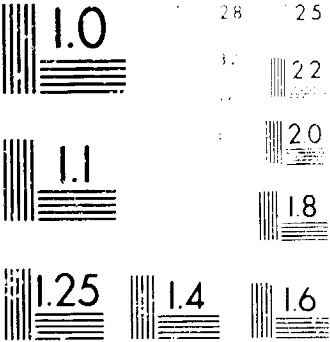
UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org



MICROCOPY RESOLUTION TEST CHART

Distr.
LIMITED

UNIDO/IS.378
28 March 1983

ENGLISH

UNITED NATIONS
INDUSTRIAL DEVELOPMENT ORGANIZATION

THE IDIOM USER'S HAND BOOK *

Prepared by the
Global and Conceptual Studies Branch

World Modelling Working Paper

L. Alan Winters

* This document has been reproduced without formal editing.

V.83-53880

THE IDIOM USER'S HANDBOOK

Page

FOREWORD

This manual is being issued in order to provide information on the IDIOM software package, developed by Cambridge Econometrics Ltd. and installed at UNIDO.

The manual was written to make easier the task of using the package, especially for new users. The package is a substantial one and requires, in its present form, a medium-sized computer to run it. It is hoped in the future to make the package generally available (in fact it has already been supplied by UNIDO to one developing country). The issuing of this manual is a step in this process.

However, the manual itself can also be seen as a summary of possibilities in the area of dynamic input-output modelling of a national economy. The package provides a means by which different aspects of the task, such as investment, trade, employment and consumption can be integrated and rendered consistent both from an accounting and structural point of view.

Being developed out of the work of the Cambridge, United Kingdom, the particular interests and emphases of that work are to some extent carried over. Not all functional forms offered are appropriate to a developing country, nor is the necessary data for their estimation widely available. More fundamentally, the model is demand-oriented, and supply constraints are not explicitly embodied. A flexible software package, which allows for repeated simulations of alternative demand patterns, can, however, go some way towards the examination of supply constraints, and IDIOM may thus be of some value in the study of very different economies. More generally, it is felt that the range of functional choices and the detailed accounting framework can provide useful indications to those modelling national economies.

The IDIOM Handbook was prepared for UNIDO by Dr. Alan Winters of the University of Bristol, United Kingdom.

AUTHOR'S PREFACE

In 1980 UNIDO took delivery, from Cambridge Econometrics Limited, of a computer package called IDIOM. It was designed to facilitate the construction and solution of dynamic input-output models along the lines of the Cambridge Growth Project's model MDM (see Barker, Borooah, van der Ploeg and Winters, 1981). The computer programmes were extensively commented, and the package was provided with a User's Manual, but unfortunately these guides proved insufficient for anyone without a Cambridge background to penetrate the complexities of the programme. This handbook was therefore commissioned from the present author, one of IDIOM's authors, who had by that time left Cambridge. It is intended to make the use of IDIOM much more obvious to non-specialists, than did the original documentation.

There have been several difficulties in the implementation of IDIOM in Vienna and the author has played a considerable part in uncovering and, hopefully, recovering from them. The upshot of this activity, however, has been to delay the present document by a few months and to render it more provisional than might be hoped. All computer programmes are subject to errors, especially those of the size and complexity of IDIOM. These gradually emerge during acceptance trials and during the first few years of use. IDIOM has been no exception. Unfortunately, however, there existed a chicken and egg problem: until it was used as intended, IDIOM would never be completely tested, and until it was properly documented, it could not be properly used; but, until it was properly tested, it could not be definitively documented. This paper, then, is intended to break the dead-lock and must be treated as being to some extent provisional. The heart of the paper - the function descriptions - will not be changed, except perhaps where the need for changes is indicated below, but the sections on the behaviour of the programme and on hints to users will no doubt have to be altered in the light of experience. Indeed, during the preparation of the handbook, several unfortunate features of IDIOM have come to light, and rather than postpone the handbook further, we have merely noted them, evaded them and passed on. Hence it is certain that IDIOM will evolve somewhat from the present version. This will date the handbook, so that in the near future a slightly revised version of it will be necessary.

In preparing this handbook I have received invaluable help from William Peterson, of Cambridge University, for which I am very grateful.

L. Alan Winters
University of Bristol

I. INTRODUCTION

THE PHILOSOPHY AND HISTORY OF IDIOM

IDIOM is a computer package developed by the Cambridge Growth Project to aid the specification and solution of large-scale dynamic input-output models. It is designed to handle a wide variety of models, from a static Leontief system to a full dynamic econometric model of the sort currently used in Cambridge, and in all cases offers the user considerable flexibility over the specification of economic relationships. It undertakes various accounting activities automatically and also provides facilities for storing and analyzing the model results. Within the limits of simple and efficient use, IDIOM obviously cannot provide complete freedom of specification, but the package has also been designed so that the user may edit it to his own requirements with a minimum of fuss. Hence the user with some experience of both IDIOM and FORTRAN programming may have almost unlimited freedom, over economic relationships, while the package looks after accounting and house-keeping operations.

IDIOM permits the user to specify his own disaggregation for up to sixteen different economic concepts - commodities, industries, exports, investment assets, types of labour etc. - and also to vary the list of endogenous variables. This is particularly relevant for modelling developing countries, for then data inadequacies often make it impossible to model all sectors of the economy in equal detail.

The history of IDIOM is in a sense the history of the Cambridge Growth Project. For over twenty years the latter group have been developing disaggregated medium-term models of the British economy - stretching from an essentially static Leontief model (Stone 1964) to a dynamic simulation model, MDM - most recently described in Barker, Borooah, van der Ploeg and Winters (1980). The latter is built around the input-output table but additionally incorporates many features of the now familiar macro-economic simulation model. For instance it incorporates a full set of financial flows, endogenous prices, dynamic functions explaining all elements of final demand, and a fully specified, cost-minimizing, production sector. IDIOM makes this level of generality simply and conveniently available to model-builders elsewhere in the world. It is based on the CGP model (indeed the current CGP model uses IDIOM), drawing on the experience of MDM for its economic structure, its internal construccion and its user-interface. Hence while IDIOM is a relatively new programme, it incorporates the experience of one of the longest-lived and most successful of the world's model-builders.

IDIOM is potentially much more general than MDM (the CGP model which forms its base), for any user may adapt it to his own requirements. Hence in no way need the experienced user adopt a 'CGP' view of the world in order to utilize IDIOM. On the other hand, the new user may wish initially to use IDIOM with little or no reprogramming, and so will make use of the current structure. This inevitably reflects its background and is rather similar to the latest version of MDM. To get a feel for IDIOM, therefore, as well as reading the IDIOM literature, the user should read Barker et al (1980) and Barker (ed.,1976). The former gives an overview of MDM, while the latter, referring to an earlier version, gives more detail of the programme and its use, expecially on individual sectors. While the latter is somewhat dated, it provides the soundest basis from which to explore the more recent developments within IDIOM.

Barker (ed)(1976) not only describes the economic and simulation features of MDM, but also contains some information on the estimation of the various relationships. Data and estimation are essential steps in any model-building exercise and it is important that IDIOM users have either access to econometric advice or econometric experience themselves. Parts of this handbook offer advice on estimation and data preparation, but they are no substitute for an adequate background. For those potential users who have not much experience of econometrics, we would recommend Intriligator (1978) as a book which maintains a sensible balance between econometric theory and practice. Certainly anyone who knows the material therein will have no problem in using and exploiting both IDIOM's sister programme - the estimation package MREG - and IDIOM itself.

THE IDIOM LITERATURE

The first element of the IDIOM literature is the general introductory paper Barker, Peterson and Winters (1979), a copy of which is supplied with this handbook. This describes the history, motivation and philosophy behind IDIOM, but gives no practical details at all. Secondly, there are two manuals supplied by Cambridge Econometrics and written largely by William Peterson. These cover respectively IDIOM and MREG and provide technical descritpions of the use of these programmes. They do not, however, give any details of the programme construction, nor of the economic or econometric significance of the packages. It is the purpose of this handbook - the third element of the IDIOM literature - to fill in these gaps. The econometric package, MREG, is straight-forward, and as suggested already its features will be transparent to anyone with a little experience of econometrics. Consequently, except in passing, this handbook will not describe the use of MREG, although we shall make a few comments on the preparation of data for estimating the relationships IDIOM requires.

Turning to IDIOM there is more need for the handbook. There are three main elements to this publication. First, a description of the economic framework - stretching from the accounting system to the details of each set of available functions and advice on estimating them. Second, a description of the programming structure. This is necessary partly to give the less experienced user insight into how the programme will behave under various circumstances and partly to provide the more experienced user some guidance when he comes to edit the programme to meet his own specific requirements. This is the shortest part of the handbook, partly because the material is touched upon in the manual, but also because our purpose is not to describe the programme fully and show how to make any conceivable change, but rather to introduce the experienced FORTRAN programmer to the basic structure and set him a few guidelines on how to find his way around the programme. It is to be emphasized that, while editing IDIOM is perfectly legitimate and straight-forward, it is not for the inexperienced.

THE IDIOM LITERATURE

The first element of the IDIOM literature is the general introductory paper Barker, Peterson and Winters (1979), a copy of which is supplied with this handbook. This describes the history, motivation and philosophy behind IDIOM, but gives no practical details at all. Secondly, there are two manuals supplied by Cambridge Econometrics and written largely by William Peterson. These cover respectively IDIOM and MREG and provide technical descritpions of the use of these programmes. They do not, however, give any details of the programme construction, nor of the economic or econometric significance of the packages. It is the purpose of this handbook - the third element of the IDIOM literature - to fill in these gaps. The econometric package, MREG, is straight-forward, and as suggested already its features will be transparent to anyone with a little experience of econometrics. Consequently, except in passing, this handbook will not describe the use of MREG, although we shall make a few comments on the preparation of data for estimating the relationships IDIOM requires.

