000
   W2             5.                    .40000    6.70000    7.70000
   T1931          2.                    .00000    6.00000    7.00000
   G              9.                    .30000   11.00000   13.00000
  ----------------                     -----------------------------

                                       odel I
```

Figure 41. Model Solution Manger Screen: View Menu

Notice first that certain elements are still grayed out, especially those that refer to comparisons between one solution and another, which we have not yet addressed. The options available include the **Standard Solution Table**, which we have just seen, and **Current Solution Plots** and **Individual Variable Plots**, which we are about to consider. In addition, ignoring, for the moment, **Variable List Selection**, we could look at the **Model Equations**, or **List Model Variables**, or even view **Descriptions of Model Variables**. And, of course, there is the **Solution Log**, should you wish to view this again. Finally, independently of any solution file, we can look at any **Macro or Other Textfile** on our hard disk or even display a **MODLER-related WinWord**

Document, the presumption being that Microsoft Word is the default word processor previously chosen. The significance of "MODLER-related" is that the document to be viewed is located in the directory specified in the directory settings as that to be used for MODLER-related documents.

If you click on the menu item **View**, then select **Current Solution Plots**, you will be able to display successively plots of each of the model variables for the period solution file period 1921-1941. Figure 42 displays one of these plots. Click on the **Next** button to display each plot in succession or, alternatively, press the <Enter> key.

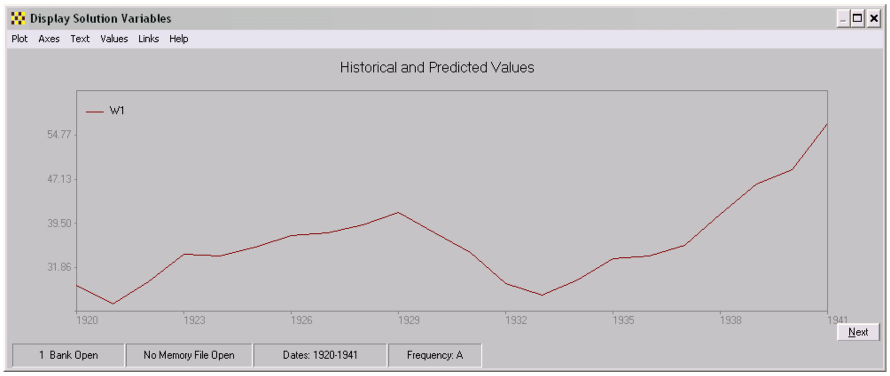


Figure 42. Solution Plot of Private Labor Income (Total Wages)

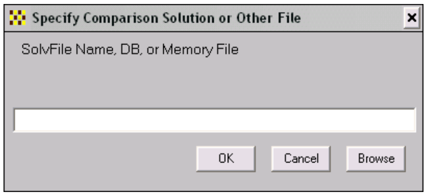
When you first see this plot, it will not look like Figure 42. In particular, it will be noticeably less wide, but you can reformat the original plot and spread it horizontally, by dragging at the edges of the plot dispay with your mouse as has been done here, with the effect that the impression of sharp upward movement during the period 1935 to 1940 is considerably diminished. When you see the plot displayed initially, you may get a distinctly different impression from its original aspect ratio of the speed of economic recovery during the 1930s. Plots can be illuminating, but they can also mislead the unwary, as this example demonstrates.

The First Simulation: A Comparison

Obviously, when considering this simulation, you need to take note of the fact that we began the simulation in 1933. You might now wish to make some more simulations, for instance successively starting them in 1925, 1928, and 1931, to see how the choice of initial solution period affects the overall simulation performance of the model during this general period of time. To make an alternative solution, first click on **Solve**, then select **Open Solvfile As…** and initially choose Kln1Base; you will then be asked for the name of your new Solvfile. Choose KlnSol2. This operation captures our base solvfile, created as described at the end of the last chapter, and uses it to create KlnSol2. The first simulation we performed is still in KlnSol1, and its values will be unaffected by the next simulation. Recall that by creating a succession of distinct solution files in this way, not only will you be able to make comparisons later between solutions, should you wish, but more importantly you also prevent one simulation from contaminating another, as could happen were you to use the same solvfile to make a succession of simulations.

If you make alternative simulations using the historical values of the exogenous variables, you will generally find that the behavior pattern is broadly similar, the reason being of course that in each case these values do not change from simulation to simulation. The exogenous variables include Government Purchases, as well as the public sector wage bill, and tax yield. Understandably, the historical values of these variables have an important effect, irrespective of the choice of initial solution period. However, the choice of initial period does affect the results. To see this explicitly, click on **Execute Solution**, under the **Solve** menu item of the Model Solution Manager screen. This time, set the initial solution period date shown on the Execute Solution form to 1927, and then click **OK**.

You can display this solution in the same way as the last, displaying a table and one or more graphs of the behavior of particular variables. However, it is more interesting to consider the two solutions that we have made relative to each other. Such comparisons can be made both in tabular form and graphically. There are in fact several different comparison tables that can be generated. However, to begin, we need to tell MODLER that it should permit such comparisons; this telling is done by selecting the **Solve** menu item of the Model Solution Form and then selecting the option **Compare Solution To**. When you click on this option, you should immediately see the form shown in Figure 43.



Specify Comparison Solution or Other File

SolvFile Name, DB, or Memory File

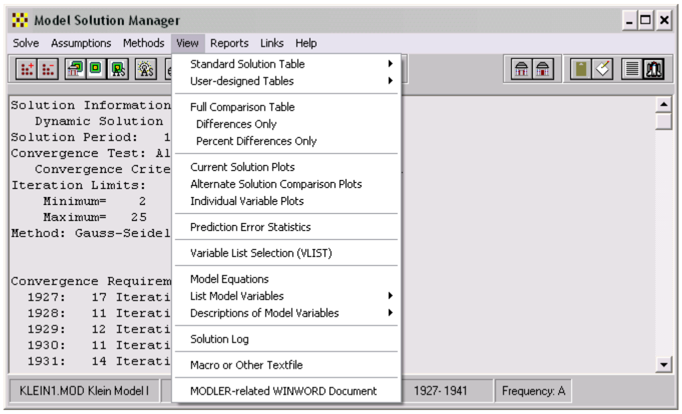
Figure 43. Compare Solution To

When you provide a filename for this form, evidently you can choose either another solution file, a data bank, or the Memory File. If you wish to compare the current solution to an alternative solution, simply enter the name of the solution file. If that file is in the model files directory, it will immediately be found. Otherwise, you will generally need to give the path as well as the filename. If you wish to compare the solution to data bank series values, then you must either give the complete data bank name, including the extent BNK, or precede the name with “DB:”. However, generally you will only need to include the path if the bank chosen is not in the data bank directory specified in your MODLER settings. If there were a Memory File open, then you would put the words Memory File, to compare the solution to data in the Memory File. In the general case, you can also put MB (multiple banks and Memory File) to compare the solution to observations located in either the Memory File or any open data bank; this option is also not relevant at the moment.

Notice that if we compare the current solution to the data bank values, the comparison here will be between history and an in-sample solution. If we compare the two solutions we have made, the comparison does not directly tell us about the properties of either simulation, relative to history, but only in relation to each other. Let us consider both of these comparisons.

To compare the historical simulation, KLNSOL1, to the observed historical values, put the code name DB (Data Bank) into the text area of the form shown in Figure 43. Make sure that the data bank, KLEINBNK.BNK is currently open; if it is not, cancel out of this form, click on **SOLVE** and then open this bank using the dropdown menu that appears (Notice this option is shown in Figure 37 above). When you have entered DB in the form in Figure 43, and then clicked **OK**, you should immediately observe a new status bar item, namely KLEINBNK.BNK, immediately to the right of the item that reads KLNSOL1.SOL. These status bar entries tell you that KLNSOL1 is attached and that it can be compared to KLEINBNK.

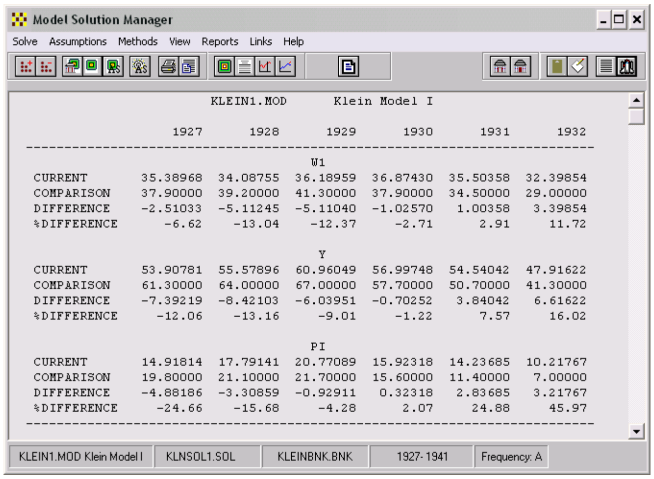
There are several different comparisons that can be made. Look at the dropdown menu that appears when you click on the **View** item of the Model Solution Manager menu bar, as now shown in Figure 44. The options **Full Comparison Table**, **Differences Only**, and **Percent Differences Only**, newly enabled, provide tabular comparisons. **Alternate Solution Comparison Plots** provides a way to compare the current solution graphically, successively displaying graphs of the model variables, endogenous first. In addition, whenever appropriate, you can compute **Prediction Error Statistics**, although this option requires you to first establish a Variable List (**Variable List Selection**), or VList, to identify the particular model variables for which to compute prediction error statistics. Generally, the prediction error statistics are only meaningful when a comparison is made between endogenous variable solution values and observed historical values of the comparable variables in a data bank or Memory File; however, MODLER will allow you to make the computations between alternative solutions as well, but the interpretation is then entirely your own.



```
Solution Information
   Dynamic Solution
Solution Period:   1
Convergence Test: Al
   Convergence Crite
Iteration Limits:
    Minimum=    2
    Maximum=   25
Method: Gauss-Seidel


Convergence Requirem
  1927:   17 Iterati
  1928:   11 Iterati
  1929:   12 Iterati
  1930:   11 Iterati
  1931:   14 Iterati
```

Figure 44. View Options

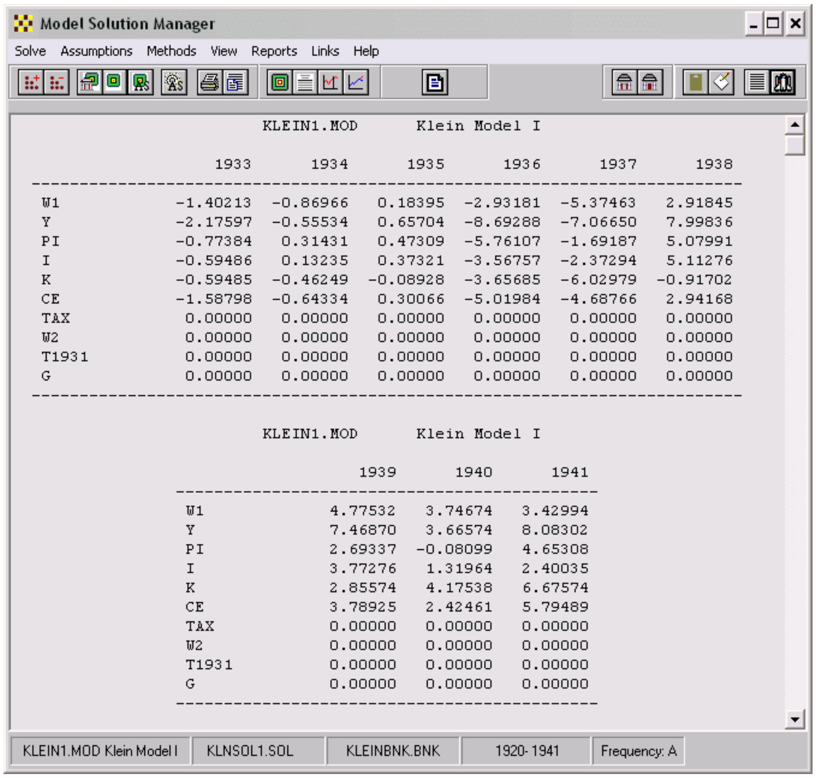
The Full Comparison Table is worth producing, inasmuch as it provides a tabular comparison that, variable by variable, displays the values of each variable in the model, plus the differences and percent differences. If you display this table without doing anything else, you will produce a scrollable table (albeit extensive) that shows every variable in the model, in model solution order. Alternatively, you can make a Variable List Selection, to create a specific variable list (VLIST) that allows you to display this table for specific model variables.



```
                        KLEIN1.MOD       Klein Model I

                   1927       1928       1929       1930       1931       1932
 ---------------------------------------------------------------------------
                                          W1
  CURRENT       35.38968   34.08755   36.18959   36.87430   35.50358   32.39854
  COMPARISON    37.90000   39.20000   41.30000   37.90000   34.50000   29.00000
  DIFFERENCE    -2.51033   -5.11245   -5.11040   -1.02570    1.00358    3.39854
  %DIFFERENCE      -6.62     -13.04     -12.37      -2.71       2.91      11.72

                                           Y
  CURRENT       53.90781   55.57896   60.96049   56.99748   54.54042   47.91622
  COMPARISON    61.30000   64.00000   67.00000   57.70000   50.70000   41.30000
  DIFFERENCE    -7.39219   -8.42103   -6.03951   -0.70252    3.84042    6.61622
  %DIFFERENCE     -12.06     -13.16      -9.01      -1.22       7.57      16.02

                                          PI
  CURRENT       14.91814   17.79141   20.77089   15.92318   14.23685   10.21767
  COMPARISON    19.80000   21.10000   21.70000   15.60000   11.40000    7.00000
  DIFFERENCE    -4.88186   -3.30859   -0.92911    0.32318    2.83685    3.21767
  %DIFFERENCE     -24.66     -15.68      -4.28       2.07      24.88      45.97
 ---------------------------------------------------------------------------
```

Figure 45. Full Comparison Table

Alternatively, to obtain a more general view, we can look at the table of differences (**Differences Only** option), as displayed in Figure 46. In the absence of setting a specific VLIST, you will observe that this table shows how the model simulates each of the variables relative to the observed historical values in KLEINBNK. The exogenous variables are of course the same in each case, so that these differences are zero, as we would expect.



```
                    KLEIN1.MOD      Klein Model I

                   1933      1934      1935      1936      1937      1938
 ----------------------------------------------------------------------------
 W1            -1.40213  -0.86966   0.18395  -2.93181  -5.37463   2.91845
 Y             -2.17597  -0.55534   0.65704  -8.69288  -7.06650   7.99836
 PI            -0.77384   0.31431   0.47309  -5.76107  -1.69187   5.07991
 I             -0.59486   0.13235   0.37321  -3.56757  -2.37294   5.11276
 K             -0.59485  -0.46249  -0.08928  -3.65685  -6.02979  -0.91702
 CE            -1.58798  -0.64334   0.30066  -5.01984  -4.68766   2.94168
 TAX            0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
 W2             0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
 T1931          0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
 G              0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
 ----------------------------------------------------------------------------