Turning to IDIOM there is more need for the handbook. There are three main elements to this publication. First, a description of the economic framework - stretching from the accounting system to the details of each set of available functions and advice on estimating them. Second, a description of the programming structure. This is necessary partly to give the less experienced user insight into how the programme will behave under various circumstances and partly to provide the more experienced user some guidance when he comes to edit the programme to meet his own specific requirements. This is the shortest part of the handbook, partly because the material is touched upon in the manual, but also because our purpose is not to describe the programme fully and show how to make any conceivable change, but rather to introduce the experienced FORTRAN programmer to the basic structure and set him a few guidelines on how to find his way around the programme. It is to be emphasized that, while editing IDIOM is perfectly legitimate and straight-forward, it is not for the inexperienced.

II. BASIC IDIOM

AN OVERVIEW OF BASIC IDIOM

This section describes the broad features of the types of model that can be built with IDIOM without reprogramming. We do not describe here the details of particular functions, but rather the broad concept and structure of the models as a whole. Even without reprogramming, IDIOM provides much flexibility and so this section should be regarded as indicative rather than definitive. Also the user should realize that the full list of features listed here need not be implemented in any particular model, for various functions may be suppressed, and this can drastically alter the overall look of the model. Hence if the model you have in mind does not look like what follows, do not despair, it is still quite possible that IDIOM will handle it.

(a) Overall Features:

(i) basically "Keynesian"

(A) no supply constraints: supply expands to meet demand in all markets, although this might affect prices and hence feed back onto demands and supplies.

(B) fix-price model structure: prices are basically cost determined rather than adjusting to clear the market to which they refer. However, one option allows the profit mark-up to vary cyclically and wages may also respond to excess demand in the labour market. Hence something akin to neo-classical, or market clearing, prices is possible.

(C) Financial variables do not feature prominently in the equations for real variables. They may feature by altering income flows or, via interest rates, affecting costs, but the effect on real variables is always indirect.

(D) basically flow equilibrium: little attention is paid to stock equilibrium in asset markets although it features in investment behaviour.

(ii) dynamic - one year is, to a considerable extent determined by the last. Many relationships incorporate lags, and all could do so if required. Dynamic also in the sense that the sources of economic growth have to be found within the model; hence growth of the capital stock is only possible through investment, the resources for which must be found from within the model.

(iii) disequilibrium - most markets clear, in that supply = demand, but neither necessarily attains its long-run values (given the values of exogenous data) immediately. Endogenous cycles are possible - even likely. Two crucial markets do not clear even in the sense used earlier. The labour market and the foreign exchange markets can both show persistent excess supply or demand if government policies are assumed unresponsive to their condition. The labour market may be controlled to some extent by the endogeneity of wages, but in plenty of cases stable under-employment solutions are found. In the foreign exchange market the exhaustion or over-accumulation of reserves is assumed to act via changes in government policy (user - supplied) rather than endogenously and directly on the exchange rate.

(iv) monetary - subsistence sectors can be fully modelled only to the extent that they can be valued in money terms. There is scope however for the personal sector to employ workers directly (at a given wage) and the absorption of surplus labour in rural sectors may be modelled through this.

(v) national - it handles one nation (or region) at a time, taking the rest of the world (nation) as exogenous.

(b) Principal Exogenous Variables

(i) the rest of the world - incomes and prices
- foreign exchange rate (this could be fairly easily endogenized if desired)

(ii) demographic variables - population
- participation rates
- dependency ratios

(iii) economic policy - especially tax system
- government expenditure

(Some elements here are defined such that while the system is exogenous, the actual expenditure is not - e.g. unemployment benefit depends on unemployment, which is endogenous.)

(iv) economic behaviour, including the formation of expectations, i.e. obviously IDIOM needs to be supplied with parameters for each of its behavioural functions.

(c) The Main Relationships

(i) The final demands:

Final demands may be modelled explicitly using functions whose arguments are drawn from all parts of the model. They are the prime movers of the model in that supply expands to meet whatever is demanded, although there is room for feedbacks from supply to demands, e.g. imports rising and exports falling if demand exceeds "normal" supply, or via prices and inflation (the Phillips curve). The following elements of demand may be explicitly modelled:

q_x exports
q_c consumption
q_i investment
q_s stock-building
q_g government expenditure
q_m imports - treated either as negative final demand, or complementarities.

(ii) Production:

Supply is met by combining labour, capital and materials. The material demands are handled via the input-output segment (although input-output coefficients are not necessarily fixed); the remaining two are derived from production relationships. A number of different relationships allow one to derive investment by asset and employment by type given gross output, material inputs, various prices and certain exogenous data. Given employment for each industry, aggregate employment and unemployment can be derived. Finally the stock-building relationships determine stock-building within each sector from gross output and, possibly, other data.

The final demands and production are linked by a commodity balance which states that supply and demand for each commodity must be equal. Hence, writing q for total domestic output:

$$q + q_m = q_c + q_i + q_g + q_s + q_x + q_n$$

where q_n is intermediate demand by commodity

$$q_n = A_{gy}y$$

where y is output by industry and A_{gy} the input-output (mix) matrix giving commodity inputs per unit of industry output.

The commodity balances are the lynch-pins of the whole of IDIOM's accounting system and modelling approach.

Related but less important are the industry balances which state that inputs and outputs must balance by industry. Hence:

$$y = (i'A_{gy}\hat{y})' + w + \pi + t + m_d$$

- w wage and salary bill (including national insurance, etc.)
- π profits, etc.
- t indirect taxes
- m_d direct imports (i.e. firms' imports not entering commodity accounts e.g. business travel; may be zero)

$(i'A_{gy}\hat{y})'$ is total intermediate input ($\hat{y}$ denotes a matrix comprising zeroes except for the elements of y down the diagonal; hence: $A_{gy}\hat{y}$ gives the matrix of intermediate flows and i' sums it for each industry - column sums)

(iii) The price relationships:

Industrial prices are basically cost determined, being driven by labour costs, material costs, occasionally capital costs and indirect taxes. Material costs may include a substantial element of import costs. As alternatives, industrial prices may be constrained to equal import prices, or be derived from cost, but with a variable mark-up (leading possibly to something akin to neo-classical pricing behaviour).

Given industrial prices, the prices of other elements of final demand may be found. Import prices may be related to local prices (hence p and p_m are mutually dependent) or merely fixed in foreign currency. Export prices may similarly either reflect the domestic economy (via p_q) or be constrained to world levels. The price of domestic absorption - presumed common to the remaining elements of final demand - is fixed as a weighted average of domestic output and import prices. These prices possibly have to be converted to the correct classification before use in their relevant functions.

(iv) The cost of labour:

The average wage may be a function of taxes, inflation etc. This enters the employment functions, where employment is determined, and from this, noting any non-constant returns (to labour or to overall scale), unit labour costs emerge. These may then determine prices if required.

(v) The income flows:

For each type of income (e.g. wages and salaries, profits, taxes) accounts are kept and for each institutional sector (households, companies, etc.) both incomes and expenditures are calculated. These may feed back into the demand equations most obviously via the consumption function, but also possibly through investment being related to profitability etc. The accounts also form the basis of a complete set of financial accounts showing flows of funds etc. All these accounts allow for government taxes and transfers.

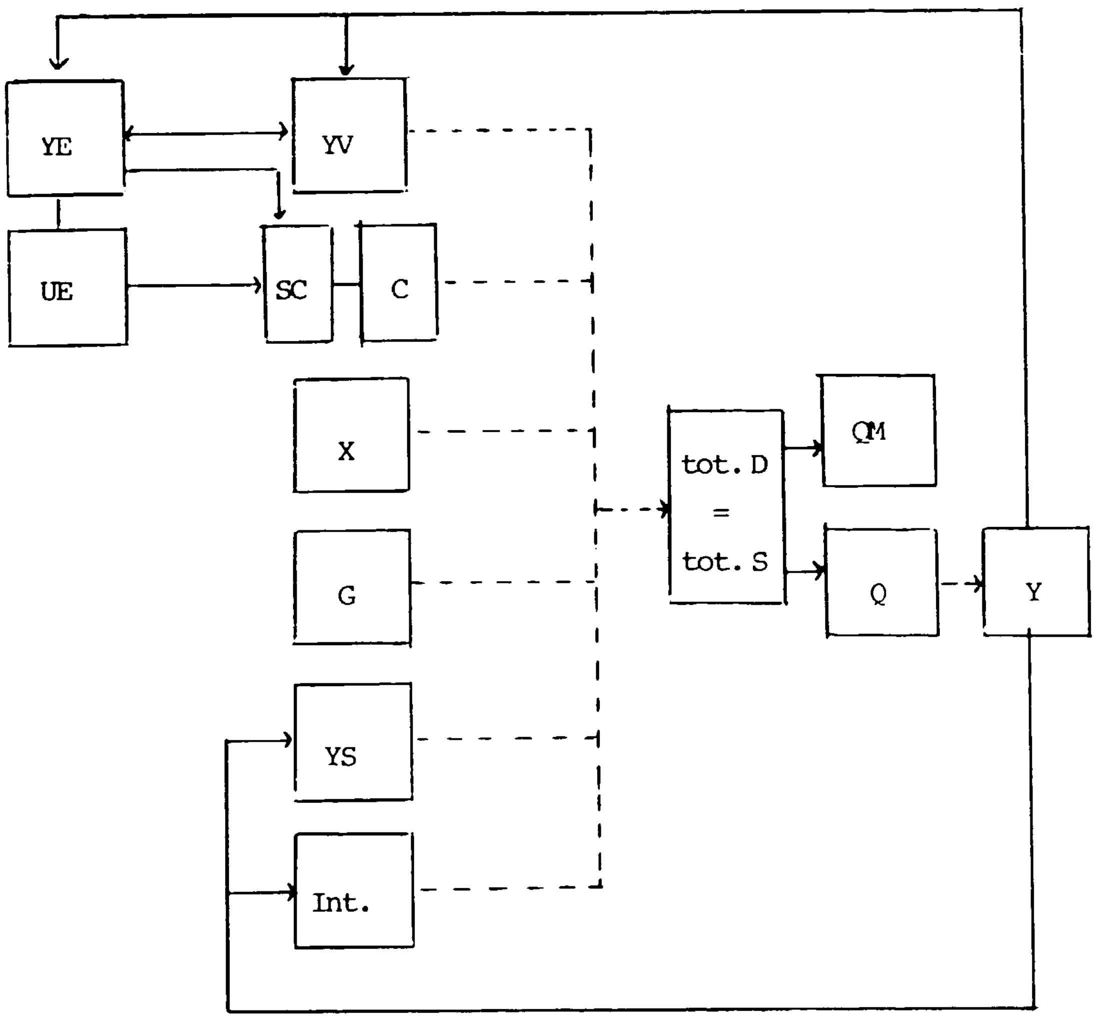
(d) Flow Diagrams:

The main causal chains and interactions in basic IDIOM are represented below. Two types of link are represented: the behavioural links (solid lines) where the user defines the nature of the link by means of IDIOM functions, and the accounting links (broken lines) which are carried out by the model independently. The latter comprise either aggregations or conversions from one classification to another.[1/]

The diagrams show the direct links between the variables they contain They do not, however, show either the indirect links or a full set of determinants of each variable. Hence, for example, in figure 1 the demand categories depend on prices, but these are not explicitly included. The variables are referred to by their standard names (see Naming Conventions below) and potentially each has its own classification.

1/ The classification converters themselves are often endogenous, but the act of conversion is automatic.

Figure 1 The Main Links of the Real Side



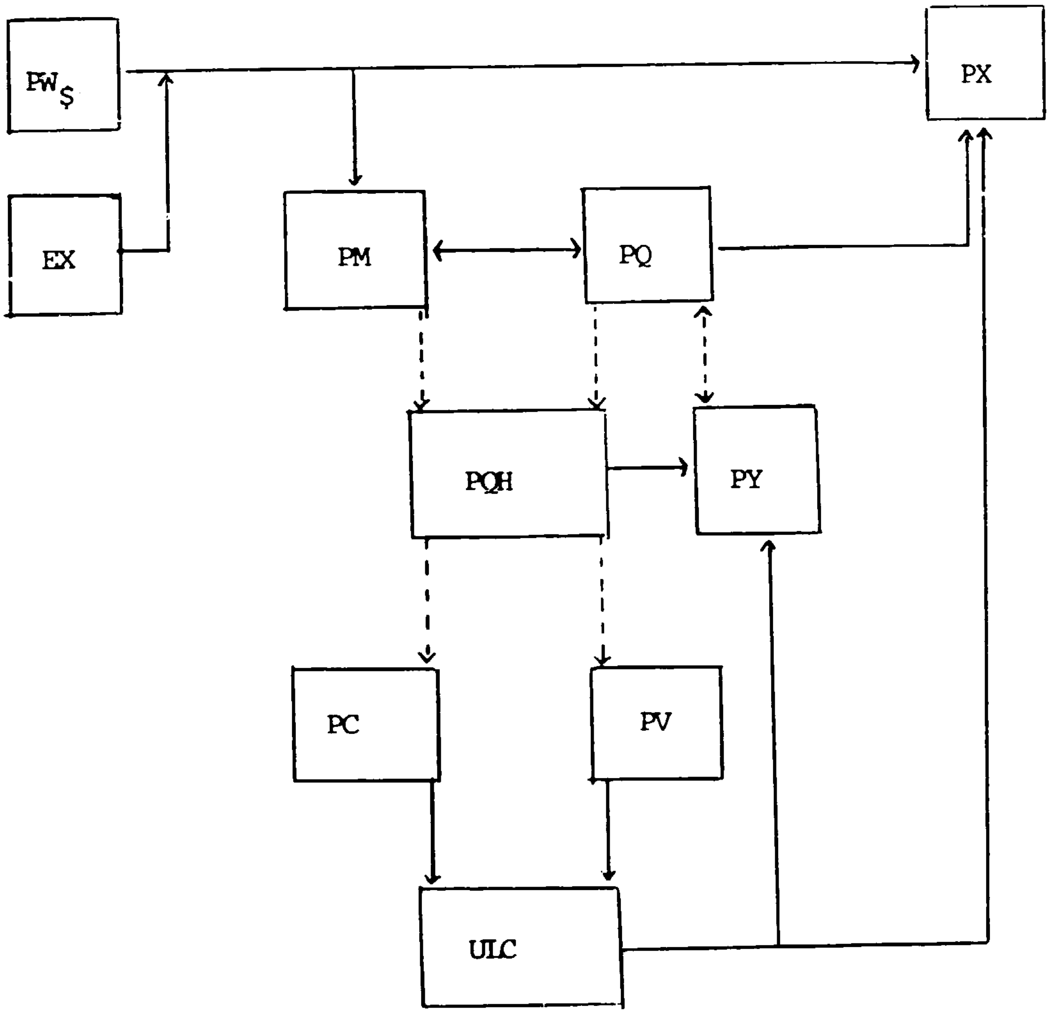
The symbols are standard names as defined in "The Naming Conventions"

Int. is intermediate demands

behavioural links (functions)

accounting links

Figure 2 The Main Links in the Price Sector



The symbols are standard names as defined in "The Naming Conventions", except for $PW_\$$ which is general world prices in foreign currency

behavioural links (functions)

accounting links

THE ECONOMIC STRUCTURE

We turn now from the broad over-view to the economic detail of IDIOM (programming details are left until later).

The most crucial concept in IDIOM is the calssification which defines the disaggregation of each major variable. We start by discussing the classifications (and hence implicitly defining the major endogenous variables). After that we consider the accounting operations of IDIOM (standard national accounting conventions), and the conversion between different classifications. The behavioural relationships or functions are examined in the next section but one.

CLASSIFICATIONS

One of the most obvious yet significant features of IDIOM is that it is disaggregated. Virtually every important economic variable may be disaggregated in IDIOM models, and potentially at least, each may have a different disaggregation. This allows the user to exploit local data and specialist research to the full, for he can adopt whatever classification best suits a particular sector without regard to the rest of the model. The only requirement is that he be able to supply a converter, or bridge matrix to convert data from one classification to another.[1/] The list of classifications used by IDIOM is given in Table 1 below, along with certain naming information and an indication of the order of disaggregation used in MDM.

The user may have as many or as few elements as he wishes in each classification with the following restrictions:

(i) each classification has at least one element

1/ Obvioulsy all classifications do not have to be mutually linked. The list of required conversions is given in Table 3 of Naming Conventions, below.

TABLE 1: IDIOM FUNCTIONAL CLASSIFICATIONS

Classification		Description
Q	1	Commodities
Y	2	Industries
C	3	Consumers' expenditure categories
G	4	Government expenditure categories
R	5	Types of receipt and payment
H	6	Income-receiving sectors
S*	7	Stocks/inventories
V	8	Durable capital assets
F*	9	Financial assets
X	10	Exports
M	11	Imports
E	12	Employment
D	13	Direct tax brackets (income distribution)
T	14	Indirect taxes
A	15	World areas
K	16	Social capital formation
O	17	Overall calssification (dummy: used to hold data relating to classifications 1-11)

* redundant in IDIOM version 3

(ii) The number of industries is not more than the number of commodities. Each industry must have a principal product (a commodity which accounts for a large part of its output and may roughly be taken as representative of its output).

(iii) R must include at least the following five categories:

wages and salaries
profits
indirect taxes
direct imports
goods and services

(iv) H must include at least the following four sectors:

households
corporations
central government
rest of the world

(v) O must contain 11 elements

Table 1 also implicitly defines the main economic variables considered by IDIOM. The split between exogenous and endogenous is fairly obvious and further details of the latter now follow.

ACCOUNTING

We distinguish three elements to model building - housekeeping, accounting and behaviour. The first includes such things as keeping track of information, input and output, error trapping, and diagnostics etc. It is essentially a question of computer programming and, to the extent that we discuss it at all, it is covered in Chapter III. The last two elements are the economic features. They tend to merge with each other at the edges, but broadly behaviour covers those areas of the model where economic agents may exercise their discretion, while accounting covers those areas - mostly comprising definitions - where there is no scope for variation.

Among the more obvious examples of accounting we have the commodity and industry balances, the definition of personal disposable income, the calculation of indirect tax receipts once quantities and prices are known, and the calculation of stock levels from stock-building data. Possibly less obviously but still basically accounting are conversions from one classification to another. This is carried out using bridge, or converter, matrices which express the elements of one classification as proportions of the elements of another. For instance given a vector of exports disaggregated by the export classification, we need to convert this to the (domestic) commodity classification in order to use it in the commodity balances; this is accomplished by using a matrix (QXC), a typical row of which shows the proportion of each export group entering the commodity of that row. Similarly, given the prices of commodities on the domestic market (PQH), we can generate the prices of each element of the consumers' expenditure classification by means of a commodity - consumer converter (QCC).

Stretching the definition of accounting still further we also include the standard input-output calculation by which intermediate demands by commodity are calculated from industry gross outputs (Y), and the mix matrix (QYC) which shows inputs per unit output. From this example it will be seen that the column sums of the converters need not sum to unity (for total intermediate input is less than gross output). It will also be seen below that although the converter is fixed exogenously to any particular accounting operation, it may still be endogenized elsewhere in the model, for there are several possible ways of updating the mix matrix within IDIOM. Accounting operations similar in nature to the input-output operation include the calculation of value-added in the various non-industrial sectors (given the proportions of any class of expenditure that is value-added).

The importance of separating accounting from behavioural operations in IDIOM is that the former are programmed directly and unavoidably for the user, whereas the latter offer the user considerable flexibility. Obviously the elements of the converter matrices are determined by the user, but thereafter he has no control over how and when the operation is carried out. The reason for this is that we believe that no user would actually wish to alter the standard procedures and so offering him a choice would needlessly complicate the use of IDIOM. The full set of accounting operations is not specified anywhere - the majority being far too obvious to enumerate - but a list of required converters - and hence of feasible accounting conversions - is given in the section Naming Conventions, which follows.

IDIOM NAMING CONVENTIONS

The following conventions are used for referring to variables both within the FORTRAN source of IDIOM and for communication with the programme during the various phases of using IDIOM. They will also be used fairly freely within this hand-book. In addition to this, the user should also consult the dataset JRNS.IDIOM.TABLS which defines all the vectors and matrices used in IDIOM, along with their dimensions and units.