                    KLEIN1.MOD      Klein Model I

                             1939      1940      1941
               ---------------------------------------------
                W1        4.77532   3.74674   3.42994
                Y         7.46870   3.66574   8.08302
                PI        2.69337  -0.08099   4.65308
                I         3.77276   1.31964   2.40035
                K         2.85574   4.17538   6.67574
                CE        3.78925   2.42461   5.79489
                TAX       0.00000   0.00000   0.00000
                W2        0.00000   0.00000   0.00000
                T1931     0.00000   0.00000   0.00000
                G         0.00000   0.00000   0.00000
               ---------------------------------------------
```

Figure 46. Differences: KLNSOL1 versus Historical

Observe also that if you click on the menu item **Solve**, at the upper left hand corner of this screen, a drop down menu will appear the elements of which include the options **Print to Printer** and **Copy Display to Word Processor**. The first of these obviously will permit you to produce a hard copy version of the displayed table(s). The second will also permit you to edit the results. In this case you need to bear in mind that the table is originally produced in Courier monotype format, rather than a justified typeface like Times Roman, which will not preserve the columns. However, if you are careful, you can do quite a bit of editing of this table and in the end produce a comparison table that could be shown to someone else in order to communicate the effects. Notice, in addition, that it is possible to edit this table on screen: the blotter space in which it is displayed is a standard textbox. Thus, if you wish, you could for example remove the rows that contain the zero change values for the exogenous variables, then copy the edited display to your word processor. If you place your mouse on the blotter space, then click your *right* mouse button, MODLER will give you access to the usual variety of text editing facilities. If you instead double-click your *left* mouse button, you will be asked if you wish to edit the text display using MODLER's inbuilt text editor.

Similarly, with a little work, you can copy all or part of this display to PowerPoint or other presentation facility, although you may need to allow explicitly for the text, rather than image, nature of the display when you capture the table(s). MODLER provides you with access to a word processor, a spreadsheet program, or a presentation manager program, provided that you have set these up properly as defaults, as described in the MODLER *Getting Started Guide*, but you will need to understand how to use each of these in order to make full use of them.

It is also useful to look at a graphics display (**Alternate Solution Comparison Plots** option), and the comparison of the simulation of Consumption Expenditures with the historical values is shown in Figure 40. You will be able to display these for each of the model variables in turn. From 1920 to 1932, the values are the same in KLNSOL1 and KLEINBNK. In 1933, the first period of the simulation, differences are immediately evident in this plot.

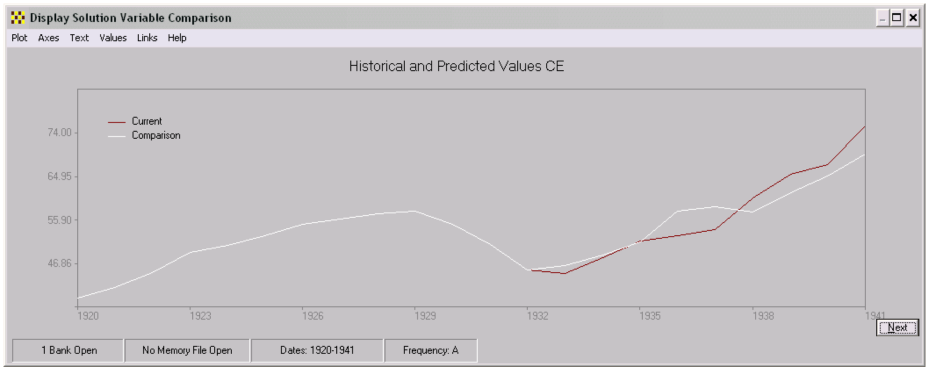
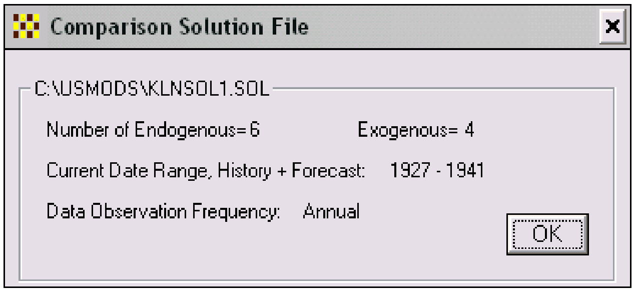


Figure 47. **Comparative Plot**

Finally, observe from Figure 44 that you can, if you wish, display also the model equations at this point, so that you can look at the results in that context as well. As a matter of screen control, if you display the equations using the **View** option **Model Equations**, MODLER will of course initially focus on these, keeping you from simultaneously looking at a plot or table. However, you can use the inbuilt capability to edit this display screen, using your right mouse button, in order to cut-and-paste the equations to your notes editor, which will then allow you to control better the alternative screen displays; click on the menu item **Links**, in order to execute your notes editor. The virtue of using your notes editor here is that then the displays of tables and plots will be under MODLER's control whereas, completely separately, the equations will be displayed by Notepad, Wordpad, or whichever notes editor you use, providing you with the ability to manipulate both displays simultaneously. Alternatively, you could instead use MODLER's facilities to capture in turn, tables and plots, pasting these to your notes editor display, then use MODLER to display the equations. Within a given program, there is inevitably a tendency to focus on the display of one thing at a time, but by taking full advantage of the linkages and display capture facilities, you have the capability to remove this limitation. Your only real limit is the size and resolution of your screen.

If you now instead return to the Model Solution Control screen and once more select **Solve** and then **Compare Solution To**, you might want to choose to compare KLNSOL1 and KLNSOL2. Once you have selected the alternative solution, instead of the data bank, the status bar on this screen will display the names of both solution files, letting you know that comparing them is possible. Initially you will see the form shown in Figure 48, which provides you with information about the alternative solution.



C:\USMODS\KLNSOL1.SOL

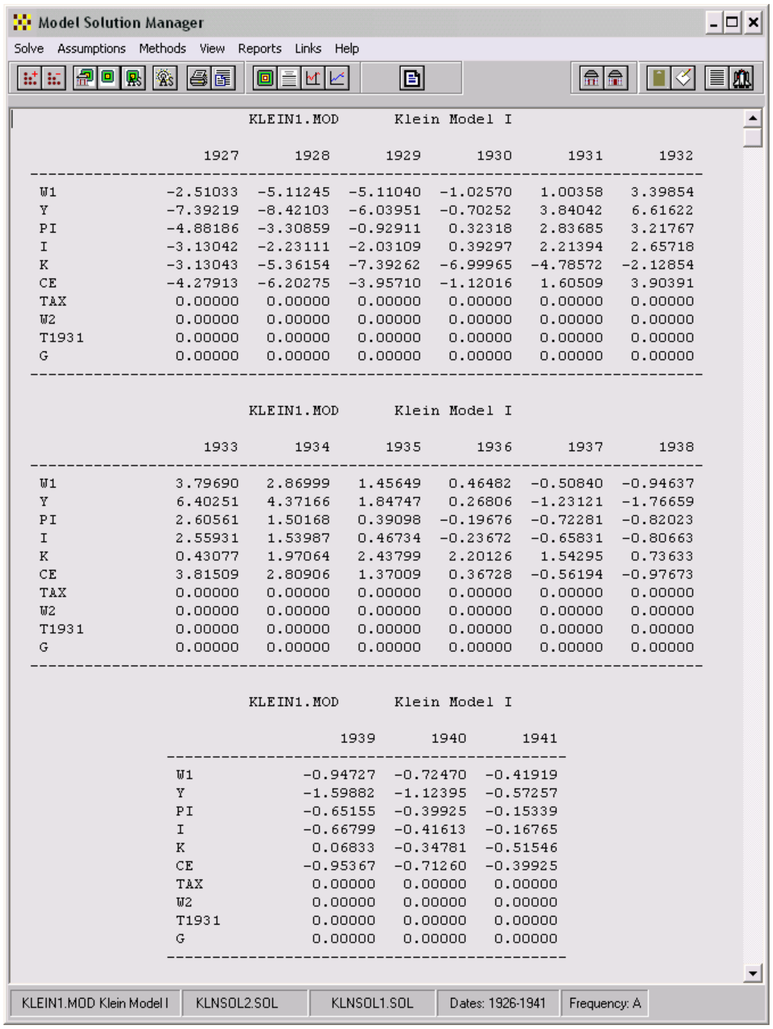
Number of Endogenous= 6 Exogenous= 4

Current Date Range, History + Forecast: 1927 - 1941

Data Observation Frequency: Annual

Figure 48. Comparison File Information Notice

The difference comparison for the two solutions is shown in Figure 49. Notice that in this case also, the exogenous variable values are the same, on both sides of the comparison. Therefore the observed differences can be traced to the fact that Klein Model I is a dynamic model, with lagged values populating its estimated equations.

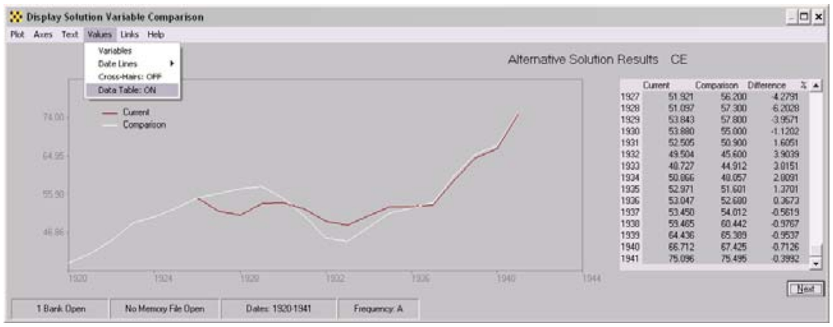


```
                          KLEIN1.MOD      Klein Model I

                     1927      1928      1929      1930      1931      1932
  --------------------------------------------------------------------------
   W1            -2.51033  -5.11245  -5.11040  -1.02570   1.00358   3.39854
   Y             -7.39219  -8.42103  -6.03951  -0.70252   3.84042   6.61622
   PI            -4.88186  -3.30859  -0.92911   0.32318   2.83685   3.21767
   I             -3.13042  -2.23111  -2.03109   0.39297   2.21394   2.65718
   K             -3.13043  -5.36154  -7.39262  -6.99965  -4.78572  -2.12854
   CE            -4.27913  -6.20275  -3.95710  -1.12016   1.60509   3.90391
   TAX            0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
   W2             0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
   T1931          0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
   G              0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
  --------------------------------------------------------------------------

                          KLEIN1.MOD      Klein Model I

                     1933      1934      1935      1936      1937      1938
  --------------------------------------------------------------------------
   W1             3.79690   2.86999   1.45649   0.46482  -0.50840  -0.94637
   Y              6.40251   4.37166   1.84747   0.26806  -1.23121  -1.76659
   PI             2.60561   1.50168   0.39098  -0.19676  -0.72281  -0.82023
   I              2.55931   1.53987   0.46734  -0.23672  -0.65831  -0.80663
   K              0.43077   1.97064   2.43799   2.20126   1.54295   0.73633
   CE             3.81509   2.80906   1.37009   0.36728  -0.56194  -0.97673
   TAX            0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
   W2             0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
   T1931          0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
   G              0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
  --------------------------------------------------------------------------