Classifications

The heart of the naming convention, as well as of the whole modelling package, is the classification system. The seventeen classifications, along with their identifiers, are given in Table 2. The identifiers come in both alphabetic and numeric form. The latter are of little importance to the user, but do figure in the internal construction of the programme; the former are the principal elements of the naming conventions.

Most variables have at least two letters in their names, even without the qualifiers that we shall describe shortly. The first defines the classification over which the data are defined and the second the economic concept in question. For example:

QC consumption by commodity
YS stock-building by industry

If, however, the two letters would be identical, they are collapsed into one, e.g.:

C consumption by consumption group
V investment by asset

TABLE 2: THE MAIN IDIOM CLASSIFICATION NAMES

Classifications: each is denoted by a single letter.

		MDM	Identifier
Q	commodities	57	1
Y	industries	40	2
C	consumers' expenditures	42	3
G	government current expenditures	5	4
R	receipts and payments	?	5
H	institutional sectors	?	6
S*	stockbuilding	2	7
V	fixed investment	9	8
F*	financial investment	?	9
X	exports	16	10
M	imports	5	11
E	employment	1	12
D	tax brackets	25	13
T	indirect taxes	8	14
A	world areas	10	15
K	government capital expenditures	5	16
O	overall SAM classification (first 10 classifications above)	10	17

Other

U	units of measurement (real x 2A4)
W,Z	real workspace
I,J	integer workspace

---

* Redundant in IDIOM Version 3

Qualifiers

The above names require qualifiers in fairly obvious circumstances. The latter include:

pre-fix:	B	parameter matrix referring to entity concerned
	S	sum of entity concerned
	N	number of items in entity concerned
	P	price of entity concerned
	SP	current price sum of entity concerned
	L	location in workspace of entity concerned (This is of significance only to programmers, rather than users)
	JSW	(or JS) switch for entity concerned
post-fix:	Ln	lag of n years on entity concerned (L is sometimes suppresed)
	TI	titles for entity concerned
	Ø	row sum of entity concerned (If attached to an entity which is normally a vector ignore it)
	R	receipts by entity concerned } only occur with H and R classification
	P	payments by entity concerned } only occur with H and R classification
	TZ	base year taxes on entity concerned as proportion of base year output
	B }	classification converter
	C }	(more details follow)
examples	BPY	parameters for industry prices, dimension (NY, NBPY)
	NY	number of industries
	NBPY	number of parameters for industry prices
	LBPY	position of start of BPY in ZZ

SG	total government current expenditure (constant prices)
PSG	price of total government expenditure
SPG	current price sum of government expenditure
PXL1	price of exports lagged once
PXL2	price of exports lagged twice
QTI	commodity titles
QM	imports by commodity
JSWPX	switch controlling export price formation
QMTZ	tariffs on imports in base year

The wide range of classifications in IDIOM necessitates a large number of conversions from one classification to another. These are accomplished by linear converters, or bridge matrices. Hence, given consumption by consumption categories, we may calculate the commodity demands they imply by means of a consumption - commodity converter, e.g.:

$$\underset{(NQ,1)}{QC} = \underset{(NQ,NC)}{QCC} * \underset{(NC,1)}{C}$$

where QCC is the converter (postfix C) from consumption (C) to commodity (Q). The same converter can also be used to define prices (often referred to as "dual prices") as follows:

$$\underset{(NC,1)}{PC} = \underset{(NC,NQ)}{QCC'} * \underset{(NQ,1)}{PQ}$$

where the ' denotes transposition.

The full range of converters is defined in Table 3. Some have the postfix B, rather than C. This is of no real significance, but arises because while the matrices are used in precisely the manner of converters (i.e. straight-forward linear multiplications of matrices), they are conceptually more akin to parameter matrices.

All converter operations are carried out in IDIOM by the routine CCVERT. Associated with each converter name are the following qualifying prefixes:

I an integer converter equivalenced to the main converter in the FORTRAN source. It is used if the converter is stored sparse.

N number of elements in converter

J indicator of type of converter

IDIOM requires the user to define the various conversion operations he will use and to say in what form the converter is to be stored; see the CONVERT command in the Manual. Naturally, he also has to supply IDIOM the data for each converter.

Additional Names:

In addition to the classifications defined so far, certain other concepts are frequently referred to in IDIOM and have their own mnemonics, many of which are qualified in exactly the same way as the classifications. The main additional concepts are as follows, grouped according to their classification:

(a) by industry

ULC unit labour costs (sometimes referred to as YULC)

YEXP expected output

VA value of tax allowances on investment

SA stock appreciation

YH hours worked

YVP investment in plant

(b) by commodity

POH the price domestic absorptions of commodities

PQHH the price of commodities both produced and sold at home

QMQ import quota variable

QY total intermediate demands for each commodity

TABLE 3: IDIOM CLASSIFICATION CONVERTERS

Classification	Identifier	Identifier of Resulting Matrix	Description
T	QM	TQMB	Tax rates on imports
T	Y	TYB	Tax rates on industries
T	C	TCB	Tax rates on consumers' expenditure
T	G	TGB	Tax rates on government expenditure
T	V	TVB	Tax rates on fixed investment
T	QX	TQXB	Tax rates on exports
R	Y	RYB	Value-added parameters (industries)
R	C	RCB	Value-added parameters (consumption)
R	G	RGB	Value-added parameters (government)
R	V	RVB	Value-added parameters (investment)
Q	Y	QYC1/	Commodity requirements per unit industry output
M	Q	MQC	Classification converter: commodities-imports
Y	Q	YQC2/	Classification converter: commidities-industry
Q	C	QCC	Classification converter: consumption-commodities
Q	G	QGC	Classification converter: government-commodities
Q	S	QSC*	Classification converter: stocks-commodities
Q	V	QVC	Classification converter: investment-commodities
Q	X	QXC	Classification converter: exports-commodities
V	K	VKC	Classification converter: social capital formation-assets
Q	YP	QYPC	Classification converter: plant-commodities
E	YE	EYEC	Classification converter: industrial employment-employment by type
E	CE	ECEC	Classification converter: consumer's employment-employment by type
E	GE	EGEC	Classification converter: government employment-employment by type
H	R	HRC	Classification converter: receipts-sectors
D	R	DRC	Classification converter: receipts-tax brackets
Q	YS	QYSC	Classification converter: industrial stocks-commodities

---

1/ This is the conventional commodity-industry input-output matrix

2/ This is the conventional industry-commodity make matrix

* redundant to IDIOM Version 3

(c) macro-concepts

UE	unemployment (sometimes UNEM; lagged UNL1)
EMPL / EM	Employment, often qualified by sector - see below
AW	average wage, often qualified
EX	exchange rate
WAGE / WAG	wage bill often qualified
GDP	gross domestic product
PDI	personal disposable income (prefix R real PDI; E expected PDI; real PDI lagged 1 and 2 RDI1, RDI2)
HUC	home unit costs (GDP deflator)
PCE	consumers' expenditure deflator (instead of name PSC)

macro-qualifiers:

In addition to the qualifiers already mentioned[1/] macro variables are frequently qualified by the sector of demand to which they refer. We use the postfixes and prefixes:

C	consumers' expenditure
G	government
Y	industry

e.g.	EMPC	employment arising directly from consumer's expenditure (rather than via an industry); includes for example domestic servants, charity workers
	SCVA	value-added arising directly from consumers' expenditure
	SPCM	value of imports arising from consumers' espenditure

additional macro-names:

DFE	domestic final expenditure
WVOL	volume of world output
WPRI	world prices
POP	population

1/ For macro-variebles the prefix B is replaced by a post-fix B to denote a parameter of some sort.

WPOP	working population
RVAT	rate of value-added tax
SRT	standard rate of income tax
RET	retentions ratio = (net earnings/gross earnings)
RINT	rate of interest (long-term government bonds)
RMOR	rate of interest or mortagages (also RMORG)
PSBR	public sector borrowing requirement
PSFD	public sector financial deficit
PWPH	personal wealth per head
SRAT	savings rate = savings/PDI
PUP	GDP factor cost deflator - (currently unused and unset)
TXPD	degree of inflation neutrality in direct tax system
TXPI	degree of inflation neutrality in indirect tax system
PCET	PCE from previous iteration
RESE	residual error
PW	personal wealth

(d) consumers expenditure:

HPD	hire purchase debt variable

(e) foreign classifications:

PC	prices of competing exports (by export good)
PW	domestic prices in export markets (by export area)
DI1	special export effects (by export area)
DI2	special export effects (by export commodity) (DI1 and DI2 are referred to elswhere as "institutional" variables.)
AD1, AD2	two aggregate demand variables (by export area)
PFM	world prices of import goods in foreign currency (in import classification)

(f) employment classification:

LF	labour force
PE	"price of employment" - essentially the average wage by employment class

(g) tax bracket classification:

D	is income tax payed by people in each tax bracket
DN	proportion of working population with income within each bracket
DBT	lower limit of each bracket
DAB	per capita allowances by bracket and income type
DDB	marginal tax rate by bracket and income type
DPOP	population in each bracket
DPDI	disposable income by bracket
DPCE	deflator for consumers' expenditure by bracket
DSC	total consumers' expenditure by bracket
DSPC	total current price consumers' expenditure by bracket

(Note: an individual is defined as belonging to the tax bracket corresponding to the highest marginal rate paid by the individual.)

(d) Using the Name Conventions

The use of the conventions to communicate with IDIOM is illustrated in the example below. The construction of IDIOM means that even simple examples require a full set of concepts and variables to be initialized, so that provided the user understands our test job, he will be able to set up his own jobs satisfactorily by emulating them.

The use of the names in re-programming is necessarily more sophisticated. The user should heed the warnings given above, but once he has decided to edit the programme, the conventions outlined here will be useful. On encountering a new mnemonic he should first apply the conventions for qualified classifications and if that fails, look at the lists of additional concepts. If the mnemonic appears in an "economic" subroutine, it may often be identified by either (a) examining its role in the subroutine solution and comparing this with the programme specifications in this handbook, and/or (b) going back to the calling statement and checking the mnemonic there. Usually a mnemonic in an economic routine is referred to in a calling statement by an address in workspace ZZ with index given by the mnemonic itself preceeded by L; hence QM in the import routine IDMSQM is passed from the calling statement in COMPUT as ZZ (LQM). However, occasionally the address index and the mnemonic do not match perfectly and one is more transparent than the other. Obviously this process could be reversed - identifying an address in the steering routines by looking at a subroutine to which it is passed.