                          KLEIN1.MOD      Klein Model I

                                   1939      1940      1941
                ------------------------------------------
                 W1            -0.94727  -0.72470  -0.41919
                 Y             -1.59882  -1.12395  -0.57257
                 PI            -0.65155  -0.39925  -0.15339
                 I             -0.66799  -0.41613  -0.16765
                 K              0.06833  -0.34781  -0.51546
                 CE            -0.95367  -0.71260  -0.39925
                 TAX            0.00000   0.00000   0.00000
                 W2             0.00000   0.00000   0.00000
                 T1931          0.00000   0.00000   0.00000
                 G              0.00000   0.00000   0.00000
                ------------------------------------------
```

Figure 49. Solution Comparison: KLNSOL1 and KLNSOL2

In the same way as before, we can take advantage of the comparison setting to display a comparison plot of the two solutions, which for Consumption Expenditures is done in Figure 50, shown on the next page. Recall that KLNSOL1 simulates from 1933, whereas the KLNSOL2 simulation begins in 1927. However, while the simulation periods differ, notice that both these solution files contain historical data from 1920, so that they are comparable in terms of the overall coverage of the time period from 1920 to 1941; in terms of the overall time period, compared solution files must be compatible with each other.

Observe also that in Figure 50, the drop down menu for **Values** is selected, in order to reveal the element **Data Table: On**. The data table is the columnar table on the right of the screen. If you click on the displayed element, this data table will disappear; it can be turned on and off by toggling this menu element. As before, to view a succession of comparison plots of the model's variables, either press the **Next** button or the <Enter> key.



	Current	Comparison	Difference %
1927	51.921	56.200	-4.2791
1928	51.097	57.300	-6.2028
1929	53.843	57.800	-3.9571
1930	53.880	55.000	-1.1202
1931	52.505	50.900	1.6051
1932	49.504	45.600	3.9039
1933	48.727	44.912	3.8151
1934	50.066	48.057	2.0091
1935	52.971	51.601	1.3701
1936	53.047	52.680	0.3673
1937	53.450	54.012	-0.5619
1938	59.465	60.442	-0.9767
1939	64.436	65.389	-0.9537
1940	66.712	67.425	-0.7126
1941	75.096	75.495	-0.3992

Figure 50. Comparing KLNSOL1 to KLNSOL2

Predicting the Future: An Ex Ante Prediction

The period since 1941 is historical, but inasmuch as the data for the Klein model were specifically constructed for this model, in conformity with its identities, we do not now have observations on the US economy from 1942 that can be regarded as directly usable. It is for this reason, as much as anything else, that the predictions about to be considered are categorized here as *ex ante*, notwithstanding that we certainly have some idea of the characteristics of the performance of the US economy during the model's immediate post-sample period.

This absence of fully compatible historical observations after 1941 in fact provides us with a certain freedom. We can, if we wish, take account of the historical reality that the period from 1941 through 1945 was a time of war. Alternatively, we can ignore World War II and pretend instead that something else happened then. Recall that, although the United States was beginning to respond to the war in Europe in 1940-1941, it was not until the very end of that time, beginning on the 7th of December 1941, that the US was actually at war. Furthermore, although the war continued from 1942 until August 1945, it was nevertheless essentially a foreign war, resulting in virtually no damage to the US mainland nor to US productive capacity, yet at the same time having an enormously stimulative effect on the US economy. Obviously, the goods and services produced during this time were to an unusual degree military, naval, and otherwise war-related. In addition, the willingness (and the political wherewithal) to commit the government to the spending program undertaken during the period 1942-1946 was directly war-sponsored. Therefore, it may be difficult to imagine the period from 1942 through 1945 as other than it was, particularly if we were to wish to imagine a period similarly economically expansive, but it is at least possible to conceive that an alternative reality might have prevailed that would have transformed the economy—to somewhat the same degree—from depression to prosperity during the decade of the 1940s.

However, we need also to consider the specific mechanisms involved in making one or more forecasts for this time. Notwithstanding history, and whatever we might be able to imagine about the post-sample period otherwise, in the context of MODLER there are three ways to specify the values of the exogenous variables as a set of assumed values. The first option is to load these values of the variables into the relevant data bank and then create a new solution file, setting the dates for this operation so as to include both history and the "future." As a general proposition, using the **Make Solvfile** operation to load *both* historical values and assumed values for the exogenous variables can always be done. Recall from earlier discussion that, when creating a solution file, MODLER will always attempt to load data for the whole of the specified Make SolvFile period, in the process copying into the solution file the values of each variable to the extent these are available, variable by variable. So long as observations exist in the data bank (or other accessible date source), to include them is simply a matter of setting the dates inclusively.

Alternatively, we can add observations to an existing solution file and there are two ways to accomplish this. The first method uses the SIMSET option and the second uses a macro file. The benefit of the first of these is that the process of specifying the assumed values is onscreen and interactive. The benefit of the second option is that a full audit trail can be generated more or less automatically, allowing us, after the fact, to determine step-by-step how any forecast was made. It is generally true that interactive methods of program control are both intuitive and relatively simple to perform—reflecting that this mode of operation is designed to have these characteristics—but there is the downside that the history of the actions performed tends to be obliterated as changes are progressively made. Moreover, even if in principal a flexible audit trail could nonetheless be laid out, it is generally difficult under these conditions to make a computer program do this in a way that permits particular operations to be undone and then redone in an alternative way. The consequence is that you generally need to make a choice of method: MODLER gives you the choice between onscreen and interactive or else creating a command macro. When a command macro is used, each variation can then be preserved and possibly run or re-run as many times as you might wish, giving you a way of preserving and documenting the specific set of assumptions used to produce each forecast made.

These alternative operating modes can each be invoked from the Model Solution Manager screen: Begin at this screen and start by clicking on the menu item **Solve** and then choose **Open Solvfile As …** , in order to create a new solution file from the baseline Solvfile. Arbitrarily name this new Solvfile something like KlnXAnte.SOL. Then whatever you set the final date of the solution file to be, make sure to set the original date as 1920. The reason to proceed in this manner is because—if we are now going to make an ex ante prediction, beginning the first period after the sample period ends—we will certainly want to use as history the (estimation) sample period values. These values should therefore be included in those retrieved from the baseline Solvfile.

Next, click on the **Assumptions** menu item of this same Solution Manager screen, which will result in a display like that shown in Figure 51.

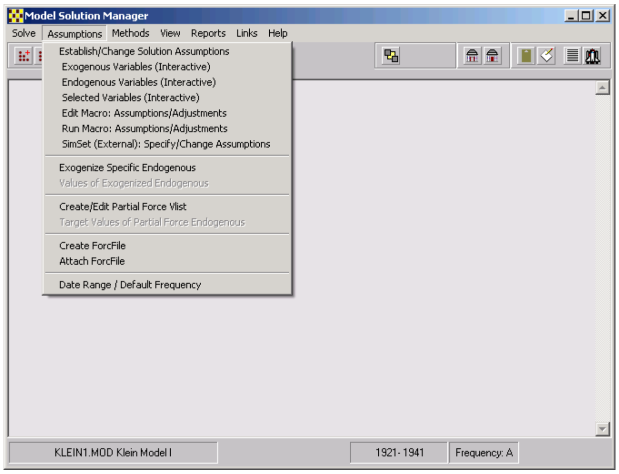
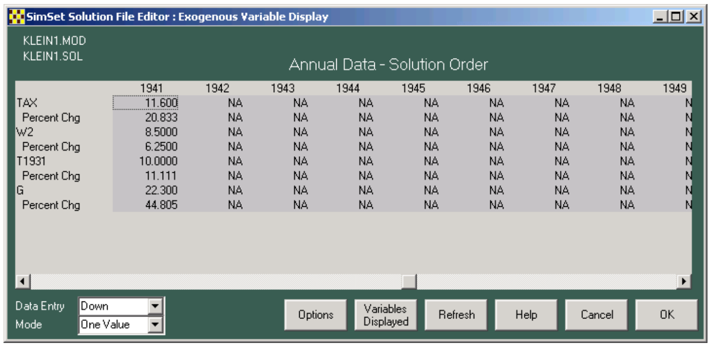

Model Solution Manager
Solve
Assumptions
Methods
View
Reports
Links
Help
Establish/Change Solution Assumptions
Exogenous Variables (Interactive)
Endogenous Variables (Interactive)
Selected Variables (Interactive)
Edit Macro: Assumptions/Adjustments
Run Macro: Assumptions/Adjustments
SimSet (External): Specify/Change Assumptions
Exogenize Specific Endogenous
Values of Exogenized Endogenous
Create/Edit Partial Force Vlist
Target Values of Partial Force Endogenous
Create ForcFile
Attach ForcFile
Date Range / Default Frequency
KLEIN1.MOD Klein Model I
1921- 1941
Frequency: A

Figure 51. Specifying Exogenous Variable Values

Observe the items that appear in the dropdown menu under the heading **Establish/Change Solution Assumptions**. The first is **Exogenous Variables (Interactive)**. The second is **Endogenous Variables (Interactive).** At present, only these first two are relevant, in part because of the extreme simplicity of Klein Model I. For instance, there is no need to "select" among the model variables.

Initially, click on the first, **Exogenous Variables (Interactive)**. A form like that shown in Figure 52 should immediately appear. Notice first its caption, especially the second part *Exogenous Variable Display*, which states that the variable values you are looking at are exogenous. Notice also the horizontal slider element just above the bottom row of buttons, which can be used to control the particular observations displayed. As Figure 52 illustrates, initially the left-most values are the last observations of the historical time period and all the values to the right (which are all NA) are "future" time periods from the prospective of the estimated model. You will need to imagine that 1942 and later are the "future."



SimSet Solution File Editor : Exogenous Variable Display

KLEIN1.MOD
KLEIN1.SOL

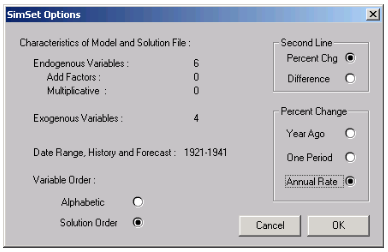
Annual Data - Solution Order

	1941	1942	1943	1944	1945	1946	1947	1948	1949
TAX	11.600	NA	NA	NA	NA	NA	NA	NA	N
Percent Chg	20.833	NA	NA	NA	NA	NA	NA	NA	N
W2	8.5000	NA	NA	NA	NA	NA	NA	NA	N
Percent Chg	6.2500	NA	NA	NA	NA	NA	NA	NA	N
T1931	10.0000	NA	NA	NA	NA	NA	NA	NA	N
Percent Chg	11.111	NA	NA	NA	NA	NA	NA	NA	N
G	22.300	NA	NA	NA	NA	NA	NA	NA	N
Percent Chg	44.805	NA	NA	NA	NA	NA	NA	NA	N

Data Entry: Down
Mode: One Value

Figure 52. SIMSET Solution File Editor: Exogenouse Variables Display

Looking next at the following few lines below the caption, the display evidently indicates that the model data are annual and that its exogenous variables are displayed in "solution order". The model is KLEIN1.MOD and the solution file used is KLEIN1.SOL. Next click on the **Options** button, which will cause the form in Figure 53 to be displayed.



SimSet Options

Characteristics of Model and Solution File :

Endogenous Variables : 6
Add Factors : 0
Multiplicative : 0

Exogenous Variables : 4

Date Range, History and Forecast : 1921-1941

Variable Order :
Alphabetic ○
Solution Order ●

Second Line
Percent Chg ●
Difference ○

Percent Change
Year Ago ○
One Period ○
Annual Rate ●

Figure 53. Simset Options

Look first at the lower left hand corner of this new form. You will see two “radio” buttons; these can be used to change the display order. For example, if you click on the white “Alphabetic” disc and then press the OK button, you will see that the display changes to “Annual Data: Alphabetic Order.” Look also at the right-hand-side of Figure 53, which indicates that the second line is percent change and that this percentage change is computed at annual rate

Look again at Figure 52. In the case of each *level* variable, observe that the immediate next row (the “second line”) displays the associated percentage change for each of its observations. Consequently the values in that row will automatically adjust whenever you change specific level values. Alternatively, if you wish to modify the percent change instead, the corresponding level values will adjust conformably. Evidently, what the screen exhibits are four pairs of rows, and within each pair the values are directly related. In some cases, it is easier to make and modify assumptions by adjusting the percent changes; in other cases, changing the levels makes more sense. Which way you do it is your choice. In addition, as just indicated by the display in Figure 53, the percentage change is computed at *annual rate*. Inasmuch as the observations are annual, this fact is not important in the present case: that is, the previous period is last year and annual changes are of course always “annualized.” However, if the model were a quarterly model, the interpretation of the percentage changes might need to be considered. For the moment, what needs to be recognized is simply that both the solution order and the characteristics of the “second line” are controlled by the Options screen. The relevant choice you might need to make is whether to show on the “second line” the percentage change or the period to period difference.

Now direct your attention to the bottom left hand corner of Figure 52. Notice the two dropdown boxes, labeled *Data Entry* and *Mode* respectively. Later, when you put your cursor on one of the observations values of a model variable, enter a new value, and press the **Enter** key, the Down data entry option signifies that the cursor will move down to the next row. However, if you put your cursor on the down arrow symbol just to the right of “Down,” you will see that Data Entry offers three choices: Down, Skip down, and Across. Down moves you to the next line down, the percent change line. Skip down skips down to the next level variable. In contrast, Across will move the cursor along a given line to the observation just to the right. In addition, it is important for you to be aware that *each time you press the Enter key*, at least one observation is added to the Solution File. For each variable and each observation, what you see on the screen is the value in the solution file. For more information, the **Help** button describes the operation of this SimSet facility, as it is called.

Furthermore, notice that *because everything happens in real time*, the **Cancel** button does *not* undo previous changes made to variable values; it simply cancels out of the SimSet facility, almost in the same way that pressing the **OK** button operates. Just for the moment, you should limit yourself to pressing only the **Options**, **Help**, **Cancel** and **OK** buttons. Please do not touch the **Variables Displayed** or **Refresh** buttons, as these need more explanation before you attempt to use them. Also, please do not change the Mode yet: leave it at One Value.

With this introduction, let us begin to consider what we might do with Klein Model I. When you begin to consider what values to enter, it is important initially to take into account that the identities:

$$Y = PI + W1 + W2$$

And

$$Y{+}TAX = CE + I + G$$

together imply that:

$$\begin{aligned} PI &= CE + I + G - TAX - (W1{+}W2) \\ &= CE + I + (G - TAX - W2) - W1 \end{aligned}$$

The Profits variable, PI, is the residual of Purchases less the wage bill and business taxes. All other things equal, by implication, growth in Government purchases net of Business Tax Yield, TAX, and the labor income originating in government, W2, increases profits. Of course, in addition, increases in W2, on its own, increases income, Y. Notice also the further implication that, although there are obviously second round effects from the respective changes in these values, it is evident that we cannot make independent assumptions about the behavior of these "policy" variables.

The characteristics of these policy variables, G, TAX, and W2, can to some degree be assessed if we look at their historical time plots, as respectively shown in Figures 54-56. However, before beginning to consider their time paths, it is relevant to remind ourselves exactly what these are. The first, G, which is called by Klein "Exogenous Investment," is the sum of several components: Net Exports and Monetary Use of Gold and Silver, deflated by the Wholesale Price Index (1934=1.00); Public Construction Expenditures, including Work Relief Construction, deflated by the American Appraisal Company's Index of Construction Costs (1934=1.00); and Government Expenditures for Goods and Services, less Government Interest Payments and Public Construction Expenditures, including Work Relief Construction, all deflated by the Index of Wholesale Prices of Non-farm products (1934=1.00). TAX, in turn, essentially consists of business excise taxes and W2 is the labor income originating in government. The TAX item actually represents the difference between national output at market prices, or Net National Product, and national income at factor costs, or Net National Income; this difference, in 1934 dollars, is approximately business (excise) taxes. Finally, W2 is government wages and salaries, including work relief wages, deflated essentially using a Net National Income price deflator variable (1934=1.00).

A careful analysis of the situation would also need to take into account the way in which the economic context of the 1930s and 1940s differs from that today. For example, in the 1930s, the military budget and entitlements were quite different compared to now, in terms of their relative size and role in the federal government budget, affecting the way in which government expenditures might reasonably be assumed to increase or decrease year to year. Similarly, specifying a reasonable pattern of tax yield changes that might be occur in the context of the economy then would need to take into account the nature of the tax yield variable in Klein Model I versus the tax yield variable(s) that might appear in a more modern macroeconometric model. For instance, the model as it stands obviously provides no mechanism for considering changes in personal income tax rates. In a formal sense, it may be possible to use this model to illustrate the use of the MODLER software, but the danger of making false inferences is substantial when doing so. We need to think carefully about the definition of each of the variables, among other things.

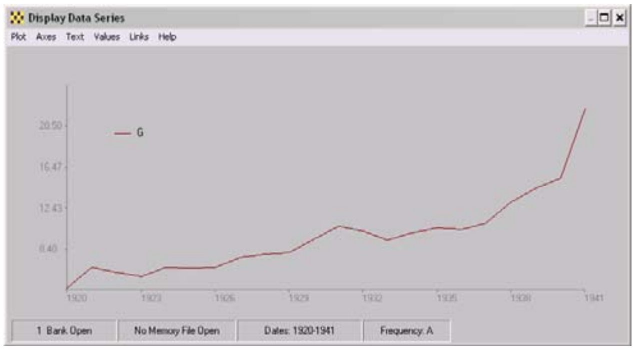


Figure 54. Government Purchases

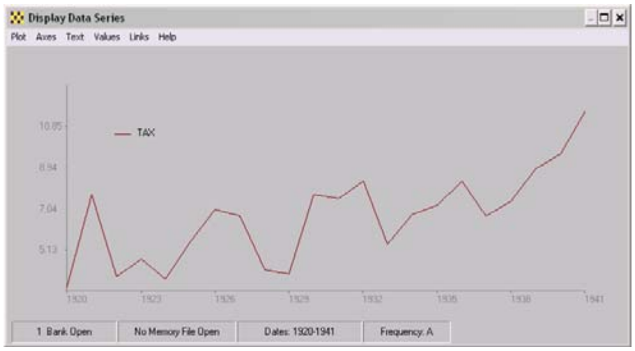


Figure 55. Business Tax Yield

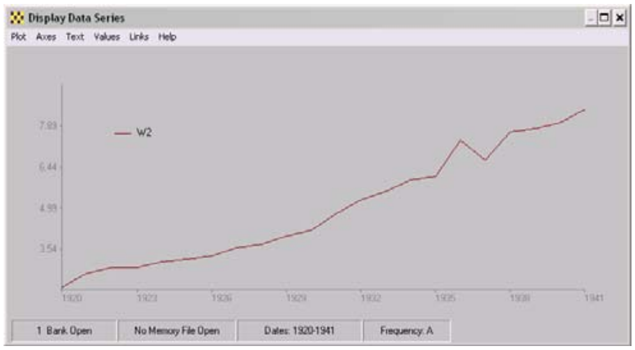


Figure 56. Labor Income Originating in Government

In contrast to these individual variable graphs, a composite view is provided by Figure 57, below, which displays historically the net of government purchases over business tax yield. What Figure 57 indicates, in particular, is that over the estimation sample period, only during the period from about 1936 did government purchases persistently begin to increase relative to the business tax yield. But, remembering the earlier discussion. Be careful not to read into this graph a statement about budget deficits, inasmuch as not all taxes (nor all non-tax payments to government) are accounted for here. Compared to more modern models, Klein Model I is restricted in its capability to simulate such things as the effect of a balanced budget.

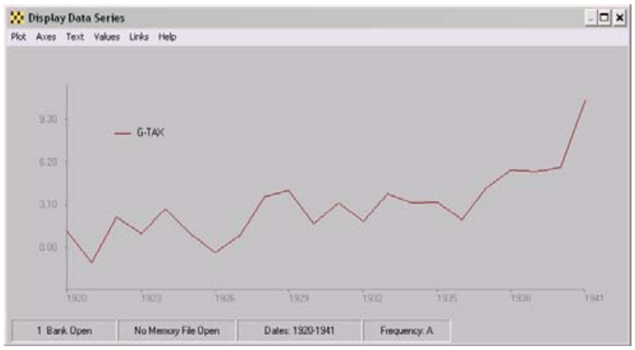


Figure 57. Government Purchases Less Business Tax Yield

In a somewhat similar manner, Figure 58 illustrates that Government non-government labor purchases only began to increase persistently beginning in 1936.

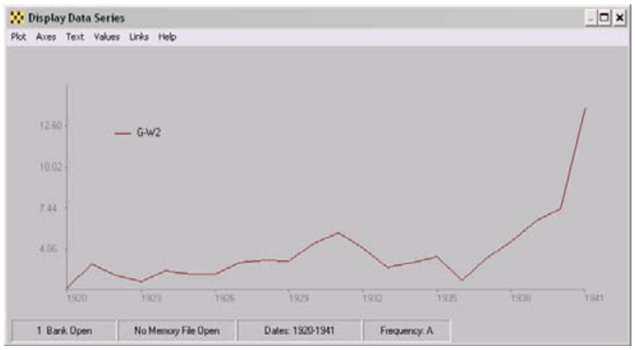


Figure 58. Government Purchases Net of Income Originating in Government

However, as just mentioned, we need to be careful in the assumptions we make. Compared to models created since 1964, or thereabouts, which take into account the complete set of US National Income and Product Accounts, produced by the Bureau of Economic Analysis of the US Department of Commerce (or the Office of Business Economics, or OBE, as it was called in the 1960s), this model is restrictive both in terms of the accounts coverage and the degree to which government purchases are split into easily identifiable components. For example, labor income originating in government does not distinguish between the number of people employed and the average annual income originating [19]. But we do know that, conceptually, a decomposition can be made as follows:

$$W2 = Ngov * AIO$$

Where Ngov denotes the number of people employed and AIO is the Average Income Originating. Simply dividing the number employed into W2, the income originating, would generate AIO. Furthermore, it follows immediately that the rate of change of W2 is:

$$\% \text{ chg } W2 = \% \text{ chg } Ngov + \% \text{ chg } AIO$$

so that it is possible, with a few simple calculations, to compute the assumed changes in both employment and the average “wage” (more strictly, average income originating), taking into account simultaneously (using MODLER notation) that:

$$W2 = (1+ \% \text{ chg } W2/100) * W2(-1)$$

The virtue of making the above decomposition is that it permits us to deal separately with the assumed increase in employees and the increase in the average "wage." In fact, if one wished to embed into the model itself these additional formulae as identities, it would be easy to do, in the process providing two additional "policy" variables. Bear in mind, however, that civilian government employees are conceptually included in the labor force, but that soldiers, sailors, marines, and airmen are not. Consequently, W2 will not incorporate the manpower effects of any increase in defense expenditures.

Recall also that the variable G, which has here been called Government Purchases, is called by Klein "Exogenous Investment" [19]. It is specifically defined as:

$$G = EXGS/WPI + GCon/CCI + (GPGS - GInt - GCon) / WPIXF$$

Where:

EXGS = Net Exports and Monetary Use of Gold and Silver
WPI = Wholesale Price Index, 1934=1.00
GCon =Public Construction Expenditures, including work-relief construction
GPGS = Government Expenditures for Goods and Services
GInt = Government Interest Payments
WPIXF = Index of Wholesale Prices of Non-farm Products, 1934=1.00

This identity also could, in principle, be included in the model, although it would then be necessary to obtain the individual component estimates from the several sources identified by Klein. If instead we are willing to be a little cavalier, we can instead use the formula:

$$G = (1+ \% \text{ chg } G/100) * G(-1)$$

In the process simply assuming the rate of growth year to year during the forecast period. However, even if we take this short cut, it is useful—as a basis for considering the assumed rate of growth—to know, from the definition above, what G refers to precisely. Here % chg G/100 is of course the percentage rate of growth, expressed in decimal form.

Finally, the variable TAX, which has above been called Business Tax Yield, appears in the accounts [19] in the context:

Y+TAX = Net National Product, measured in billions of 1934 dollars with the variable Y, in turn, individually defined as:

$$Y = PI + W1 + W2$$

Hence, it follows that:

$$TAX = Y – PI – (W1+W2)$$

as a *derived* accounting identity. However, this identity is not particularly useful conceptually as the basis on which to generate forecast assumptions for the TAX variable. What is needed instead is a formula something like:

$$TAX = (TAX/PI) * PI$$

or

$$TAX = (TAX/Y) * Y$$

where, during the forecast period, the "tax rate" variables TAX/PI or TAX/Y might then be assumed to behave in a particular way. These are each de facto a specification of an implied tax rate, relative to profits or Gross National Product. They have no equivalent in the US tax code then or now, so that these must be regarded simply as synthetic variables, constructed simply in order to try to get some sort of handle on the way in which the business tax yield could be expected to behave. To this end, their historical behavior as derived rates is shown, respectively, in Figures 59 and 60.

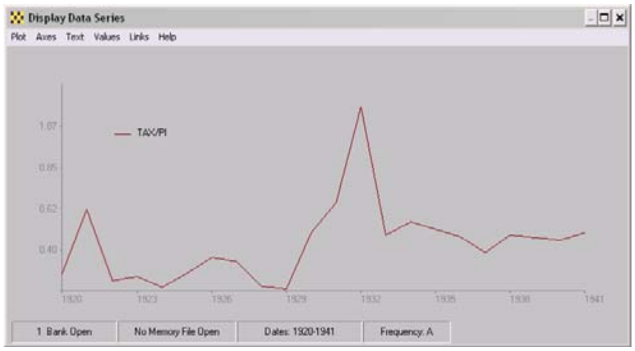


Figure 59. Business Tax Yield as a Fraction of Profits

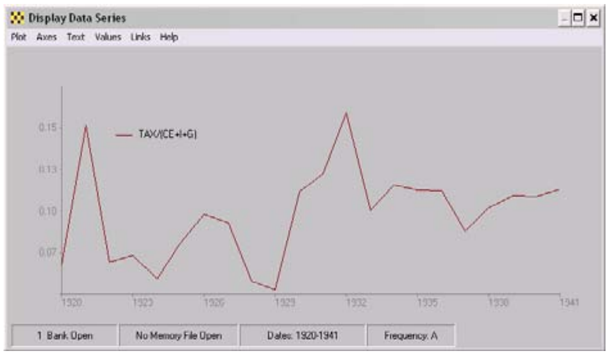
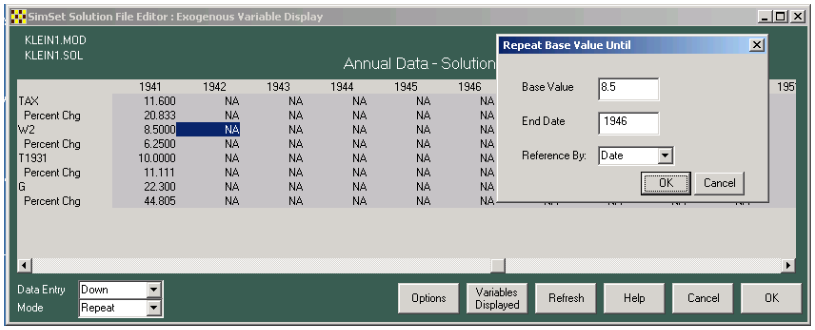


Figure 60. Business Tax Yield as a Fraction of Net National Product

Unfortunately, neither of these graphs provides a representation of a tax rate that might be expected *a priori*. For example, assuming that the tax yield and profits are correctly measured as accruing in each year, and if the revenues were generated as a fraction of profits, in the usual way, the graphical display that might be expected would be some type of step function, the steps being recorded at the points in time that the marginal or proportional rates changed historically. Inasmuch as, conceptually, the same deflator would be used to compute the real yield and real profits, the fact that the variables are computed as a ratio of real values should make no difference. However, such logic goes for naught here: in the event, an “effective” tax rate, as a fraction of profits, that varies year-to-year from approximately 30% in 1929 to more than 100% in 1932 does not appear to have much to commend it.

We can at least nibble at the edges. First consider the SimSet display shown earlier in Figure 52. We can just extend for a few years the values of W2. Let us do this by first changing the Mode setting from One Value to Repeat, as illustrated in Figure 61.



KLEIN1.MOD
KLEIN1.SOL

Annual Data - Solution

	1941	1942	1943	1944	1945	1946
TAX	11.600	NA	NA	NA	NA	NA
Percent Chg	20.833	NA	NA	NA	NA	NA
W2	8.5000	NA	NA	NA	NA	NA
Percent Chg	6.2500	NA	NA	NA	NA	NA
T1931	10.0000	NA	NA	NA	NA	NA
Percent Chg	11.111	NA	NA	NA	NA	NA
G	22.300	NA	NA	NA	NA	NA
Percent Chg	44.805	NA	NA	NA	NA	NA

Repeat Base Value Until

Base Value: 8.5
End Date: 1946
Reference By: Date
OK Cancel

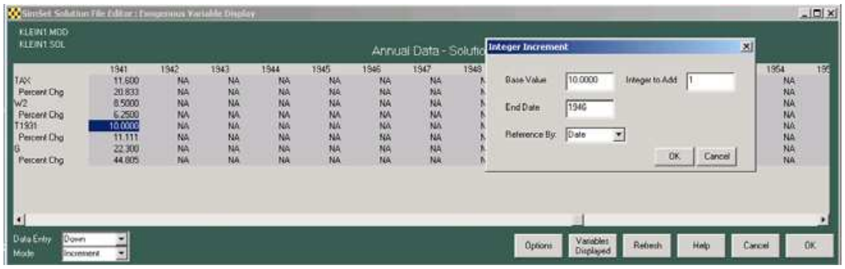
Data Entry: Down
Mode: Repeat

Figure 61. Repeating an Observation Value

Once Mode is changed to Repeat, if you then move your cursor to the 1942 value for W2 and click, the small form headed "Repeat Base Value Until" will suddenly appear superimposed on the SimSet form. However, as it is movable, simply by left clicking on the top edge of it, holding and then moving the cursor, in order to put it out of the way on the right-hand-side, as shown.

Considering this small form, the base value to which it refers is the "point of click," namely the 1942 value of W2, which has been specified at 8.5, the same value as in 1941. The "End Date" is given as 1946, and the meaning of "Reference By Date" is that the endpoint of the change is a date, rather than an observation number. When the **OK** button on this form is pressed, the effect will be to insert values for 1942 through 1946, inclusive, for W2. In effect, we are simply repeating a value for a specified number of years. However, before this will work properly, you will need to create the solution file in a particular way, as will be described.

The MODLER SimSet facility actually offers a number of Mode options in order to permit you to create observation values into the future on the basis of a variety of criteria. For instance, there is another option (Integer Increment) that will permit you to increment the values of T1931, which is simply a time trend with the characteristic that $x_{t+1} = x_t + 1.0$, starting in 1931. In the absence of this Mode facility you would need to enter each value individually. Given this facility, as shown in Figure 62, you can easily extend the incremented values of T1931 for a requisite number of years into the future. In this case, as shown, a positive increment value is used, but should you wish to decrement a variable's values, you need only to specify a negative "Integer to Add."



SimSet Solution File Editor : Exogenous Variable Display

KLEIN1.MOD
KLEIN1.SOL

Annual Data - Solutio

	1941	1942	1943	1944	1945	1946	1947	1948	1954	195
TAX	11.600	NA	NA	NA	NA	NA	NA	N	NA	
Percent Chg	20.833	NA	NA	NA	NA	NA	NA	N	NA	
W2	8.5000	NA	NA	NA	NA	NA	NA	N	NA	
Percent Chg	6.2500	NA	NA	NA	NA	NA	NA	N	NA	
T1931	10.0000	NA	NA	NA	NA	NA	NA	N	NA	
Percent Chg	11.111	NA	NA	NA	NA	NA	NA	N	NA	
G	22.300	NA	NA	NA	NA	NA	NA	N	NA	
Percent Chg	44.805	NA	NA	NA	NA	NA	NA	N	NA	

Integer Increment

Base Value: 10.0000
Integer to Add: 1
End Date: 1946
Reference By: Date

Data Entry: Down
Mode: Increment

Figure 62. Incrementing Observation Values

The full set of these Mode options are listed in Figure 63. In each case, except the "One Value" option, when any of these options is selected a small form, similar to that shown in Figure 62, will appear that will assist you by providing the necessary information.

ONE VALUE indicates that you wish to key in values one at a time.

REPEAT indicates that any value keyed in at the cursor is to be repeated k times in the following k periods. Depending upon whether the time-reference style has previously been set to DATE (the default) or PERIOD, you will be asked to specify the date or number of the period, k; if you specify the date, SimSet internally computes k.

ADD, **SUBTRACT**, **MULTIPLY**, and **DIVIDE**, respectively allow you to add to, subtract from, multiply, or divide the given variable value by another value. Procedurally, once any of these options is chosen, you first type in the value of the solution value (or simply press enter to confirm the existing value), then you will be asked to specify the value to be added (subtracted, multiplied by, or divided by) and then, once you have pressed Enter, the date or period until which the modification is to be repeated. Finally, press the form's **OK** button.

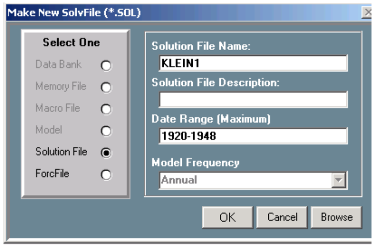
The **GROW** and **TREND** options are, roughly speaking, the converse of each other. GROW allows you to specify the percentage by which a given value should increase over a given number of periods (or until a given date). TREND allows you to specify the value in a future period (or later date), with the intervening values computed as lying along a linear trend. The growth rate under GROW will be interpreted as an annualized or period rate depending upon the PERCENT setting, which can be set or changed using the **Options** button considered earlier (the default is annual rate).

IDTREND (Increment-Decrement **TREND**) allows you to specify an additive trend for a set of values (endogenous, additive adjustment, or multiplicative adjustment). You will be asked to specify the contemporaneous additive value, the value in the future, and the date or period to which that second value refers. The implicit growth rate will be computed (including 0, implying simply the addition of some constant value).

INCREMENT (Integer Increment) permits you to add a positive or negative (usually integer) value to the base value and successive values (x_{t+1} = x_t+ value) for k periods into the future. Depending upon whether the time-reference style has previously been set to DATE (the default) or PERIOD, you will be asked to specify the date or number of the period, k; if you specify the date, SimSet internally computes the value k.

Figure 63. SimSet Data Entry Options

Operationally, the significant steps involved are the following. First, you need to create a solution file that combines in its historical and “future” observations the full date range. For example, this solution file might be created with the historical observations starting in 1920 and the “future” values extending to 1948. Starting from the Model Solution Manager Screen, first attach the model, which can be done by clicking on the *Attach Model* icon. You will also need to have previously opened the Kleinbnk data bank. Then click on the Create New Solution file icon, which will display a form like that shown in Figure 64.



Make New SolvFile (*.SOL)

Select One
- Data Bank ○
- Memory File ○
- Macro File ○
- Model ○
- Solution File ●
- ForcFile ○

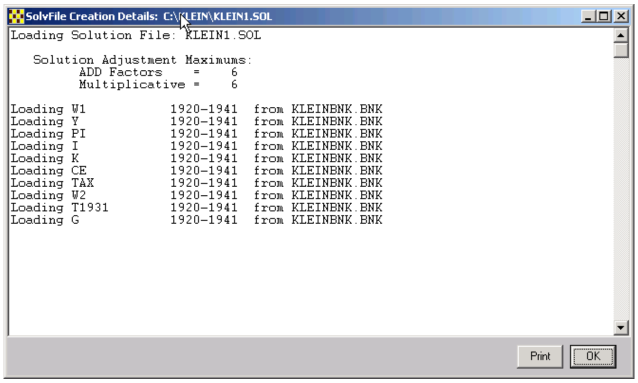
Solution File Name: KLEIN1

Solution File Description:

Date Range (Maximum): 1920-1948

Model Frequency: Annual

Figure 64. Create "Forecast" Solution File

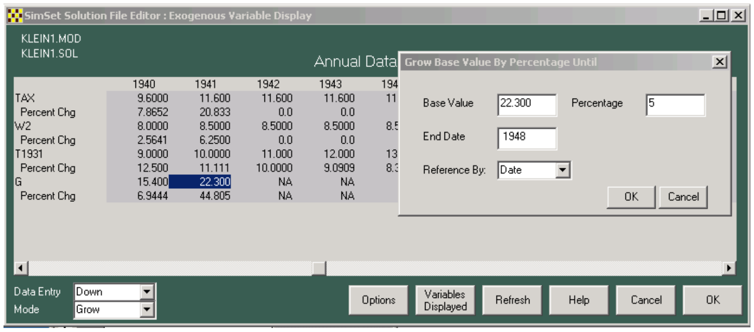


SolvFile Creation Details: C:\KLEIN\KLEIN1.SOL

```
Loading Solution File: KLEIN1.SOL

   Solution Adjustment Maximums:
         ADD Factors    =    6
         Multiplicative =    6

Loading W1          1920-1941  from KLEINBNK.BNK
Loading Y           1920-1941  from KLEINBNK.BNK
Loading PI          1920-1941  from KLEINBNK.BNK
Loading I           1920-1941  from KLEINBNK.BNK
Loading K           1920-1941  from KLEINBNK.BNK
Loading CE          1920-1941  from KLEINBNK.BNK
Loading TAX         1920-1941  from KLEINBNK.BNK
Loading W2          1920-1941  from KLEINBNK.BNK
Loading T1931       1920-1941  from KLEINBNK.BNK
Loading G           1920-1941  from KLEINBNK.BNK
```

Figure 65. Creating the New Solution File

Notice that when you click the **OK** button on the form shown in Figure 64, you will see a screen like that shown in Figure 65. Notwithstanding that you may have specified the date range from 1920 to 1948, inasmuch as the existing historical observations *exist only for the period from 1920 to 1941*, this solution file will contain only these observations. Once this solution file has been created, then choose **Exogenous Variables (Interactive)**, as previously done in the context of Figure 51, which will display a screen similar to that shown in Figure 66.



SimSet Solution File Editor : Exogenous Variable Display

KLEIN1.MOD
KLEIN1.SOL

Annual Data

	1940	1941	1942	1943	194
TAX	9.6000	11.600	11.600	11.600	11
Percent Chg	7.8652	20.833	0.0	0.0	
W2	8.0000	8.5000	8.5000	8.5000	8.5
Percent Chg	2.5641	6.2500	0.0	0.0	
T1931	9.0000	10.0000	11.000	12.000	13
Percent Chg	12.500	11.111	10.0000	9.0909	8.3
G	15.400	22.300	NA	NA	
Percent Chg	6.9444	44.805	NA	NA	

Grow Base Value By Percentage Until

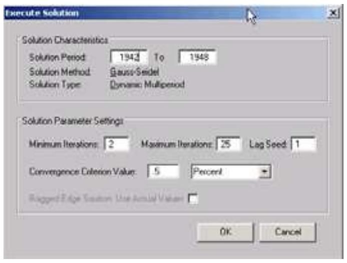
Base Value: 22.300
Percentage: 5
End Date: 1948
Reference By: Date

Data Entry: Down
Mode: Grow

Figure 66. Growing Government Expenditures by 5% Annually

Figure 66 displays the exogenous variables once the changes have been made. First, you will need to use the Repeat mode option in order to extend the tax yield, TAX, through 1948 from 1941, as well as the government wages, W2. Then use the *Increment* mode to incrementally extend the values of T1931. At this point, set the mode to "Grow" and then click on the 1941 value for G. You initially will also need to have used the slider button, to expose the values from 1940 through 48. When this SimSet screen initially appears, the leftmost observation will be that dated 1948, which will consist entirely of NA values.

Once the *Grow Base Value Until* option has been executed (by pressing the **OK** button on the small form shown in Figure 66), you will see a screen containing values through the year 1948. The percent change for TAX and W2 will be 0.0 for 1942 through 1948 – because the original values have simply been extended. For T1931, there will be a positive percentage, but variable, change each period and for G this percent change will be 5% in each of the time periods from 1942 through 1948. When you then press the **OK** button on the SimSet form, you will be returned to the Model Solution Manger screen, where you can select the menu item **Solve**. Clicking on this menu element will display the form shown in Figure 67.



Execute Solution

Solution Characteristics

Solution Period: 1942 To 1948
Solution Method: Gauss-Seidel
Solution Type: Dynamic Multiperiod

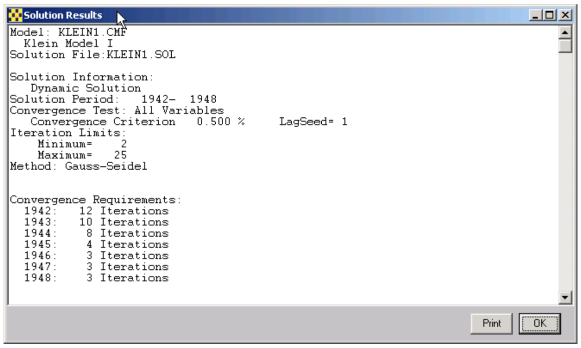
Solution Parameter Settings

Minimum Iterations: 2 Maximum Iterations: 25 Lag Seed: 1
Convergence Criterion Value: .5 Percent

Ragged Edge Solution: Use Actual Values □

Figure 67. Execute Forecast

You will need to enter 1942, as shown in Figure 67, and then click the **OK** button on this form. The form shown in Figure 68 will then appear.



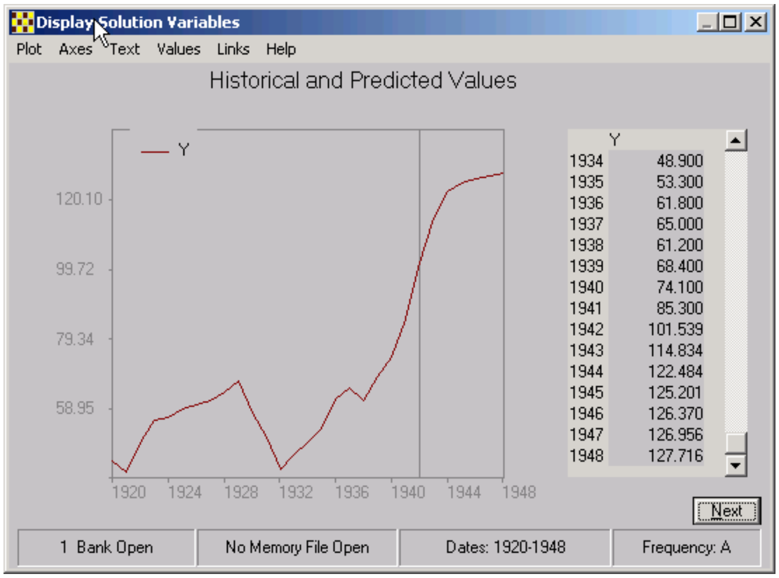
```
Model: KLEIN1.CMF
  Klein Model I
Solution File:KLEIN1.SOL

Solution Information:
   Dynamic Solution
Solution Period:   1942-  1948
Convergence Test: All Variables
   Convergence Criterion   0.500 %     LagSeed= 1
Iteration Limits:
    Minimum=    2
    Maximum=   25
Method: Gauss-Seidel