The user is cautioned against introducing further concepts into IDIOM, but if he feels he must, he should stick to the sort of conventions outlined here and, of course, put his adjusted programme through the pre-processor provided.

ECONOMIC BEHAVIOUR - THE FUNCTIONS

We have already drawn a rough and ready distinction between functions and accounts. The latter, we argued, are common to virtually all models, being largely a matter of definition, and are typically linear relationships; the former on the other hand represent economic behaviour and are therefore likely to vary both from country to country and model-builder to model-builder. Furthermore, there is no reason why they should be linear. It is the provision of a wide range of these functions that distinguishes the IDIOM package from certain other input-output modelling work.

For each element of final demand, for various components of the production process, for various income flows and for most prices, IDIOM allows the user to specify, without reprogramming, a range of different functional relationships. The choice varies according to the variable being described, but all the available functions have been found useful, at one stage or another, of the development of the Cambridge Growth Project's models. IDIOM also allows the user to specify his own functional relationships - although this requires FORTRAN programming and is dealt with elsewhere in this hand-book.

This section of the hand-book defines the available functions within IDIOM. For each variable that uses functions the options are defined by a key word (see manual), which tells the programme how many parameters to expect to be given at the input stage and how to use these and the other variables in IDIOM to estimate the variable in question. It is not necessary, in general, that all elements in a vector variable have the same functional relationship, although there are some restrictions which are noted below. It is required, however, that each element keep the same functional relationship throughout a particular IDIOM run. (It is, however, possible to either vary the parameters or to over-write or modify the result of any functional relationship at any stage in IDIOM, so some of the restrictiveness of this requirement is reduced).

For each variable and function option available, we offer below:

(a) a brief description of the economic logic behind it (where this is not obvious);

(b) some notes on the estimation of the relationship;

(c) a definition of each parameter; and

(d) some indication of the structure of the programming (where applicable).

In no sense do the first two components represent a complete description of the applied economics or applied econometrics underlying the functions, and in no sense is the last component a guide to the programme. These parts are intended only as introductory sketches on which further investigation may be based.

Certain features appear quite regularly in the following pages and so they are described here, before looking at specific functions.

(i) Working constant: The solution routines for IDIOM are divided into two phases - the UPDATE phase and the COMPUTE phase. The former is executed once per year of the solution, while the latter is iterated around each year until convergence is achieved. In most functions there are components which can be executed in the UPDATE phase - e.g. inclusion of lagged values, time trends, exogenous data etc. The working constant transfers the results of the UPDATE calculations to the COMPUTE phase where the final estimate of a variable is found as the sum, product etc. of the working constant and the COMPUTE calculations. Virtually every function is split over these two phases, hence requiring two subroutines pre-fixed IDMUxx and IDMSxx, where xx refers to the variable being estimated e.g. IDMUYS and IDMSYS. The working constants need not be initialized to any particular value on input (although they must be initialized to something).

(ii) The regression constant: The constant of the equation defining the function. Usually this is the constant of a corresponding regression equation.

(iii) ρ and Uo: Many functions may be estimated allowing for first order auto-correlation of the residuals. This affects the predictions given by the equation and so IDIOM should be told. Auto-correlation means that

$$E\ (U_t) = \rho U_{t-1}$$

hence if the equation $Y_t = X'_t$ has been estimated allowing for it the optimal prediction of Y_t given X_t is

$$\hat{Y}_t = X'_t\ \beta + \rho\ U_{t-1}$$

and for $\hat{Y}_{t+1}$

$$\hat{Y}_{t+1} = X'_{t+1}\ \beta + \rho^2\ U_{t-1}\ \text{etc.}$$

Within IDIOM, parameter matrices ρ refer to the auto-correlation parameter and U_o to the expected error in the base year of the projection exercise (usually the year preceding the first that could be projected - not necessarily the first year in any particular run, for IDIOM can start up from the middle of the projection period if required). The value U_o may come directly from regression results if the base year is part of the sample period, or it may have to be projected itself prior to entering IDIOM. If auto-correlation is permitted, but not actually used, set $U_o = \rho = 0$.

(iv) JSWxx (where xx is a variable name) These are switches (one for each element of vector xx) which control the function used to estimate xx. They are integers and correspond to the key word definitions, from which, in fact, they are built up by IDIOM during the initialization phase.

(v) Time: Many functions allow for time trends. IDIOM assumes that the trend variable is zero when NYEARS is zero. NYEARS is incremented by one at the end of the update routine; and is initially set equal to YA70, which is set by the user. The regression constant must be calculated on this assumption.

(vi) Parameter matrices' dimensions: Each parameter matrix has its final dimension set automatically by IDIOM. This dimension refers to the number of parameters used by the functions and is set equal to the maximum number required for any function being used for this variable. Hence, for instance, looking at the Import Price Functions below, if all import prices are fixed exogenously, the last dimension of BPQM is set as: NBPM = 1; but if we set just one price with the log-linear function (LLIN) this requires six parameters and so the whole matrix is given final dimension NBPM = 6.

In virtually all functions the final column of the parameter matrix is used as a working constant. This is always indexed as column NBxx (for variable xx), so that even if a particular function does not use the full set of parameters available, the working constant appears at the end of the matrix rather than in the last column that that particular function requires. Hence on the import price example above if some import prices are set by LLIN and some by PFM, the working constants for the latter appear in column six of BPM even though the PFM function could accomodate them in column one. If all import prices used PFM however, then NBPM would be 1 and the working constants would appear in column one.

In some functions other parameters are also indexed relative to NBxx rather than by absolute number. Hence the industrial investment matrix (see below) refers to ρ as BYV (I,J, NBPV-2), and to U_o as BYV (I,J, NBPV-1). Hence, if for example, one investment is explained by the neo-classical model, for which NBYV = 19, ρ, U_o, and the working constant for the other functions will be found in columns 17, 18 and 19 respectively, even though the other functions require only up to seven parameter columns in all.

Notation for the Functional Specification

Where ever possible we refer to variables by their IDIOM names (see Naming Conventions). Most of the variables determined by the functions are vector variables, i.e. their classifications contain more than one element. The functions, however, usually determine their values element by element and below we represent the equations in this, scalar, form.[1/] Hence in the function definitions of this section the use of an IDIOM vector variable implies a typical element of that vector rather than the complete vector. Where we wish to denote the whole vector the name is underlined, and where we wish to show a matrix the name is enclosed in square brackets. Usually the subscripts are suppressed in our equations because the same subscript appears on each "subscriptable" variable, but where this is not the case, subscripts are explicitly used and appear as subscripts, rather than in brackets as they would in FORTRAN. The time subscript does not appear, and lags of k years are denoted by the subscript $_{-k}$.

We refer to parameters in the following way. Assuming there is no possibility of confusion about which parameter matrix is being referred to (e.g. because there is only one possibility), and assuming also that the element subscript has been supressed (as for the variables - see the previous paragraph), we refer to the j^{th} parameter merely as β_j. The only exceptions are ρ and U_o, which have already been explained.

The time trend is always referred to below as t.

*	denotes multiplication (used only between two IDIOM variables)
$\tilde{xx}$	denotes that variable xx has been converted from its own classification to that of the dependent variable before use in the equation. (IDIOM does not have a separate name for the converted variable if it is not stored, for then the result of the conversion is kept in workspace and lost at the end of the subroutine in question.)
NTD	is a timing variable incremented one per year and initialized for the first projection year to YA72.

1/ In one or two cases it is necessary to use a more complex notation, but this is explained below.

$\overline{xx}$ a fixed value for variable xx (this is only used where necessary for expositional reasons; i.e. not every exogenous variable is so marked.)

Z... Names beginning with Z are not proper IDIOM names (because IDIOM does not need to store the variable), but are used here for expositional purposes. They are explained as they occur.

YEAR a timing variable equal to 1 plus the number of years solved since the year of original initialization.

In the tables which define the parameter matrices we use the following notation, again suppressing the element subscript:

β(xx) the coefficient on variable xx (or some transform of xx)

β(r.c.) the regression constant

xx the variable xx of some transform (usually a lagged value)

w.c. the working constant (ocassionally $w.c_i$ if there are more than one)

ρ, U_0 as above

Note: If we are storing data in parameter matrices there is a difficulty about the time period to which it refers, for the data may be updated either before or after its use in the function. The tables below report the presence of the variable required by the function to which they refer, but we also add a note to say how the particular column of the parameter matrix should be initialized.

To illustrate the notation, consider the import price equation on page 38 below. For each commodity to which it applies this equation relates the log of its import price to:

(a) a constant stored in the first column of BPQM

(b) the log of the world price of the commodity in local currency, which is built up from world prices on the import classification converted to the commodity classification and the exchange rate:

$$\beta_2 \log (P\tilde{F}M/EX)$$

(c) the log of the price of home output of the commodity:

$$(1-\beta_2) \log PQ$$

(d) a time trend, $\beta_3 t$; and

(e) a serial correlation adjustment.

Continuing this example, the table in that subsection refers to BPQM. Its first column defines the parameters for any commodity using the equation just described, the parameters being stored in the row of BPQM referring to that commodity. Hence column 1 of BPQM contains regression constants, column 2 foreign price effects, column 3 trend coefficients, column 4 serial correlation parameters, column 5 initial errors and column 6 working constants. Note also that under the FIX option the first column of BPQM contains the exogenous values required, and subsequent columns may be ignored.

Finally, with the example consider the equations at the end of the subsection. These are matrix and vector equations; they use the ' to denote matrix transposition, and * to denote matrix multiplication. In more normal notation let:[1/]

p be the (NQx1) vecotr $\widetilde{\text{PFM}}$
f be the (NMx1) vector PFM
M be the (NMxNQ) vector MQC
μ be the scalar EX
q be the (NQx1) vector PQM
s be the (NMx1) vector PM

The equations then comprise

$$p = M'f$$
$$q = \mu p$$
$$s = Mq$$

1/ We eschew this notation below so that the user may more easily relate the equations to the use and programming of IDIOM.

IMPORTS (Function QM)

Commodity Imports

Imports are determined on a commodity basis (Q-classification) rather than on an import basis (M-classification). This is to ensure that demand for any commodity is precisely balanced by domestic supply, less exports, plus imports. Only world prices of import goods are provided on the import basis.

Four functions are available:

1. LLIN - log-linear model expressing the import ratio as a function of apparent consumption, relative prices and time.

$$\log ZMS = \beta_1 + \beta_2 \log ZAC + \beta_3 \log [PQM * ZQMTØ /PQ] + \beta_4 \log [PQM * ZQMTØ /PQ]_{-1} + \beta_5 t + \rho^{NTD} U_o$$

where

ZMS is the ratio of imports to domestic sales of domestic goods i.e.

$$ZMS = QM/(Q-QX)$$

ZAC is apparent consumption, i.e. ZAC = Q + QM - QX

ZQMTØ is the extent to which tariffs etc. raise the import price in the solution year relative to the base year

$$ZQMT = (1. + QMTØ/QMØ)/(1. + QMTZ)$$

(remember QMTØ is current tax-takings on imports, QMTZ is the overall tax rate on imports in the base period)

At any iteration the basic exogenous variable is ZAC, apparent consumption. From above, ZMS is determined by it, and then, holding ZAC constant, we use ZMS to calculate a new allocation between Q and QM. Hence:

$$Q = QX + \left[ZAC/(1 + ZMS)\right]$$

$$QM = \overline{ZAC} - (Q - QX)$$

Up to this point QM has been measured inclusive of (base-year) tariffs. IDIOM actually stores it exclusive of tariffs, however, so the final step for IDIOM is:

$$QM\emptyset = QM/(1. + QMTZ)$$

Note: The function is homogeneous of degree zero in (tariff-inclusive) prices. The economic arguments for such functions as these are found in: Barker (1970), Barker ed. (1976), and Barker (1977).

Briefly the assumption is that each IDIOM commodity treated like this comprises several differentiated varieties of goods. The bundles of home sales and imports differ in the varieties they contain, and hence there is scope (i) for persistent price differences between home sales and imports and for these price differences to affect relative demands, and (ii) for the income elasticities of home sales and imports to differ, because for instance imports may comprise higher quality goods. The income elasticities are reflected above by apparent consumption, i.e. the domestic economy's total absorption of the good in question, and it is plain that even without price changes, the import share can vary if apparent consumption does.

The reason for expressing the dependant variable as the ratio of imports to domestic sales of domestic goods is that, unlike the ratio of imports to apparent consumption, this variable is not bounded above. Hence, as given, the equation cannot predict imports exceeding domestic absorpiton.

It is plain that this type of import-function is only really suited to modelling competitive imports; i.e. imports for which there is a corresponding domestic output which is capable of supplying some non-trivial proportion of the market.

2. FIXQ - domestic output, by commodity, is fixed. Imports make up the difference between this and total demand. Hence:

$$QM = QC + QS + QV + QG + QX + QY - \overline{Q}$$

This is essentially for complementary imports, where domestic output is fixed (often at zero). Its use is also discussed by Barker, especially in Barker (1970).

3. FIXM - imports are fixed; domestic output makes up the differences between this domestic demand.

$$Q = QC + QS + QV + QG + QX + QY - \overline{QM}$$

Note that Q and QM are intimately related. You cannot fix both independently.

4. QMQ - imports are determined largely by 'quota variables'.

$$QM = \beta_1 + \beta_2\ QMQ + \beta_3\ QMQ_{-1} + \rho^{NTD}\ U_o$$

For simple, binding quotas, one could use this function with $\beta_i = 0$, except for $\beta_2 = 1$, or use the FIXM option. This function, however, allows much greater flexibility, and since the variable QMQ is used nowhere else in IDIOM, it may be filled with any import determining variable you choose.

NOTE: Q and QM are jointly determined to satisfy total demand. You can not fix them independently of each other. Each, however, is subject of a non-negativity constraint. If either emerges from the above negative, it is set to zero and a corresponding reduction made to the other (e.g. if from above Q = 10 and QM = -2, IDIOM would fix the result such that Q = 8 and QM = 0).

Estimation should be straight forward for the import functions, provided the data are available. Note that the prices include all taxes levied on imports, the 'operative tariff rate' being calculated as (receipts/imports at constant prices). Note also that the 'income' or 'activity' effect is related to commodity specific apparent consumption. Hence data are reqired on this basis. QMØ is measured exclusive of tariffs, but whenever it is compared with, or added to Q, QX etc. tariffs are included. Hence ZAC, ZMS etc. are measured with tariff-inclusive imports, and the LLIN functions must take this into account.

Other Imports

In addition to commodity imports there are facilities for handling "direct imports" by industry, government and consumers. These are part of value-added and are fully described under that heading below. The calculations they entail are handled in the routines for PY, PG and PC respectively.

IMPORT PRICES (Function PQM)

Import prices (and imports) are determined directly by commodity (Q - classification),1/ and converted to the import classification by the converter MQC. Only the foreign prices of import goods appear directly on the import classification.

Four functions are available:

1. LLIN - log-linear functions incorporating foreign prices (in domestic currency), local prices of the good concerned, and time:

$$\log PQM = \beta_1 + \beta_2 \log (P\tilde{F}M/EX) + (1 - \beta_2) \log PQ + \beta_3 t + \rho^{NTD} U_o$$

1/ The purpose of this is to facilitate the situation where imports are non-competitive and are determined solely as the excess of domestic demand over an exogenously given domestic supply.

Import Parameters BQM

BQM is (NQ, NBQM)

parameter \ option	LLIN	FIXQ	FIXM	QMQ
1	r.c	$\bar{Q}$	$\overline{QM}$	r.c.
2	β (ZAC)	-	-	β (QMQ)
3	β (prices)	-	-	β (QMQ_{-1})
4	β ($prices_{-1}$)	-	-	ρ
5	β (t)	-	-	U_o
6	ρ	-	-	-
7	U_o	-	-	-
NBQM	w.c.	-	-	w.c.
Value of:				
JSWQM	1	2	3	4
NBQM	8	1	1	6

Notes:

(a) PFM has to be converted to $\widetilde{PFM}$ before use

(b) The function is necessarily homogenous of degree one in prices (in domestic currency terms)

(c) These prices are prior to the imposition of tariffs. The economic logic behind this function is that local import prices will reflect both the world price (the opportunity cost of selling in this market) and the domestic price (the price of competing goods), with higher weight to the latter, the larger the domestic market and the more highly differentiated the goods. The trend might represent product changes, the different weights of particular goods in the price indices, etc.

This equation could be interpreted as a reduced form equation derived from a structural model of the import market.

2. <u>CM</u> - complementary prices. Log-linear functions as above, but with the GDP deflator at market prices (home unit costs, HUC) replacing the local price of the particular commodity. The rationale is that even if our economy does not produce the good in question, local conditions, as represented by HUC, could still influence the price of imports.

3. <u>FIX</u> - fixed exogenously in domestic currency

4. <u>PFM</u> - fixed equal to world prices and converted to domestic currency using EX.

Import price models of these sorts are described in Barker (ed)(1976) or Lleywellyn (1974).

Estimation and data are straight-forward except for the problem of simultaneity. This might relate import prices and home prices mutually and/or import prices and imports. It would require substantial effort to build a full structural model of these relationships, and overall it is probably not worth while. Note that to use this formulation you need to distinguish the world price of the import good from the import price facing your country. If this is not possible (even crudely) it is probably best to use the PFM option. In fact for most countries the difference between PFM and PQM is likely to be small most of the time (i.e. most countries have little market power over their imports), so the PFM option will probably be the most useful anyway.

$$\underline{\widetilde{PFM}} = [MQC]' * \underline{PFM} \quad \text{(conversion to Q-classification)}$$
$$\underline{PQM} = \underline{\widetilde{PFM}} * EX \quad \text{(PFM option)}$$
$$\underline{PM} = [MQC] * \underline{PQM} \quad \text{(conversion from Q to M-classification)}$$

Unfortunately this does not guarantee that PM = PFM * EX unless [MQC] is an identity matrix.

The above refers to commodity imports. There are also certain direct imports by industry, government and consumers, that count as part of value-added. These are described under value-added below, but here we note that the prices of these imports have to be set exogenously in local currency terms. There is a single price for all imports by industry (PSYM), by government (PSGM) and consumers (PSCM), and these need to be defined at the beginning of an IDIOM run and then redefined whenever the user wishes them to change.

Import Price Parameters: BPM or BPQM

BPM has dimension (NQ x NBPM)

parameter \ option	LLIN	CM	FIX	PFM
1	β (r.c.)	β (r.c.)	$\overline{PQM}$	–
2	β ($\widetilde{PFM}$)	β (HUC)	–	–
3	β (t)	β (t)	–	–
4	ρ	ρ	–	–
5	U_o	U_o	–	–
NBPM	w.c.	w.c.	–	w.c.
value of:				
JSWPQM	1	2	3	4
NBPM	6	6	1	1

DOMESTIC PRICES (Function PY)

These functions describe the formation of domestic prices on a commodity and industry basis and also industry profits. Prices are formed basically on a commodity basis, but using, in part, industry - basis data. Where the commodities and industries precisely correspond there is a wider range of options available.

Note: 1) the number of industries must not exceed the number of commodities for an IDIOM run. Additionally each industry must have a principal product and each commodity must have a "principal producer" - the industry whose behaviour and variables control the pricing of that commodity. IDIOM is informed of the link between industries and commodities via function Q, see below.

2) all prices are index numbers.

The following functions are available whether or not there is a perfect correspondence between industries and commodities. They define the home price of home sales (PQHH), before the imposition of any tax.