Convergence Requirements:
  1942:   12 Iterations
  1943:   10 Iterations
  1944:    8 Iterations
  1945:    4 Iterations
  1946:    3 Iterations
  1947:    3 Iterations
  1948:    3 Iterations
```

Figure 68. Solution Log Form

When you click the **OK** button on this form, you will once again see the Model Solution Manager screen. It might at that point be useful to click on the **View** menu element of this form and then choose **Current Solution Plots**. An example is shown in Figure 69. You can "walk" through these, pressing the **Next** button in each case. These generally present an image of a growing economy, but it is interesting that the Investment graph turns down. If, when the first of these graphs appears, you click on its **Values** menu element, and then **Dates Lines** and **Select**, you will be able to display the vertical line shown in Figure 69 at 1942. This line essentially marks the separation of the historical values from the predicted values.

You might also want to consider different growth rates for the G variable, leaving all the other exogenous variable values the same as before. For instance, if you instead set the growth rate for G from 1942 to 1948 at, say 2%, the growth of the model tends to falter. If you set the growth rate at a higher rate the model will obviously tend to grow faster. But be careful not to set this growth rate too high. Remember the discussion earlier of what G represents. Bear in mind also that Klein Model I was never originally considered in terms of its forecast performance. Putting the model through its paces as we have just done here was, in 1950, well beyond the computational capabilities of that time.

This model has never been "stress-tested" and could yield strange behaviors. However, playing with it also possibly serves to stimulate some thought about the ways in which it might be improved, although if you become interested in this possibility, you might find it even more interesting to consider the Godley-Lavoie models as a next study, inasmuch as these models have been developed through use.



	Y
1934	48.900
1935	53.300
1936	61.800
1937	65.000
1938	61.200
1939	68.400
1940	74.100
1941	85.300
1942	101.539
1943	114.834
1944	122.484
1945	125.201
1946	126.370
1947	126.956
1948	127.716

1 Bank Open | No Memory File Open | Dates: 1920-1948 | Frequency: A

Figure 69. Income Forecast

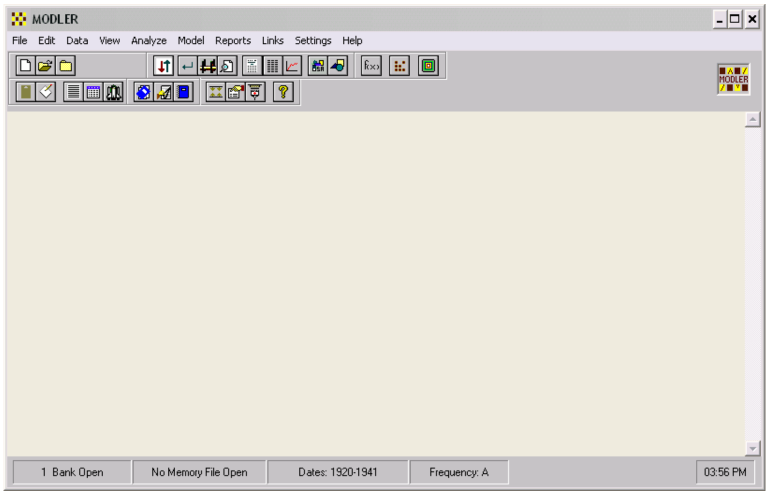
Chapter 4 Creating Presentations Using Model Results

In the course of the past three chapters, as a natural by-product of the process of building and solving Klein Model I, we have considered how to display the model itself. In particular, it has been shown that regression results can be captured as text and copied to word processing documents. Associated graphs can similarly be captured. In addition, text displays of model equations, either the complete model or some equation subset, can be copied and, in conjunction with variable glossaries created from data bank or Memory File indices, can be presented in the context of such documents. As a consequence, should you wish, it is possible to create a complete record of the estimation and construction of a model over any period of time. In addition, in the last Chapter we began to consider the use of the model, and in that context, we displayed both solution tables and graphs, even comparison tables and plots. However, so far, we have used a standard set of displays, meant essentially for the use of the model builder during the model construction and early evaluation phase.

In this chapter, in contrast, we will consider how to display the model and its results to a wider public, who in principle might be generally interested but who must be regarded as lacking the skill to interpret the model and its results in the way presented so far. Fortunately, MODLER has a number of facilities that will permit you to bridge this gap. You can, for example, generate either plain text or spreadsheet based tables that contain variable descriptions, rather than mnemonics, and exhibit specific titles, subtitles and other such features that make the presentation of the model results suitable for consumption by people who lack any knowledge of the particular characteristics of a model. Similarly, you can produce graphs and other displays that also no longer identify the variables by their mnemonics, but rather by a text description. These graphs can otherwise be enhanced so as to provide information that is not "model specific."

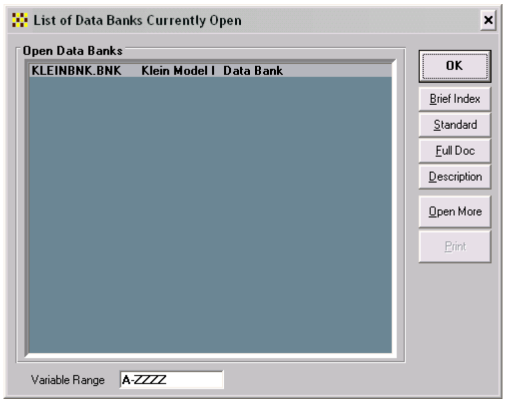
Presenting the Model Publicly

The variables that inhabit Klein Model I and the corresponding time series observations have been introduced during the process of examining how to estimate and form the model. However, in part because this model is small, the entire model glossary can be displayed easily. For example, if you click on the status bar of the Central Control Screen shown in Figure 70, specifically on the item that specifies that there is “1 Bank Open,” you will immediately see the screen shown as Figure 71.



1 Bank Open | No Memory File Open | Dates: 1920-1941 | Frequency: A | 03:56 PM

Figure 70. Central Control Screen, Displaying Status Bar

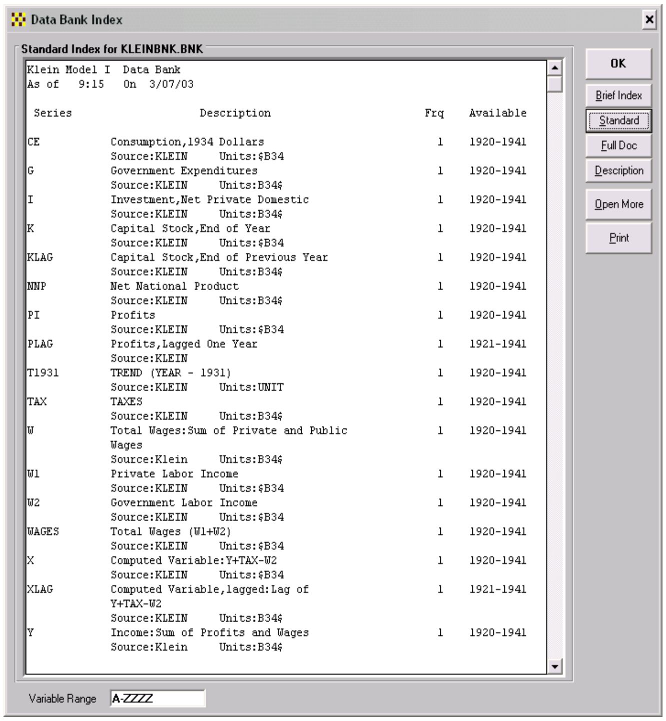


KLEINBNK.BNK Klein Model I Data Bank

Variable Range A-ZZZZ

Figure 71. List of Banks Currently Open

You will notice that this form has a series of buttons down its right-hand-side, and because there is only one bank open, Kleinbnk is already selected. Consequently, if you click on the button labeled **Standard**, the form shown in Figure 72 will immediately appear.



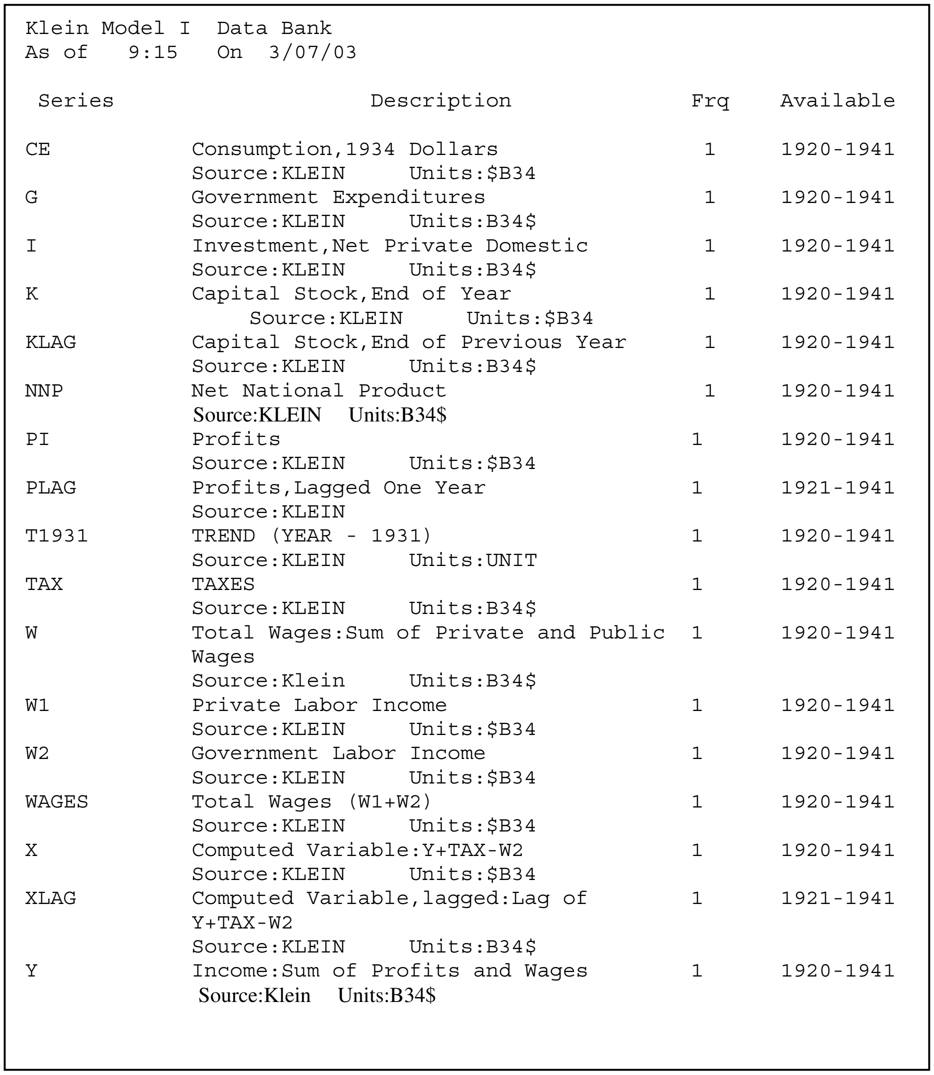
```
Klein Model I  Data Bank
As of   9:15   On  3/07/03

 Series                     Description                          Frq    Available

CE          Consumption,1934 Dollars                              1     1920-1941
            Source:KLEIN     Units:$B34
G           Government Expenditures                               1     1920-1941
            Source:KLEIN     Units:B34$
I           Investment,Net Private Domestic                       1     1920-1941
            Source:KLEIN     Units:B34$
K           Capital Stock,End of Year                             1     1920-1941
            Source:KLEIN     Units:$B34
KLAG        Capital Stock,End of Previous Year                    1     1920-1941
            Source:KLEIN     Units:B34$
NNP         Net National Product                                  1     1920-1941
            Source:KLEIN     Units:B34$
PI          Profits                                               1     1920-1941
            Source:KLEIN     Units:$B34
PLAG        Profits,Lagged One Year                               1     1921-1941
            Source:KLEIN
T1931       TREND (YEAR - 1931)                                   1     1920-1941
            Source:KLEIN     Units:UNIT
TAX         TAXES                                                 1     1920-1941
            Source:KLEIN     Units:B34$
W           Total Wages:Sum of Private and Public                 1     1920-1941
            Wages
            Source:Klein     Units:B34$
W1          Private Labor Income                                  1     1920-1941
            Source:KLEIN     Units:$B34
W2          Government Labor Income                               1     1920-1941
            Source:KLEIN     Units:$B34
WAGES       Total Wages (W1+W2)                                   1     1920-1941
            Source:KLEIN     Units:$B34
X           Computed Variable:Y+TAX-W2                            1     1920-1941
            Source:KLEIN     Units:$B34
XLAG        Computed Variable,lagged:Lag of                       1     1921-1941
            Y+TAX-W2
            Source:KLEIN     Units:B34$
Y           Income:Sum of Profits and Wages                       1     1920-1941
            Source:Klein     Units:B34$
```

Variable Range: A-ZZZZ

Figure 72. Data Bank Index as Model Glossary

If you should now choose to print this index, using the button labeled **Print**, you will then be offered the choice of printing it on a printer, assuming that you have a printer attached to your computer, or to a file. Should you choose the latter, the file generated will look like the listing shown in Figure 73. This file can be used as is or it can be edited. The degree to which it describes the variables in the model is of course a given in this particular case. However, in general, the degree of descriptiveness will depend upon the care taken by the person or persons who created the bank in first place.

Of course, should you wish, you can capture both the model equation display and the glossary listing and merge them into the same document. The effect will then be to provide a more or less complete description of the model, especially if you should also include a textual description in addition that explains aspects of the model.