1. PQMF The domestic price of domestic goods is set equal to the import price:

   PQHH = PQM

2. LMAT - linear functions of all material costs and labour costs and of time. This is essentially the 'dual' price, although the profit mark-up, assumed not to change as a proportion of costs since the base year of the index numbers, has to cover all rents, non-labour taxes and profits. It is also not necessary that the weights on the various elements of costs equal their shares in total input costs. For commodity I, the 'principal producer' of which is industry J:

$$PQHH(I) = \beta_1 + \beta_2 t + \beta_3 ULC(J) + \beta_4 ZMAT(J) + \rho^{NTD} U_o$$

where ZMAT(J) is material costs per unit output for industry J, i.e.

$$\underline{ZMAT} = [QYC]' * \underline{PQH}$$

This is straight-forward cost-plus pricing with a constant mark-up. It may be hypothesized to arise from oligopolistic market structures, which often show such stability. The ULC variable is endogenous to IDIOM, and may represent either actual unit labour costs or normalized unit labour costs.

3. LKEY - linear function of key input costs and unit labour costs. This is similar to the previous case, but instead of linking prices to the cost of all materials, they are linked to the prices of three "key" material inputs. For imperfectly competitive markets, collusion may be enhanced if each firm prices according to certain objective indicators that are observed by both itself and its rivals. (Such price changes cannot be interpreted as aggressive if all firms agree the reason for them, and so price wars are unlikely). The equation is:

$$PQHH(I) = \beta_1 + \beta_2 t + \beta_3 ULC(J) + \beta_5 PQH(\beta_4) + \beta_7 PQH(\beta_6) + \beta_9 PQH(\beta_8) + \rho^{NTD} U_o$$

hence β_4 defines the commodity which is the first key input and β_5 its weight in commodity I's pricing. Similarly for β_6 and β_7 and β_8 and β_9.

If not all three key inputs are required, set the redundant values of β_i to zero. Also it is possible to use the import price of the key input rather than the home price. This is done by defining the relevant β_4, β_6 or β_8 as minus the index of the key commodity.

While the key input hypothesis has some intuitive appeal - especially for the short-run - it did also present some problems in Cambridge. If, over the long-run, the key input prices do not behave like other commodity prices, it can result in highly implausible output prices and profits.

4. LLM as option 2 LMAT, but with the log of PQHH related to the logs of ULC and ZMAT.

5. LLK as option 3 LKEY, but with the log of PQHH related to the logs of ULC and of the key inputs.

Restrictions

1. It is possible to use key inputs for export prices, but for any export price doing so, every commodity entering that export commodity (as defined by the export-commodity converter QXC) must have its home price also set by the key input method. Furthermore, both export and home prices must use the same key inputs and the same weights, except that for exports the weights may be multiplied by a constant (see export prices, below).

2. All commodity prices must be set either by methods 1 to 5 above, or by methods 6 and 7 below; mixing across these two groups is not permitted.

From PQHH certain other prices and variables can be built up:

(i) PQ - the price of gross output by commodity is built up using both export and domestic data:

$$PQ = ZA * PQX + (1 - ZA) * PQHH$$

where ZA is the share of exports in gross output

$$ZA = QX/Q$$

(ii) PQH - the price of domestic absorption is built up using data on imports and PQHH

$$PQH = ZM * PQM + (1 - ZM) * PQHH$$

where ZM is the import share in domestic absorptions

$$ZM = QM/(Q + QM - QX)$$

(iii) PY - the price of gross output of industry. Current price output by industry (ZYP) is calculated from current price output by commodity (ZQP) using converter |YQC|, the make matrix:

$$\underline{ZPY} = [YQC] * \underline{ZQP}$$

and industry prices as:

$$PY = ZPY/Y$$

(iv) profits - calculated as total value of output less material costs and all other elements of value-added.

$$YP = ZPY - ZMAT * Y - ZYR\emptyset$$

where ZMAT is unit material costs (hence ZMAT * Y is total material costs)

ZYRØ is YRØ - the industry's value added payments in total - excluding profits.

The remaining two price formation methods work directly on industry and are only available if NQ = NY and FUNCTION a is IDEN (i.e. there is an identity between commodities and industries).

6. PVA - prices for net output are determined by log-linear functions

$$\log ZPVA = \beta_1 + \beta_2 t + \beta_3 \log YULC + \beta_4 \log ZYT + \beta_5 \log PQM + \rho^{NTD} U_o$$

where ZPVA is the price of value-added

YULC is unit labour costs (from YULC functions)

ZYT is tax per unit of output

$$ZYT = YTØ/Y$$

PQM is the price of <u>competing</u> imports.

The price of value added is then combined with material prices to give industry prices. Hence:

$$PY = ZPVA * (1 - ZMT) + ZMAT$$

where ZMT is the share of materials in gross output

$$ZMT_j = 1 - \sum_i QYC_{ij}$$

ZMAT is unit material costs.

Finally industry profits are derived as:

$$YP = (PY - ZMAT) * Y - ZYRØ$$

7. <u>PRS</u> - profit share functions

$$ZPS = \beta_1 + \beta_2 t + \beta_3 (Y/YEXP) + \rho^{NTD} U_o$$

$$YP = ZWAG * ZPS/(1. - ZPS)$$

where ZWAG is the industry wage bill (hence it is the share of wages plus profits that ZPS refers to) and

$$PY = (YP + ZMAT * Y + ZYRØ)/Y$$

From the industry prices just derived, IDIOM calculates commodity prices as follows:

$$\underline{PQ} = [YQC]' \underline{PY}$$

$$PQHH = (PQ * Q - PQX * QX)/(Q - QX)$$

$$PQH = (PQ * Q - PQX * QX + PQM * QM)/(Q - OX + QM)$$

Note: These notes describe the present position so far as price formation is concerned. They seem to imply a somewhat unfortunate feature. This is that with the exception of the PVA option, taxes on industry do not affect prices, but merely reduce profits. It is a relatively simple matter to adjust this and I suggest that is done relatively soon.

Prices of domestic absorptions

Government current expenditure, investment (including government), consumption and stock-building all absorb goods from the domestic market. Their prices, on their own classifications, are all formed in the following manner as accounting operations. For the sake of an example we take consumption prices:

$$\underline{PC} = [QCC]' * \underline{PQH} + [\hat{C}]^{-1} \underline{CRØ}$$

Prices comprise a material inputs part, calculated via the converter, and a value-added part which equals total value added generated by the commodity in question, divided by consumption. Note, therefore, that the converter QCC does not necessarily have column sums of unity; they are, in fact, unity less value-added per unit consumption (in the base year).

PG is formed percisely analogously to above, and PYS analogously except that stock-building generates no value-added (hence there is no second term in the equation). For PV the process is as above for all assets except plant, where a separate converter exists. The principle is the same even here, however, that the price is equal to unit material costs plus unit value-added costs. The only difference is that for each industry individual materials components and taxes on materials can be calculated. Hence:

$$\underline{PYVP} = ([QYPC]' * \underline{PQH}) * (I + [\hat{ZIT}]) + \underline{ZPV} * ZVS$$

where QYPC is the industry plant - commodity converter

ZIT is the rate of tax on industry's investments in plant materials (Hence the factor $(I + [\hat{ZIT}])$ grosses up material demands to account for tax.)

ZPV is current price investment in plant (by industry)

and ZVS is a scalar representing value-added per unit of investment in plant averaged over industries. The value-added components of plant investment are not disaggregated by industry but are handled in aggregate, as for other assets, via RVB etc. Hence in the above the total value-added from plant investment is shared among industries according to current price investment.

THE DOMESTIC PRICE PARAMETERS BPQ

BPQ has dimension (NY, NBPQ)

option / parameter	PQMF	LMAT LLM	LKEY LLK	PVA	PRS
1	-	β(r.c.)	β(r.c.)	β(r.c.)	β(r.c.)
2	-	β(t)	β(t)	β(t)	β(t)
3	-	β(ULC)	β(ULC)	β(ULC)	β(Y/YEXP)
4	-	β(ZMAT)	$\beta(PQ(\beta_5))$	β(ZYT)	ρ
5	-	ρ	1st key input 1/	β(PQM)	U_o
6	-	U_o	$\beta(PQ(\beta_7))$	ρ	-
7	-	-	2nd key input 1/	U_o	-
8	-	-	$\beta(PQ(\beta_9))$	-	-
9	-	-	3rd key input 1/	-	-
10	-	-	ρ	-	-
11	-	-	U_o	-	-
NBPQ		w.c.	w.c.	w.c.	w.c.
value of JSWPQ	1	2,4	3,5	6	7
NBPY	0	7	12	8	6

1/ If you are using PQM rather than PQ, set these numbers negative.

INDUSTRIAL INVESTMENT (Function YV)

Industries' investment is determined by both industry and asset, so that the industrial investment variable is an (NY, NV) matrix. Five options are permitted.

1. NCL: a neo-classical investment function, based on Jorgenson (1963). Investment is derived from a model of long-run cost minimization as a function of the price of investment relative to that of output (this proxies relative factor prices), the change in output, depreciation, and time.

$$\log YV = \beta_1 + \beta_2 t + \rho^{YEAR} U_o$$

$$+ \beta_{12} \log \beta_8 ZPV + \beta_9 ZPV_{-1} + \beta_{10} ZPV_{-2} + \beta_{11} ZPV_{-3}$$

$$+ \beta_7 \log \beta_3 ZY + \beta_4 ZY_{-1} + \beta_5 ZY_{-2} + \beta_6 ZY_{-3}$$

where ZPV is the "effective" price ratio of investment to output

$$ZPV = PVA * PV/PY^{1/}$$

where PVA is the rate of effective investment allowances for industry I's investment in asset J. [PVA] is (NY, NV)

and ZY is the effective change in output as measured at time Υ

$$ZY_\Upsilon = Y_\Upsilon - \beta_{13} Y_{\Upsilon-1}$$

Hence β_{13} is a depreciation parameter reflecting the fact that without further investment the capital that produced $Y_{\Upsilon -1}$ at time Υ-1, would produce only $\beta_{13} Y_{\Upsilon-1}$ at time Υ.

1/ ZVP is calculated for years 0 and -1 from the basic data, but for years -2 and -3, it is stored directly as PYVL1 and PYVL2 respectively. Hence in the updating for, say, 1977, PYVL1 contains data for 1975; if 1977 were the first projection year, PV, PY, PVA etc. would be initialized to 1976 values and PYVL1 to 1975 values etc.