```
Klein Model I  Data Bank
As of   9:15   On  3/07/03

 Series                     Description              Frq    Available

CE           Consumption,1934 Dollars                 1      1920-1941
             Source:KLEIN     Units:$B34
G            Government Expenditures                  1      1920-1941
             Source:KLEIN     Units:B34$
I            Investment,Net Private Domestic          1      1920-1941
             Source:KLEIN     Units:B34$
K            Capital Stock,End of Year                1      1920-1941
                 Source:KLEIN     Units:$B34
KLAG         Capital Stock,End of Previous Year       1      1920-1941
             Source:KLEIN     Units:B34$
NNP          Net National Product                     1      1920-1941
             Source:KLEIN    Units:B34$
PI           Profits                                 1      1920-1941
             Source:KLEIN     Units:$B34
PLAG         Profits,Lagged One Year                 1      1921-1941
             Source:KLEIN
T1931        TREND (YEAR - 1931)                     1      1920-1941
             Source:KLEIN     Units:UNIT
TAX          TAXES                                   1      1920-1941
             Source:KLEIN     Units:B34$
W            Total Wages:Sum of Private and Public   1      1920-1941
             Wages
             Source:Klein     Units:B34$
W1           Private Labor Income                    1      1920-1941
             Source:KLEIN     Units:$B34
W2           Government Labor Income                 1      1920-1941
             Source:KLEIN     Units:$B34
WAGES        Total Wages (W1+W2)                     1      1920-1941
             Source:KLEIN     Units:$B34
X            Computed Variable:Y+TAX-W2              1      1920-1941
             Source:KLEIN     Units:$B34
XLAG         Computed Variable,lagged:Lag of         1      1921-1941
             Y+TAX-W2
             Source:KLEIN     Units:B34$
Y            Income:Sum of Profits and Wages         1      1920-1941
              Source:Klein    Units:B34$
```

Figure 73. Model variable listing as a glossary

Creating and Editing Spreadsheet Tables

Another option you have is to produce a set of tables that are created in the context of your spreadsheet program, but are defined within MODLER. Initially, these tables must be produced line by line. However, this process can be recorded, and subsequently, a table of this type can be produced nearly automatically whenever you choose.

Now consider the solution manager screen as shown in Figure 74. Observe the main menu item **Reports**. If you click on this item, a dropdown menu will appear, like that shown in the figure. Notice specifically that this **Reports** menu item appears both on the Model Solution Manager screen and the Central Control Screen. Which of these you use is simply a question of the path you wish to take to reach a destination.

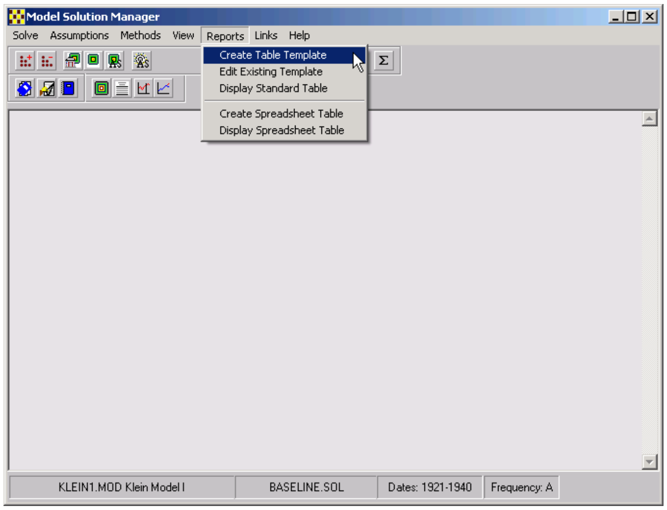
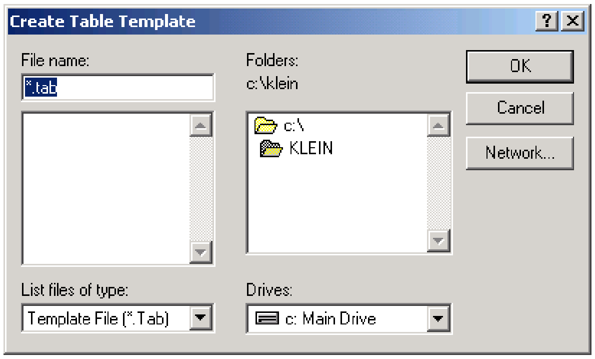


Figure 74. Creating a Report Table

The highlighted dropdown element is **Create Table Template** and if you click on this element, you will next see a form like that shown in Figure 75. Lets name this template First.Tab. The extent Tab implies that the file (in fact, a type of macro file) is to be interpreted as a Table Template file. Once you have named the template First.Tab and pressed the OK button of the form shown as Figure 75, you will see the MODLER Text and Data File Editor. Now, as it happens, at this point we are left to our own devices. Should you persevere, MODLER will expect you to fill this file with template commands, and it is possible that you are not intimately familiar with these.



Create Table Template

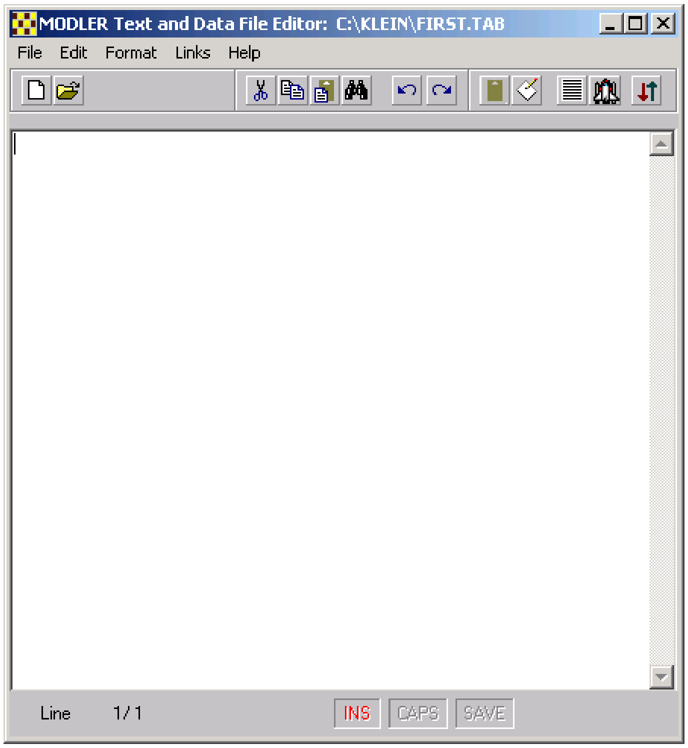
File name: *.tab

Folders: c:\klein

List files of type: Template File (*.Tab)

Drives: c: Main Drive

Figure 75. Naming the Table Template

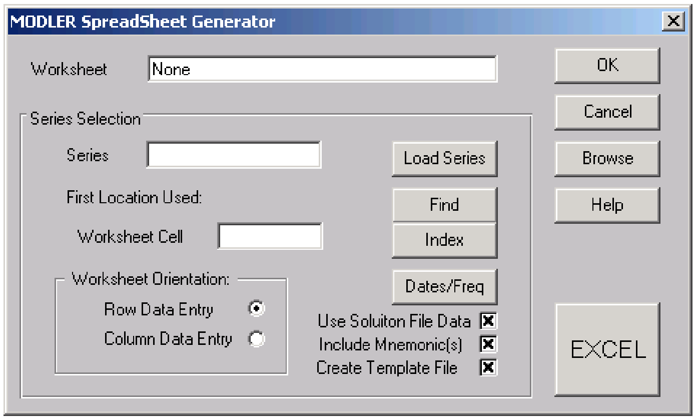


MODLER Text and Data File Editor: C:\KLEIN\FIRST.TAB

Line 1/ 1

Figure 76. Text and Data File Editor Screen

You could at this point drag this screen to the side, using your mouse, then click on the **Help** menu item on the MODLER Solution Manager Screen and choose first the help topic **Solution Facilities Outlined** and finally, at the bottom of that help screen, click on **Table Templates**. There you would find the necessary information to create a table template from scratch. But if you do this, you will have jumped in at the deep end. It might be best instead to backtrack, to close the Text and Data File Editor screen, click once more on the **Reports** menu item of the Solution Manager Screen and then choose the dropdown element second from the bottom, namely **Create Spreadsheet Table**. If you make this choice, you should then see the form shown in Figure 77.



MODLER SpreadSheet Generator

Worksheet: None

Series Selection

Series:

First Location Used:

Worksheet Cell:

Worksheet Orientation:

Row Data Entry ●

Column Data Entry ○

Use Soluiton File Data ☒

Include Mnemonic(s) ☒

Create Template File ☒

Figure 77. Creating a Spreadsheet Table Template

This form can be used to create a Table Template to be used specifically with your default spreadsheet program, which is here assumed to be Excel. Notice that, at the top, there is a text line, labeled *Worksheet*, which could be used to identify an existing worksheet. However, as none yet exists, ignore this line. Instead press the button that displays the name of the default spreadsheet, here the **EXCEL** button. Once you press this button, after a moment, the spreadsheet program should be executed. At this point your screen should display both a blank (Excel) worksheet and the MODLER Spread-Sheet Generator form, floating on top. Other MODLER screens may have vanished for the moment.

If the worksheet does not appear after a reasonable delay, press the (**EXCEL**) button again. Of course, you must previously have configured MODLER to execute a default spreadsheet program. Its possible that the spreadspread program will appear with its blotter blank. In this case, click on the **File** menu element of the spreadsheet program and then choose **New**. However, you are cautioned that you should not do much more than this with spreadsheet program itself. The process we are embarked upon, of using MODLER to write to another program, requires that Windows not become confused about which program is active at which moment. If it becomes confused, then the screens of both programs are likely to become frozen, until you are able to communicate to Windows how to proceed. General advice on how to act in this event cannot be given, inasmuch as the proper action depends on exactly what you might have done.

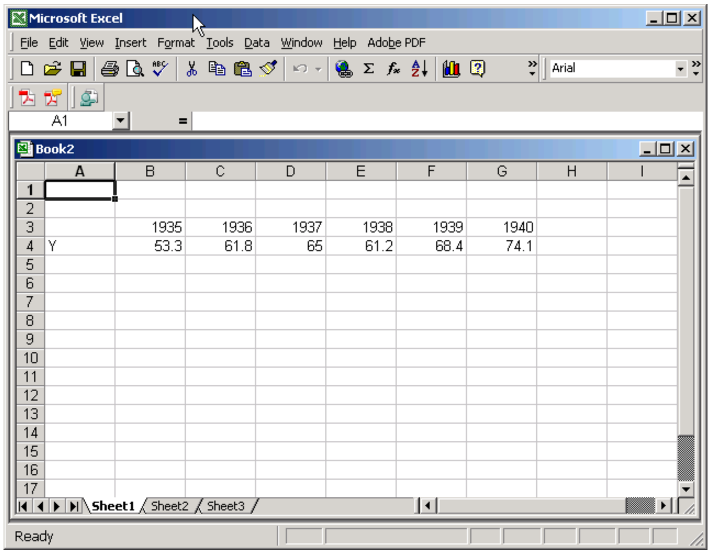
Look again at the spreadsheet generator form and observe that there is a checkbox (actually an x box) that is labeled *Use Solution File Data*. Leave this box as is – it should be checked. Notice also the checkbox labeled *Create Template File*, which should also be checked. Then click on the Dates/Freq button, and use the form that appears to set the date range as 1935-1940; the frequency should be Annual. It is important that you set the date range; otherwise, MODLER will take as a default whatever was previously set. Obviously, up to the column limit of your spreadsheet program (which may be 256), you are not restricted in the date range you set, provided that the first date is numerically greater than or equal to the first *historical* date for the solution file. In turn, the last date generally should not exceed the last solution file date. Next look at the *Worksheet Orientation* radio buttons. The default is Row Data Entry, which means that the table created will consist of horizontal rows, each data column corresponding to a particular year; the leftmost such column will be labeled using the first date of the date range you set.

Simply as an example, consider the model identity:

$$Y = CE + I + G - TAX$$

We might consider creating a table the first data row of which contains observations on Y, the next C, and so on. Notice the textbox labeled *Series* and the one labeled *Worksheet Cell*. One by one, we will enter the above variable names into the *Series* textbox. Once this has been done for a given name, say Y to start, we will then enter the Worksheet Cell specification corresponding to the leftmost column and row in which we wish the observations on that variable to appear. Actually, inasmuch as the checkbox labeled *Include Mnemonics* is checked, what will in fact happen is that MODLER will place the mnemonic Y in that leftmost row and column and in the successive columns of that row put the observations on Y for the years 1935-1940.

For example, enter A4 in the *Worksheet Cell* textbox, having put Y into the *Series* textbox. Then press the **LOAD SERIES** button. Once you have done this, you should observe that a row of values appears in the fourth row of the spreadsheet worksheet. Next, put the word Dates into the *Series* textbox and A3 in the *Worksheet Cell* textbox. When you have done this, your worksheet should generally look like that shown in Figure 78.


Microsoft Excel
File Edit View Insert Format Tools Data Window Help Adobe PDF
Arial
A1
=
Book2
Sheet1
Sheet2
Sheet3
Ready

	A	B	C	D	E	F	G	H	I
1									
2									
3		1935	1936	1937	1938	1939	1940		
4	Y	53.3	61.8	65	61.2	68.4	74.1		

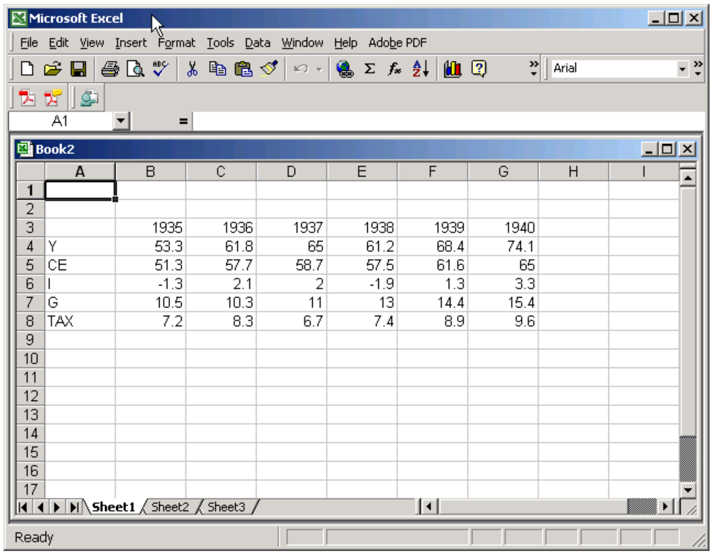
Figure 78. Solution File Report, First Stage

As you may have surmised, “Dates” is a report keyword that generates a series of dates to be used for display. You will see from this example that the combination of having specified the initial cell location in each case together with having previously set the date range to be 1935-1941 is sufficient to tell MODLER how to populate the worksheet so far.

Next we can select CE as the series and locate it as beginning in the cell A5. Alternatively, it would be possible to first remove the check from the *Include Mnemonic(s)* checkbox and to specify the first cell location as B5. In this case, the variable values for CE will appear, but without the series name CE. Separately, we can fill in the leftmost worksheet column with the words, Consumption, Investment, Government Expenditures, and so on, telling MODLER to locate the values in rows beginning with the second column of the worksheet. However, *it is important not to attempt to write directly to the worksheet during the attempt to fill in the variable values* using MODLER. As mentioned earlier, if you do this, Excel and MODLER are likely to freeze, inasmuch as Windows may become confused as to which of these two programs should be treated as active. Notice that we have not yet saved or otherwise identified the worksheet, so that presently everything is somewhat tentative. In this situation, it is better to restrict ourselves strictly to using the MODLER spreadsheet generator form. Later, the worksheet can be elaborated, once it has been defined.

At first, it is helpful to use the mouse to select first the series textbook, then insert a series name (mnemonic), and next click on the worksheet cell text box, specify the cell value, and then click on the **Load Series** button. However, you might be interested to know that MODLER is set up so that once you have entered the first series name, if you then press the **Enter** key, control next jumps to the worksheet cell textbox. If you enter a value and then press the **Enter** key, control jumps to the **Load Series** button. If you again press the **Enter** key, control jumps *back* to the Series textbox. Consequently, it is possible to enter each relevant value and then use the **Enter** key, with the result that you need not raise your hands from the keyboard. It is therefore possible to create an extensive table rather easily.

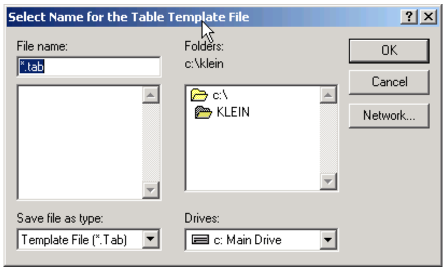
Next, lets enter CE as the series name and locate it as cell A5. Then enter I and located it as cell A6, and so on, entering then G and TAX as cells A7 and A8 respectively. The final result should be that shown in Figure 79. Your screen may differ from the display shown in this figure in certain ways. The number of blank cells displayed may be different and the workbook form may not be entitled *Book2*. What matters is the table entries themselves. Any other differences are inessential.


Microsoft Excel
File Edit View Insert Format Tools Data Window Help Adobe PDF
Arial
A1
=
Book2
Sheet1
Sheet2
Sheet3
Ready

	A	B	C	D	E	F	G	H	I
1									
2									
3		1935	1936	1937	1938	1939	1940		
4	Y	53.3	61.8	65	61.2	68.4	74.1		
5	CE	51.3	57.7	58.7	57.5	61.6	65		
6	I	-1.3	2.1	2	-1.9	1.3	3.3		
7	G	10.5	10.3	11	13	14.4	15.4		
8	TAX	7.2	8.3	6.7	7.4	8.9	9.6		

Figure 79. Final Table, as entered

Next, press the OK button on the spreadsheet generator form. You should then immediately see the form displayed in Figure 80, which will permit you to name the table template file. Notice that it has default extent .TAB. Lets enter into the Filename textbox of this form the word FIRST. You neither need to capitalize this name, as done here, nor include the extent .TAB. So long as you do not provide another extent, which you generally should not, MODLER will automatically add the extent .TAB.



File name: *.tab

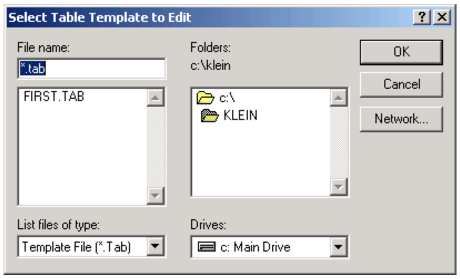
Folders: c:\klein

Save file as type: Template File (*.Tab)

Drives: c: Main Drive

Figure 80. Naming the Table Template File

Once you have entered the template name and pressed the **OK** button on this form, the template will be both created and named. Next look once again at the solution manager form. Notice not only the **Reports** menu item, but also the icons at the far left of the second icon row. The leftmost of these, as you can determine by moving your mouse over it, is captioned *Create Report*, short for Create Spreadsheet Report. Obviously, we could have initially used this icon, rather than the **Reports** menu item, to instigate the creation of the table template we just made. Move your mouse next to the icon just to the right, which should expose the caption *Edit Table Template*. Press this icon button. You should then see the form shown in Figure 81.



File name: *.tab

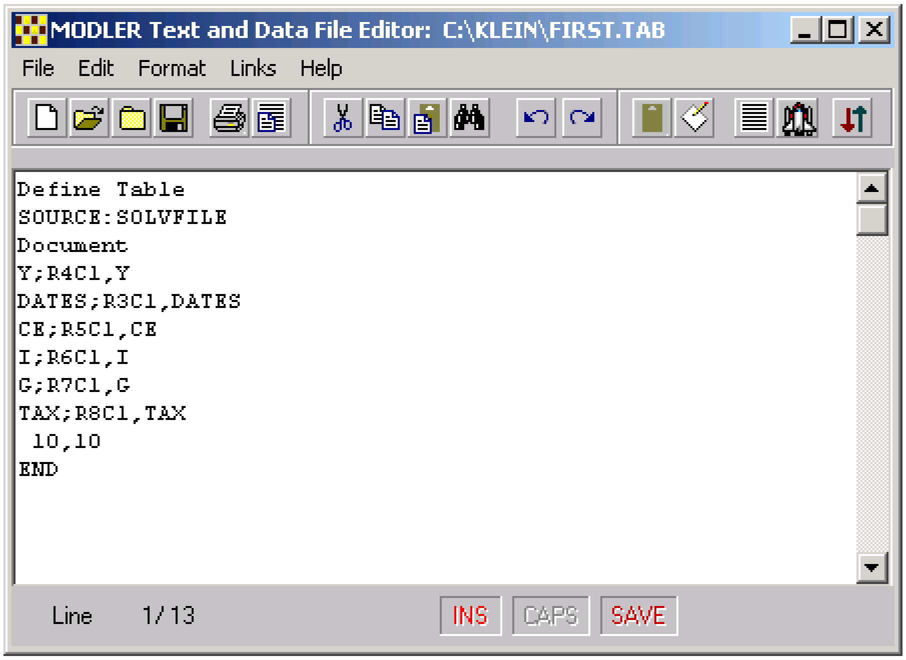
Folders: c:\klein

List files of type: Template File (*.Tab)

Drives: c: Main Drive

Figure 81. Selecting a Table Template to Edit.

Select FIRST.TAB and then press the OK button on this form. You should then see the screen displayed in Figure 82, which is obviously the text and data file editor screen. The file display, as is indicated by the caption line of this form is FIRST.TAB, which in this case happens to be located in the directory named KLEIN. This file contain 11 lines of text, beginning with the line Define Table and ending with the word END. Evidently, a table template should begin with the entry Define Table and end with the word END. Here, there happens to be, in addition, two blank lines, which explains the status bar entry 1/13..


MODLER Text and Data File Editor: C:\KLEIN\FIRST.TAB
File Edit Format Links Help
Line 1/ 13
INS
CAPS
SAVE

```
Define Table
SOURCE:SOLVFILE
Document
Y;R4C1,Y
DATES;R3C1,DATES
CE;R5C1,CE
I;R6C1,I
G;R7C1,G
TAX;R8C1,TAX
 10,10
END
```

Figure 82. First Table Template

This table template is significant for what entries do not appear in it. First, notice that there any date range entry is missing. Next, observe that the line SOURCE:SOLVFILE does not reference a specific Solvfile. The implications are that, to use this template, we need to have previously attached a particular solution file (as well as a model, of course). We also need to have specified a date range. At the moment, we have focused on a single solution file, but recall that a model can be associated with any number of solution files, so that our table template can be used with any solution file we create for the particular model Klein1. Similarly, if we should make a number of model solutions, for various date ranges, the template can be used with any of these. If you dig deeply into the MODLER helpfile, you will discover that a template file can be specialized to a particular solution file and a particular date range, which has the benefit of specialization, but in many case generality is a virture, just so long as you remember to set the dates.

Editing the table template explicitly is also not the only option, but as we will see, it may be the best option. The first three lines of this table template ordinarily should be left as they are; the order of these should not be changed. The Document command defines where the commands that specify the rows of the table begin. The order of the next 6 lines, you may recall, reflect the order in which we entered commands into the spreadsheet generator form. We began with Y and ended with TAX. Second, we entered Dates – which here has been capitalized by default, inasmuch as it is a special keyword in a table template file.

However, you may also notice that we entered the cell references as A4,A3, and so on, whereas the table template has entries that display the entries RmCn, where m and n are integers. These are cell references. They are interpreted as referring to row m and column n. Once of the reasons for this change is that Excel itself internally uses this type of cell reference. The A4,A3 form is used by that spreadsheet package as a human interface default (although it optionally permits the use of this RmCn type of reference), but internal to the table template it is generally most useful to use the RmCn type.

Because each of the variable line entries are individually keyed to a particular table row (as well as an originating column), when editing the table template we have the freedom to transpose these. For instance, we can transpose the Y and DATES lines, we might be useful inasmuch as this will instantly tell us, when looking at the table template, that this first line of the body of the table consists of dates. However, if done judiciously, we can also change the RmCn values themselves. If we no longer wish the table to be fully left justified in the spreadsheet worksheet, we could for instance change the C1 in each case to C4, which would cause the table to begin in the fourth column. We can also open up row space in the table: notice that only the rows of the spreadsheet that are specifically referred to in the table template are written to, implying that we can leave gaps between the rows. This can be useful, as well will discover later, should we wish to use the table template together with an existing named worksheet that might have certain description rows that we do not wish to overwrite. For the time being, incidentally, ignore the line 10,10 that appears just before the END line – simply treat these two lines as both required at the end of a table template.

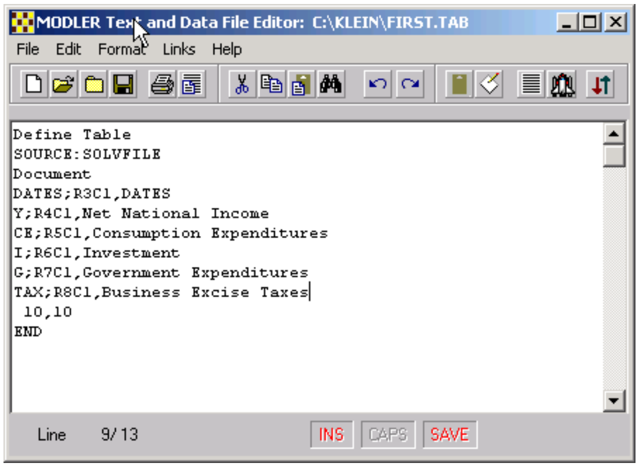
A further change that we can make is to change the *trailing* mnemonics in each of the table row entry lines – with the exception of the DATES entry, which for the moment at least should be left as it is. For instance, in the case of the line:

```
Y;R4C1,Y
```

we can replace the trailing Y with Net National Income:

```
Y;R4C1,Net National Income
```

And so on in the case of the other *trailing* mnemonics. What we must not change is the beginning mnemonic in each of these row entry lines. The purpose of this mnemonic is to tell MODLER the series to be retrieved from the solution file. Any changes made to these mnemonics could have a catastrophic effect on the table template. Consider therefore the edited table template shown in Figure 83.



```
Define Table
SOURCE:SOLVFILE
Document
DATES;R3C1,DATES
Y;R4C1,Net National Income
CE;R5C1,Consumption Expenditures
I;R6C1,Investment
G;R7C1,Government Expenditures
TAX;R8C1,Business Excise Taxes
 10,10
END
```

Figure 83. Edited Table Template

Now we can save our work, by first clicking on the **File** menu element of this text and file editor screen and then selecting **Save As…**

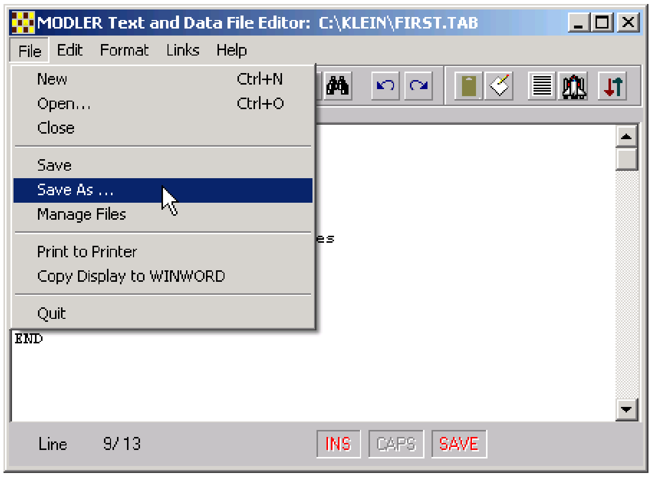
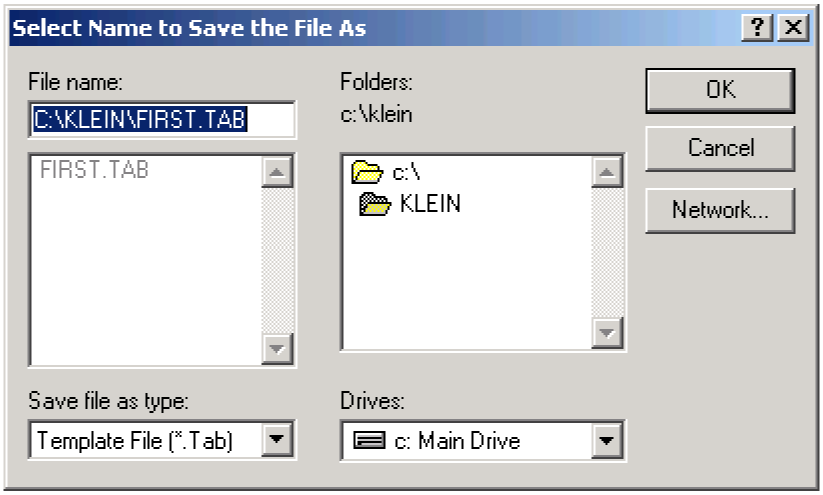


Figure 84. Saving the Edited Template

Just in case we made some mistake, it might be best to leave the original table template as it was and to name this second template *Second*, using the form shown as Figure 85.



Select Name to Save the File As

File name: C:\KLEIN\FIRST.TAB

FIRST.TAB

Folders: c:\klein

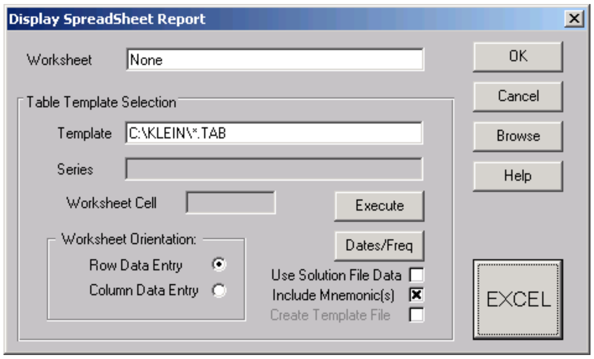
c:\
KLEIN

Save file as type: Template File (*.Tab)

Drives: c: Main Drive

Figure 85. Naming the Edited Template File

Then we can exit from the text and data file editor screen, which will return us to the solution manager screen. There we can either click on the third icon from the let in the second icon row or choose the **Reports** menu item and then select the bottom most element of the dropdown list, namely **Display Spreadsheet Table**. The effect of either of these actions will be to display the form shown as Figure 86.



Display SpreadSheet Report

Worksheet: None

Table Template Selection

Template: C:\KLEIN\*.TAB

Series:

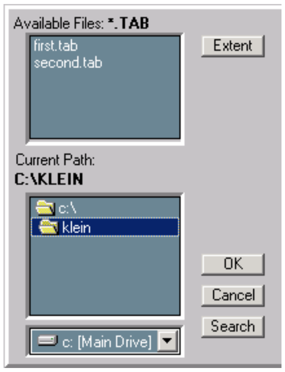
Worksheet Cell:

Worksheet Orientation:
Row Data Entry ●
Column Data Entry ○

Use Solution File Data □
Include Mnemonic(s) ☒
Create Template File □

Figure 86. Spreadsheet Table Display Form

Possibly the best first action to now take is to highlight the entry of the textbox entitled *Template*. Then click the **Browse** button. You should then see the form that appears as Figure 87. The highlighting, then clicking the Browse button, identifies which type of file we will be browsing for. Notice the alternative worksheet textbox, which should be ignored for the time being. Using the form shown in Figure 87, double click on second.tab. The result should be to transfer this tabfile name to the template textbox.



Available Files: *.TAB

first.tab
second.tab

Current Path:
C:\KLEIN

Figure 87. Browse Tab Files

Next click on the Spreadsheet button (here EXCEL), which should have the effect of once again executing the spreadsheet program. You may need to click this button twice and also click the File main menu element of the spreadsheet program and then choose New.

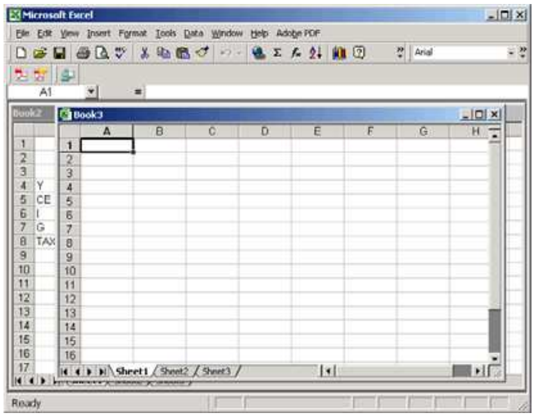
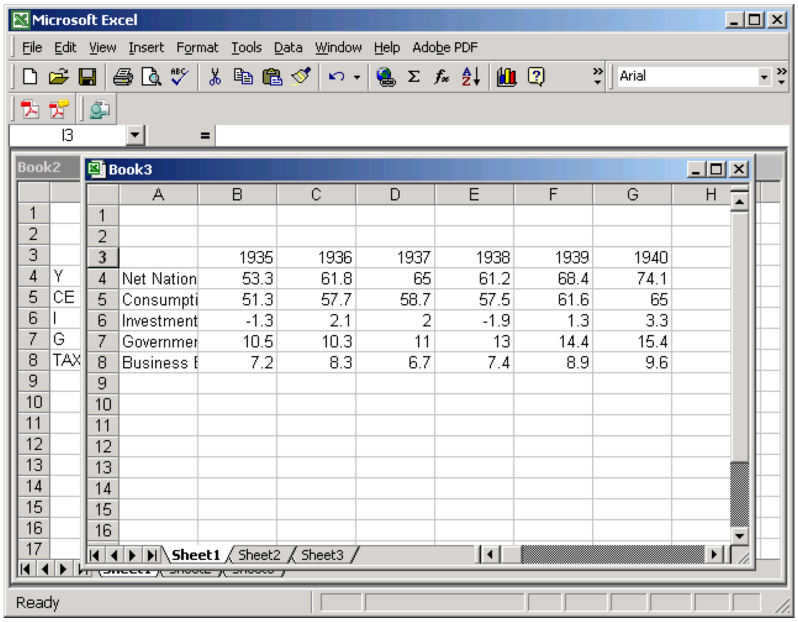


Figure 88. Initial Spreadsheet Display

Notice next, looking back at Figure 86, that the check box Use Solution File Data is *not* checked. The Create Template File checkbox is grayed out, and therefore not enabled. At this stage, MODLER does not “know” that we might use a solution file. You can, if you wish, check the Use Solution File Data check box, but this is not necessary. However, it is *very important* now to press the **Dates/Freq** button and to set the dates properly; you may as well choose the date range 1935-1940. If you do not set the dates, the wrong dates may be used; the effect may be that the execution of the template aborts. But assuming that the dates are now set, the next step is simply to press the **Execute** button. You should see a display similar to that shown in Figure 89.


Microsoft Excel
File Edit View Insert Format Tools Data Window Help Adobe PDF
Arial
I3
=
Book2
Book3
Sheet1 Sheet2 Sheet3
Ready

Book2:

4	Y
5	CE
6	I
7	G
8	TAX

Book3:

	A	B	C	D	E	F	G	H
3		1935	1936	1937	1938	1939	1940	
4	Net Nation	53.3	61.8	65	61.2	68.4	74.1	
5	Consumpti	51.3	57.7	58.7	57.5	61.6	65	
6	Investment	-1.3	2.1	2	-1.9	1.3	3.3	
7	Governmer	10.5	10.3	11	13	14.4	15.4	
8	Business E	7.2	8.3	6.7	7.4	8.9	9.6	

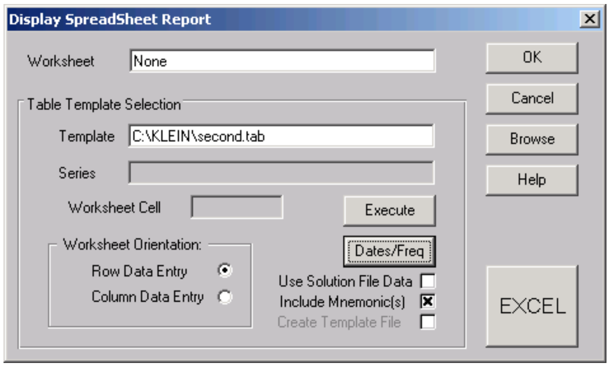
Figure 89. Execution of the Edited Template

At this stage, as shown, the series descriptions we entered are partially obscured. Separately from MODLER, you will now need to use your mouse to increase the width of the first spreadsheet column. Alternatively, you can format this column, using the facilities of the spreadsheet program, so that it is wide enough to display the series descriptions entries. Should you wish, you can now edit this table in various ways, adding headlines, providing emphasis, and the like.

Using a Predefined Workbook/Worksheet

Now let us consider putting together a previously formatted spreadsheet workbook/worksheet and a MODLER table template. An aspect of this that must be considered carefully is that you must match the template and the predefine workbook/worksheet. In particular, the rows that the template intends to write to must be rows that in the spreadsheet workbook/worksheet are either blank or expected to be overwritten.

Starting by clicking the Display Spreadsheet Table icon, once again the form shown in Figure 90 will be displayed. You will know already how to browse for or otherwise specify the name and location of the MODLER table template used.



Display SpreadSheet Report

Worksheet: None

Table Template Selection

Template: C:\KLEIN\second.tab

Series:

Worksheet Cell:

Worksheet Orientation:

Row Data Entry ●

Column Data Entry ○

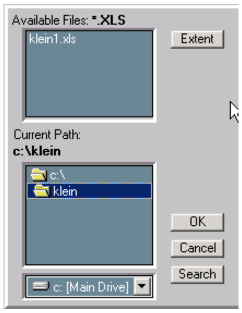
Use Solution File Data □

Include Mnemonic(s) ☒

Create Template File □

Figure 90. Display Spreadsheet Report Form

Therefore click on, or otherwise highlight the entry (None) in the Worksheet texbox and then press the **Browse** button on this form causing the search form in Figure 91 to appear.



Available Files: *.XLS

klein1.xls

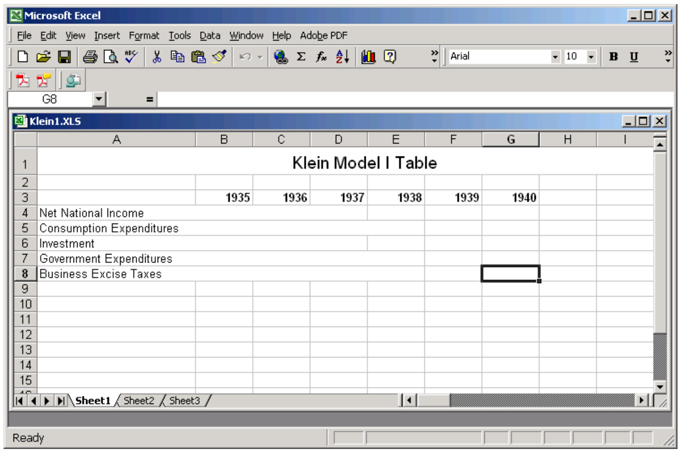
Current Path:
c:\klein

c:\
klein

c: [Main Drive]

Figure 91. Search Form

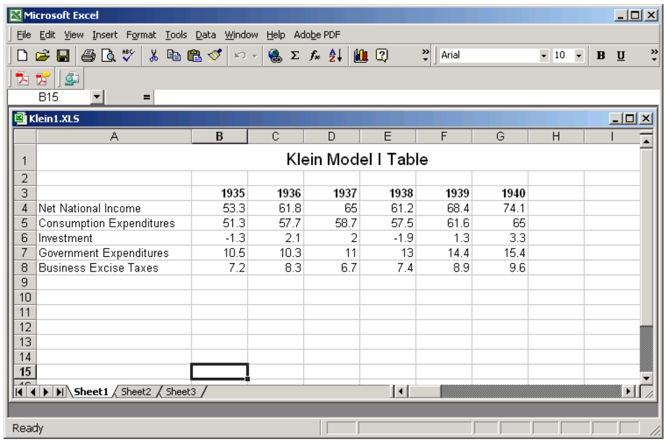
Doubleclick on the name of the Workbook file (here Klein1.XLS), which will transfer this name (and its path) to the Worksheet text box. Then click on the Dates/Freq button and establish the date range for the table, here assumed to be 1935-1940. Press the spreadsheet execution button (here EXCEL), which should cause the named workbook to be displayed, as shown in Figure 92. Notice that a general title has been included, in a larger typesize and using a bold font. The dates are also in a bold font – but as it happens these will be overwritten. So also will be the series descriptions. These "sacrificial" table elements are included here simply as a way of defining the principal elements of the table in advance. They could have been omitted. Of course, you will need to create this "Blank" named workbook yourself.


Microsoft Excel
File Edit View Insert Format Tools Data Window Help Adobe PDF
Arial
10
B
U
G8
=
Klein1.XLS
Sheet1
Sheet2
Sheet3
Ready

	A	B	C	D	E	F	G	H	I
1	Klein Model I Table								
2									
3		1935	1936	1937	1938	1939	1940		
4	Net National Income								
5	Consumption Expenditures								
6	Investment								
7	Government Expenditures								
8	Business Excise Taxes								
9									
10									
11									
12									
13									
14									
15									

Figure 92. "Blank" Named Workbook

Now simply press the Execute button on the Display Spreadsheet Repot form (Figure 90). What should be displayed next is the filled in workbook table shown in Figure 93. You will of course be able to see this filling occur. It will be obvious for the series values that a put into blank cells. It will not be obvious for the Dates (except if you should change these to, for instance, 1933-1938), nor for the series descriptions. Provided that you have formatted the cells of the workbook in advance, as MODLER writes the entries, they will be written so as to conform with your formatting. You have the ability to control which fonts are used and what the table generally looks like. You can, for instance, include horizontal separator lines and cause certain data columns to be displayed in an italic font. You can include footnotes. The only thing that you need to bear in mind, in particular, is not to create conflicts. MODLER limits what it writes to specific cells in the spreadsheet table, as controlled by the table template file.


Microsoft Excel
File Edit View Insert Format Tools Data Window Help Adobe PDF
Arial
10
B15
=
Klein1.XLS
Sheet1
Sheet2
Sheet3
Ready

Klein Model I Table

	1935	1936	1937	1938	1939	1940
Net National Income	53.3	61.8	65	61.2	68.4	74.1
Consumption Expenditures	51.3	57.7	58.7	57.5	61.6	65
Investment	-1.3	2.1	2	-1.9	1.3	3.3
Government Expenditures	10.5	10.3	11	13	14.4	15.4
Business Excise Taxes	7.2	8.3	6.7	7.4	8.9	9.6

Figure 93. Filled in Workbook Table

Appendix Using Macros to Build the Model

The description of the construction of Klein Model I in the foregoing chapters of this workbook is intended to provide a guide to building and using a model step-by-step. It is therefore oriented towards the interactive use of MODLER. However, MODLER also permits a model to be built using macro files, that is files that in this context contain “model building” commands. The principal reason why you might consider using a macro file, instead of operating the program interactively, is that macro files are replicable. Once commands have been included in such a file, it can be saved and used time and time again. Such files provide you with the capability not only to replicate operations, but thereby also to create a complete audit trail of what you have done. When you create or use a model interactively, you not only need to re-type (or re-select from a menu) commands you have used previously, you also may not be able to determine when and where you made any particular mistake that you might have made. The ability to replicate, with or without subsequent, documented changes, is a very powerful MODLER feature.

The technical note at the front of this workbook lists the files that can be used to create the Klein Model I using MODLER. Among these are three files that have the MAC extent; these are the macro files that you can start with. One of these is called KIDENT.MAC, another is called ESTMOD.MAC, and the third is called TSKLEIN.MAC. These are respectively displayed below as Figures A1, A2, and A5.

However, before considering these particular files, it may be helpful to pause first and consider what is involved when a model is created using the MODLER model definition command. The original creation of the Model File, as described in Chapter 1, involves either issuing the command:

```
DEFINE MODEL KLEIN1
```

or selecting the appropriate menu file item, **Create New Model**. In either case, the result is the creation of a MOD (model) file. You will recall that this command was issued in Chapter 1 as a separate command, after which other commands were individually issued that together caused the model to be formed.

As an alternative, it is possible to create a model entirely from within a macro, embedding the DEFINE MODEL command in that macro. This is the approach that will be taken here. Considering the effect of this command, on its own, bear in mind that whichever way in which you create a model, once you have named it, possibly described it, and established the observation frequency of its data, if that is all you have done, then the Model File that is created on your hard disk will be in effect an empty box, simply a container for a model that has yet to be formed. The definition of the model just causes the creation of a file to contain the model. As discussed in Chapter I, it is the next step, that of putting one or more equations in the model, that actually results in the creation of a model that you can display and work with.

This next step is not, however, rigidly defined procedurally. In fact, in terms of the model building process, you have a number of alternative choices at this stage, and which of these you should make is in part simply a matter of what you wish to do. For instance, you can begin by first entering the model's identities. Alternatively, you can estimate parameters of the behavioral equations first, adding them each to the model, then include the identities. Or you can include the various equations in some mixed order. As a piece of software, MODLER does not care which of these options you choose. And insofar as the final model is concerned, provided that the equations are properly specified as you go, each of these choices can lead to the creation of exactly the same functional model.

The only problem is that mistakes are easily made, and thus it may matter which choice you make: there is an issue of best practice. Generally, it is best to create the model identities separately, which can mean creating these first. The reason is that model specification errors can be made at any stage, and you may later wish to test against this possibility in order to verify that you have correctly specified at least the identities. This testing can be done by creating a submodel, with its own individual name, that contains nothing but identities, then solving this as a separate model and comparing the results to the data in your data bank. A characteristic of identities is that they do not (should not) involve any residual errors. If you compare the solution results for a model that contains only identities and if you observe errors, ordinarily this will imply that you have misspecified one or more of them. Thus at some stage, you are likely to wish to be able to collect all a model's identities together, so you may as well have done this at the beginning, by creating a macro file containing nothing but identities. Alternatively, if you do not know in advance what identities your final model will contain, you can create a macro file on the side as you estimate behavioral equations, which can then be used to create an identities submodel or add the identities to the main model. Remember the remark made in Chapter 2 that a model can be created by collecting together the equations of models created earlier, thus allowing you to test a model part by part as it is formed.

The Klein Model I identities are displayed in Figure A1, which displays the contents of KIDENT.MAC. These three identities specify respectively National Product net of Tax, the growth in the capital stock as the sum of net investment this period plus the capital stock at the end of the previous period and PI as Net National Product less wages. Note that they are preceded by the DEFINE MODEL KLEIN1, DESC:Klein

Model I, and FREQ=ANNUAL commands, and followed by the SAVE command. Of these the DESC: command is optional, but it is often useful for the model description to be displayed whenever the model equations are.

```
DEFINE MODEL KLEIN1
DESC:Klein Model I
FREQ=ANNUAL
Y=CE+I+G-TAX
PI=Y-(W1+W2)
K=I+K(-1)
SAVE
```

Figure A1. KIDENT.MAC File

As a process, these identities can be loaded into a Model File named KLEIN1.MOD by simply executing the command:

RUN KIDENT

which has the effect of executing the macro that consists of the commands shown in Figure A1. The initial DEFINE MODEL command causes the MOD file named KLEIN1.MOD to be created, and the SAVE command at the end causes the model file to be saved with the three equations included. The FREQ command causes the model to be saved as an annual frequency model. The commands between specify the identities in the model. The RUN command can be typed into the blotter of the Central Control Screen. Alternatively, under the **File** main menu item on either this screen or the Model Builder and Equation Editor Screen, you will find the dropdown menu element **Run a MODLER Macro**, which if selected will permit you to execute KIDENT.MAC.

You can at this stage then display the model, if you wish, using the **View** model commands associated with the Model Building and Editing screen. At this point, you can also compile the model, if you wish. From a human perspective, the model as yet is nothing but a set of identities, but from the point of view of MODLER, any set of equations is a model and can be solved, at least in principal. The point in compiling the model now would be to allow you to create a test model and a solution file, which can be used to verify that the identities are properly specified, when compared to the observations in the data bank.

This is not to say that the Klein Model identities are in any ultimate and final sense correct. The current US National Income and Product Accounts do not define the relationships among the variables as did either Klein in 1950 or the early US National Income and Product Accounts, as defined then. However, the data for the Klein model *are* constructed in conformity with the Klein Model identities, and it is this conformity that would be tested. Similarly, any other model you might build would involve the question whether the identities you have specified agree with the accounting identities implicit in the data base you use, hence the data in the data bank. Incidentally, note that the macro shown in Figure A2 does not specify any data bank. The process of specifying the identities of a model consists simply of first defining the model and then stating the identities. No data are required.

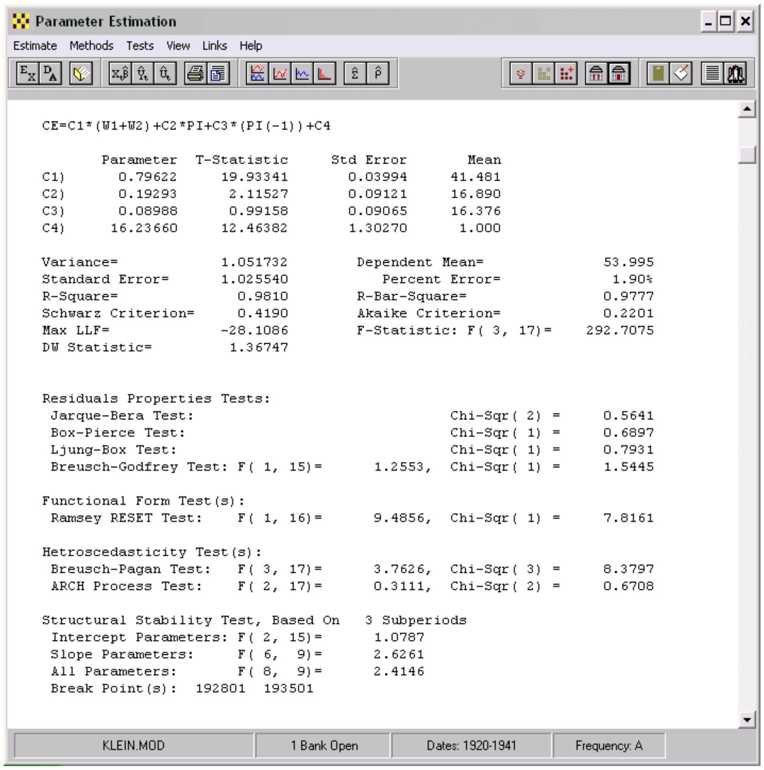
In contrast, in order to estimate the parameters of the behavioral equations, it is necessary to access a data bank, as well as to specify the observation frequency of the data. Note also that although it is usual for the frequency of the model to be the same as the estimation frequency of the behavioral equations, MODLER does not require this. Instead, even if the frequencies are the same, MODLER requires that you specify these separately. The macro ESTMOD.MAC shown in Figure A2 begins with commands that respectively access the data bank, KLEINBNK, and set the frequency and specify the sample period date range. This is the estimation frequency. The macro KIDENT.MAC shown in Figure A1 also specifies an observation frequency. This is the observation frequency of the model, and specifies the observation frequency of the data to be used with the model. The use of the semi-colon as the first character on a line permits a notation to be included in the macro; when the macro is executed, any such line is ignored.

```
; Estimation of Behavioral Equations Using a Macro
ACC KLEINBNK
SET FREQ=ANN
SET DATES=21-41
CE=F(PI,W1+W2,PI(-1))
PUTEQ
I=F(PI,PI(-1),K(-1))
PUTEQ
W1=F(Y+TAX-W2,Y(-1)+TAX(-1)-W2(-1),T1931)
PUTEQ
```

Figure A2. ESTMOD.MAC file

Observe in Figure A2 that the three regression commands that specify the estimated, behavioral equations for the model are each followed, on the next line, by the command PUTEQ. In each case, the regression command causes the estimation to be performed. The PUTEQ command in turn causes that equation, as estimated, to be included in the model. As noted in Chapter I, this process is called autocoding and the result is that the estimated equation is inserted into the model automatically, without any need for you to code estimate equations separately. Among other things, autocoding prevents mistakes being made in the coding of the behavioral model equations. Note, however, that the PUTEQ command does presuppose that a model is attached at the time this command is issued—which will be the case here inasmuch as the KIDENT.MAC macro is presumed to have been run just prior to ESTMOD.MAC.

When you run ESTMOD, incidentally, you should observe as output a screen display similar to that shown in Figure A3.



```
CE=C1*(W1+W2)+C2*PI+C3*(PI(-1))+C4

       Parameter  T-Statistic     Std Error       Mean
C1)      0.79622     19.93341       0.03994     41.481
C2)      0.19293      2.11527       0.09121     16.890
C3)      0.08988      0.99158       0.09065     16.376
C4)     16.23660     12.46382       1.30270      1.000

Variance=            1.051732        Dependent Mean=             53.995
Standard Error=      1.025540           Percent Error=            1.90%
R-Square=              0.9810        R-Bar-Square=               0.9777
Schwarz Criterion=     0.4190        Akaike Criterion=           0.2201
Max LLF=             -28.1086        F-Statistic: F( 3, 17)=   292.7075
DW Statistic=         1.36747


Residuals Properties Tests:
 Jarque-Bera Test:                                Chi-Sqr( 2) =     0.5641
 Box-Pierce Test:                                 Chi-Sqr( 1) =     0.6897
 Ljung-Box Test:                                  Chi-Sqr( 1) =     0.7931
 Breusch-Godfrey Test: F( 1, 15)=      1.2553,   Chi-Sqr( 1) =     1.5445

Functional Form Test(s):
 Ramsey RESET Test:    F( 1, 16)=      9.4856,   Chi-Sqr( 1) =     7.8161

Hetroscedasticity Test(s):
 Breusch-Pagan Test:   F( 3, 17)=      3.7626,   Chi-Sqr( 3) =     8.3797
 ARCH Process Test:    F( 2, 17)=      0.3111,   Chi-Sqr( 2) =     0.6708

Structural Stability Test, Based On   3 Subperiods
 Intercept Parameters: F( 2, 15)=      1.0787
 Slope Parameters:     F( 6,  9)=      2.6261
 All Parameters:       F( 8,  9)=      2.4146
 Break Point(s):  192801  193501
```

Figure A3. Scrolling Parameter Estimation Display

Note the scroll bar down the right-hand-side of this form. On your screen you will be able to view each of the estimated equations in turn by scrolling this display. In terms of the layout and presentation, each of them looks very much like what is shown in this figure, although of course the specific variables and estimated parameter values differ. You have the choice of simply reviewing this "printout," or you can capture these results, save them to a file, print them to a printer, or copy them to your word processor, in the process providing full documentation of this model estimation step.

As indicated earlier, this process of first running the KIDENT macro and then ESTMOD is purely by choice. There is no requirement that the two macros be run separately. Alternatively, it is perfectly permissible to create a single, combined macro that concatenates the contents of KIDENT.MAC with that of ESTMOD.MAC. The only reason not to do this is either for simplicity's sake, or else to allow the model's identities to be treated and tested separately, as discussed earlier. In any case, once the commands contained in the two macros have been executed, the whole of Klein Model I will have been created. Moreover, what is created is exactly the same as if a composite macro were used. If, at this stage, you view the complete set of equations in the model, they should look like the set of equations shown in Figure A4.

A warning is, however, in order concerning the way you might use the ESTMOD macro. Notice that the PUTEQ commands used in this macro do not explicitly state where the estimated equations are to be put in the model. The implication of this generality is that MODLER will successively append each estimated equation to the existing model. Among other things, this successive appending means that if you were to run ESTMOD twice the result would be that the model would then contain 9 equations, with each of the estimated equations duplicated in the form of two sets of estimated equations. Other forms of the PUTEQ command can be used that would keep this duplication from occurring, but the most important point is that when you use macros you should treat them with respect, and recognize that each time a macro is run something is likely to happen, whether you might intend for it to or not.

Assuming that you have executed the two macros, Klein Model I is now ready to be compiled, just as was done in Chapter 2. In fact, if you wish, you could add the compile command:

```
COMPILE
```

to the end of the ESTMOD macro, which would result in the model being both estimated and compiled. It is possible also to add further commands, such as those to create a solution file and even to solve the model. Plot and table display commands cannot generally be used in a model building macro, but the other commands associated with the creation and use of a model can be. This capability is demonstrated in the MODLER workbook, *Building and Using Theoretically Defined Operative Economic Models*, found on the Learning Tools page of www.modler.com.

```
  Klein Model I

Equation Number   1:   NNP

Y=CE+I+G-TAX

Equation Number   2:   K

K=I+K(-1)

Equation Number   3:   PI

PI=Y-W1+W2

Equation Number   4:   CE

CE=0.19293*PI+0.79622*(W1+W2)+0.08988*(PI(-1))+16.23660
    (2.12)     (19.93)          (0.99)           (12.46)

          SEE=     1.025540   R-Bar-Sq= 0.9777   DW=   1.367
          Ordinary Least Squares
          Sample Period=   1921-1941

Equation Number   5:   I

I= 0.47964*PI+0.33304*(PI(-1))-0.11179*(K(-1))+10.12579
    (4.94)     (3.30)          (-4.18)          (1.85)

          SEE=     1.009447   R-Bar-Sq= 0.9192   DW=   1.810
          Ordinary Least Squares
          Sample Period=   1921-1941

Equation Number   6:   W1

W1= 0.43948*(Y+TAX-W2)+0.14609*(Y(-1)+TAX(-1)-W2(-1))+0.13025*T1931+1.49704
     (13.56)            (3.90)                          (4.08)        (1.18)

          SEE=     0.767146   R-Bar-Sq= 0.9852   DW=   1.958
          Ordinary Least Squares
          Sample Period=   1921-1941
```

Figure A4. Klein Model I Equations

The third macro mentioned earlier, TSKLEIN.MAC, provides an alternative to ESTMOD.MAC. This macro permits you to estimate the behavioral equations using Two Stage Least Squares, instead of Ordinary Least Squares. Its contents are displayed in Figure A5 below. Notice that this macro is similar to ESTMOD.MAC in terms of its commands to access KLEINBNK and set the frequency and dates. However, it also contains an INSTRUMENTS command, selecting as instruments the model exogenous variables, and specifies that each equation is to be estimated by Two Stage Least Squares. One aspect of the INSTRUMENTS command needs comment: Observe that the lagged variables PLAG [=PI(-1)], KLAG [=K(-1)], and XLAG [=Y(-1)] are included as pre-computed transformations; the elements of the INSTRUMENTS command must be simple variable names and cannot be expressions.

```
; Two Stage Least Squares Estimation Example
ACC KLEINBNK
SET FREQ=ANN
SET DATES=21-41
INSTRUMENTS:W2,TAX,G,T1931,PLAG,KLAG,XLAG
TSLS(2):CE=F(W1+W2,PI,PI(-1))
PUTEQ
TSLS(1):I=F(PI,PI(-1),K(-1))
PUTEQ
TSLS(1):W1=F(Y,Y(-1),T1931)
PUTEQ
```

Figure A5. TSKLEIN.MAC File

Essentially, the TSKLEIN.MAC macro should be run after KIDENT.MAC instead of ESTMOD.MAC. As an option, you could create an alternative version of KIDENT.MAC in which you choose another name for the model in the context of the DEFINE MODEL command; for instance, KLEIN2S:

DEFINE MODEL KLEIN2S

This change would allow you to create an alternative model file, KLEIN2S.MOD, and an alternative version of the model. You would need to separately compile this model and the result would be a compiled model file KLEIN2S.CMF. Furthermore, it would require its own solution file(s), inasmuch as a solution file is particular to each compiled version of a MODLER model. However, otherwise, this estimated version of

KLEIN Model I could be used, solved, and displayed in the same way as the OLS version, as described in Chapter 2 and the following chapters of this workbook. As discussed at the beginning of Chapter 2, once a model's parameters have been estimated or otherwise put into the behavioral equations, the process of using the model is essentially the same, whatever the method employed. Of course, the statistical properties of the model's predicted or simulated endogenous variable values may be affected by the parameter estimation.

References

1. Adams, FG, *Email re software used in the Economics Research Unit, Department of Economics, University of Pennsylvania*, CG Renfro, Recipient. 2003: Boston.
2. Adams, MO, *The Kentucky Economic Information System: a resource for government, academic, and private sector researchers and libraries*, in *National Online Meeting Proceedings - 1981*, ME Williams and TH Hogan, Editors. 1981, Learned Information: Medford, NJ.
3. Adams, MO, *The Kentucky Economic Information System*, in *Exemplary Systems in Government Awards '85-'86: the State of the Art*. 1986, Urban and Regional Information Systems Association. p. 144-158.
4. Berndt, ER, *The Practice of Econometrics: Classic and Contemporary*. 1991, Reading, MA: Addison-Wesley Publishing Company.
5. Bodkin, RG, LR Klein, and K Marwah, *A History of Macroeconometric Model-Building*. 1991, Brookfield, Vermont: Edward Elgar. 573.
6. Christ, CF, *Econometric models and methods*. 1966, New York,: Wiley. xxiii, 705.
7. Desai, M, *Applied econometrics*. 1976, Oxford: P. Allan. x, 272.
8. Evans, MK, *Macroeconomic Activity:Theory, Forecasting, and Control; an Econometric Approach*. 1969, New York: Harper & Row, Publishers, Inc.
9. Friedman, M and AJ Schwartz, *Alternative approaches to analyzing economic data.* American Economic Review, 1991. **81**: p. 39-49.
10. Godley, W and M Lavoie, *Prolegomena to Realistic Monetary Macroeconomics: A Theory of Intelligible Sequences*, in *Levy Institute of Bard College, Working Paper No. 441*. 2006, Levy Economics Institute: Annandale-on-Hudson. p. 39.
11. Godley, W and M Lavoie, *Monetary Economics. An Integrated Approach to Credit, Money, Income, Production and Wealth*. 2007, Basingstoke: Palgrave Macmillan.
12. Goldberger, AS, *Impact Multipliers and Dynamic Properties of the Klein-Goldberger Model*. 1959, Amsterdam: North Holland Publishing Company.
13. Goldberger, AS, *Econometric Theory*. 1964, New York: John Wiley & Sons, Inc.

14. Griliches, Z, *Data and econometricians--the Uneasy Alliance.* American Economic Review, 1985. **75**(2): p. 196-200.
15. Hendry, DF, *Econometrics : alchemy or science? : essays in econometric methodology*. New ed. 2000, Oxford, UK ; New York: Oxford University Press. xviii, 542.
16. Howrey, EP and HH Kelejian, *Simulation versus analytical solutions*, in *The Design of Computer Simulation Experiments*, TH Naylor, Editor. 1969, Duke University Press: Durham, N.C. p. 207-231.
17. Intriligator, MD, RG Bodkin, and C Hsiao, *Econometric models, techniques, and applications*. 2nd ed. 1996, Upper Saddle River, NJ: Prentice Hall.
18. Kendrick, D and A Meeraus, *Forward (to the special issue on modeling languages).* Journal of Economic Dynamics and Control, 1983. **5**: p. 1-4.
19. Klein, LR, *Economic Fluctuations in the United States 1921-1941*. Cowles Commission Monograph. Vol. 11. 1950, New York: John Wiley & Sons, Inc.
20. Klein, LR and RF Kosobud, *Some econometrics of growth: Great ratios in economics.* Quarterly Journal of Economics, 1961. **75**(May): p. 173-98.
21. Klein, LR, *An Essay on the Theory of Econometric Prediction*. 1971, Chicago: Markham Publishing Company.
22. McCracken, MC and CA Sonnen, *A system for large econometric models: management, estimation, and simulation*, in *Proceedings of the Association for Computing Machinery Annual Conference, Boston, Massachusetts, August 1972*. 1972, Association for Computing Machinery.
23. McCullough, BD and CG Renfro, *Some numerical aspects of nonlinear estimation.* Journal of Economic and Social Measurement, 2000. **26**: p. 63-77.
24. Quandt, RE, *Computational problems and methods*, in *Handbook of Econometrics*, Z Grilliches and MD Intriligator, Editors. 1990, North Holland: Amsterdam. p. 699-764.
25. Renfro, CG, *Economic Data Base Systems: Some reflections on the state of the art.* Review of Public Data Use, 1980. **8**(3): p. 121-140.
26. Renfro, CG, *On the Development of Econometric Modeling Languages: MODLER and its first twenty-five years.* Journal of Economic and Social Measurement, 1996. **22**(4): p. 241-311.
27. Renfro, CG, *Macroeconometric models and changes in measurement concepts: an overview.* Journal of Economic and Social Measurement, 1998. **24**(2): p. 63-82.
28. Renfro, CG, *Econometric Software: the first fifty years in perspective.* Journal of Economic and Social Measurement, 2004. **29**(1-3): p. 9-108.
29. Renfro, CG, *The early development of econometric modeling languages.* Journal of Economic and Social Measurement, 2004. **29**: p. 145-166.
30. Renfro, CG, *The Practice of Econometric Theory*. 2009, New York: Springer.
31. Theil, H, *Principles of econometrics*. 1971, New York: Wiley.