The estimation of this function in full is complex, for it is non-linear in parameters and the data likely to be highly collinear.[1/] If facilities do not exist for non-linear estimation probably the best procedure is to fix the lag weights and the depreciation rate a priori. Both will vary according to the industry concerned, and both will affect the dynamics of the system. Hence care should be exercised.

2. NCLP: This applies the Jorgenson model to plant alone. Economically it is the same as the previous case but there is a practical difference. Because plant varies so much from industry to industry IDIOM allows the user to distinguish industry-specific prices of plant and machinery (PYVP). This function substitutes these for the common price of plant in PV that option NCL uses to form ZPV. The PYVP are formed automatically within IDIOM using domestic absorption prices (PQH), the plant-commodities converter (QYPC) and indirect taxes on plant investment.

The converter QYPC is also used to translate investment in plant back into commodity demands (as part of QV), regardless of whether NCLP or any of the other functions were used to determine it.

3. ACC: linear function of output, output lagged and time

$$YV = \beta_1 + \beta_2 t + \beta_3 Y + \beta_4 Y_{-1} + \rho^{YEAR} U_o$$

If $\beta_4 = -\beta_3$ this is a simple accelerator with no replacement and if $\beta_4 = \delta - \beta_3$ it is an accelerator with replacement at δY p.a. (Note that depreciation is here related to output rather than capital stocks, but this makes little difference. If depreciation is at the rate γK, where K is the capital stock, this is equivalent to γMY, where M is the average capital output ratio (= K/Y), hence $\delta = \gamma M$).

1/ There are also likely to be identification problems between β_{12} and β_8 to β_{11} and between β_7 and β_3 to β_6.

INVESTMENT PARAMETERS (BYV, or in some routines YVB)

BYV has dimension (NY, NV, NBYV)

paramters \ function	NCL or NCLP	ACC	FIX	PROF
1	β (r.c.)	β(r.c.)	$\overline{YV}$	β (r.c.)
2	β (t)	β(t)	-	β (t)
3	β (ZY)	β(Y)	-	β (YP/PV)
4	β (ZY_{-1})	β(Y_{-1})	-	-
5	β (ZY_{-2})	-	-	-
6	β (ZY_{-3})	-	-	-
7	β $\lvert \log (\Sigma \beta . ZY) \rvert$	-	-	-
8	β (ZPV)	-	-	-
9	β (ZPV_{-1})	-	-	-
10	β (ZPV_{-2})	-	-	-
11	β (ZPV_{-3})	-	-	-
12	β log (ΣβZPV)	-	-	-
13	β (depreciation)	-	-	-
14	-	-	-	-
15	w.c. 1/	-	-	-
16	w.c. 2/	-	-	-
NBYV - 2	ρ	ρ	-	ρ
NBYV -1	U_o 3/	U_o 3/	-	U_o 3/
NBVY	w.c.	w.c.	-	w.c.

value of				
JSWYV	1 or 2	3	4	5
NBYV	19	7	1	6

1/ This w.c. accumulates the exogenous component of the output term within the UPDATE phase; i.e. all elements referring to lagged variables.

2/ This w.c. accumulates the exogenous component of the price term within UPDATE.

3/ This variable is updated once per year by multiplying by ρ in the UPDATE phase. This takes place before it is added into the function. Hence if we know the error for the first year of the projection period - U* say - U_o should be initialized to (U*/ρ). Then the error process will generate U* in year 1, ρ U* in year 2 etc. If the error in year 1 is not known, initialize U_o to the error (actual or predicted) in the year preceeding year 1.

The economics of the accelerator are crude, to say the least, but the function is economical of data and estimation effort.

4. FIX: investment is fixed to an exogenous value:

$$YV = \beta_1$$

5 PROF - investment is related to time and real profits. This is a rather degenerate neo-classical investment function.

$$YV = \beta_1 + \beta_2 + \beta_3 YP/PV + \rho^{YEAR} U_o$$

INDUSTRIAL EMPLOYMENT (Function YE)

For each industry these functions determine a single, aggregate, level of employment. Disaggregation into types of labour is carried out later by means of an employment-industry converter EYEC. Hours as well as men are also calculated. Five functions are permitted.

1. RECU: A recursive model of the production sector. Assuming investment (the capital stock) has already been determined on the basis of the investment functions, these functions carry out a second, short-run, cost-minimization process with respect to the variable factors men and hours. The full rationale behind this sophisticated approach is discussed in Peterson (1978). Briefly, the argument is as follows:

   a) Labour "effort" (ZEE) supplied comprises three elements: employment (in numbers), YE; hours worked (in a week), YH; and effort or quality ZJ (non-measurable). Hence:

$$ZEE = YE * YH * ZJ$$

b) Labour effort demanded results from a cost minimization exercise taking the capital stock and investment as given:

$$d \log ZEE = \alpha_o + \alpha_1 \, d \log Y + \alpha_2 \, d \log ZK$$

where d is the differential operator
and ZK the capital stock

But $d \log ZK = (YV\emptyset - ZR)_{/ZK} = v \, (YV\emptyset/Y) - \delta$

where YVØ is total investment
ZR is replacement investment
v is the capital-output ratio
and δ is the rate of depreciation

c) Labour effort or quality is assumed to be related to unemployment (unemployment either persuades workers to work harder, or the first workers to be sacked are the worst, hence rising unemployment raises the average quality):

$$\log ZJ = \alpha_3 \log UE$$

d) There is some notion of "standard hours" (ZSH) to which actual hours are gradually converging. Standard hours are fixed exogenously and the convergence might never be completed in any IDIOM time-scale:

$$d \log YH = \alpha_4 + \alpha_5 \, d \log ZSH + \alpha_6 \, (\log ZSH_{-1} - \log YH_{-1})$$

The last term represents the rate of "catch-up". Assuming α_6 is positive, the larger the difference between standard and actual hours, the greater the change in actual hours.

e) There is also partial adjustment of actual employment towards the desired level of supply. Hence:

$$d \log YE = (1-\gamma)\ d \log (ZEE/YH*ZJ) + \gamma d \log YE_{-1}$$

where the first term comes from (a) above.

In the simulation model, IDIOM, these equations are combined to yield predictions of YH, ZSH and YE, although ZSH is not passed to the rest of the model:

$$d \log YH = \beta_1 + \beta_2 (\log YH_{-1} - \log ZSH_{-1}) + \beta_3\ d \log ZSH + \beta_4 \log UE \qquad (1)$$

where the effect of ZJ (effort or quality) has been submerged into YH. That is, it is implicitly assumed that extra effort is reflected in fewer hours but no loss of output. An alternative justification of this approach would be to relate the change in hours in paragraph 4 above directly to unemployment rather than via the unobservable ZJ. Of course the IDIOM user can always ignore the unemployment effect by setting $\beta_4 = 0$.

$d \log ZSH$ is stored as β_5

and ZSH_{-1} is stored as β_6, and is updated automatically by the programme, <u>after</u> use in the function. Hence on initialization β_6 should contain standard hours for the year preceding the first projection year and β_5 the assumed proportionate rate of change in standard hours. This is the trend rate of change of standard hours and must be constant through the projection exercise.

$$\left.\begin{aligned} d \log ZEE &= \beta_7 + \beta_8 \log Y + \beta_9 \log Y_{-1} + (\beta_{10} - 1) \\ &\quad \log YE_{-1} + \beta_{11} (YV\emptyset/Y) + \beta_{12} (YV\emptyset/Y)_{-1} \\ \log YE &= \log YE_{-1} + d \log ZEE - d \log YH \end{aligned}\right\} \qquad (2)$$

Obviously the penultimate equation could be substituted into the last one, and the coefficient β_{10} is formed assuming it has been: β_{10} covers both the partial adjustment effect $(\beta_{10} - 1) \log YE_{-1}$ in the ZEE equation and the unit coefficient arising in the last equation because: $\log YE = \log YE_{-1} + d \log YE$.

In the simple case where labour/output ratios are constant, we can set $\beta_8 = -\beta_9 = 1$, in which case $d \log YE = d \log Y + \ldots$

The estimation of these functions is demanding of data, but the system is linear; it should be reformulated so that YE and YH are the dependent variables (rather than the unobservable ZEE).

Clearly this system determines not only employment, but productivity; Hence the latter is likely tc vary.

2. LLIN: log-linear function of output and time.

$$\log YE = \beta_1 + \beta_2 t + \beta_3 \log Y + \beta_4 \log Y_{-1}$$

$$+ \beta_5 \log YE_{-1} + \rho^{NTD} U_o$$

With lagged exogenous and endogenous variables this allows moderately sophisticated dynamics. Note also that there could be both numerical and interpretive difficulties if the full unrestricted equation were estimated allowing for serial correlation, for even without the $\rho^{NTD} U_o$ term, the non-linear parameter restriction $\beta_5 = -\beta_4/\beta_3$ is equivalent to estimating:

$$\log YE = \gamma_o + \gamma_1 t + \gamma_2 \log Y$$

allowing for auto-correlation. Hence the quoted equation is a second order dynamic equation with one common factor, see Hendry and Mizon (1978).

As before, this equation endogenizes productivity.

3. LLWA: log-linear function as above but with an additional term reflecting real wages (from the producers' viewpoint):

$$\log YE = \beta_1 + \beta_2 t + \beta_3 \log Y + \beta_4 \log Y_{-1} + \beta_5 \log YE_{-1}$$

$$+ \beta_6 \log (ULC * Y/(YE * PY)) + \rho^{NTD} U_o$$

4. EFIX: employment is fixed; hence productivity is endogenous

$$YE = \beta_1$$

5; PFIX: productivity is fixed; hence employment is endogenous

$$YE = \beta_1 Y$$

STOCK-BUILDING BY INDUSTRY (Function YS)

These functions determine stock-building by the industry under-taking it. Stock-building is not disaggregated, and so includes raw materials, work in progress and final output; it is sometimes refered to as working capital. Four options are available.

1. FULL: Stock-building is determined by a full econometric model from data on an industry basis: the latter includes output, the price of stocks, the rate of interest, the level of stocks and time.

   Industries hold stocks in the long-run to facilitate the production process, for instance, by smoothing production schedules, allowing for machine break-downs etc. Hence stocks are, in equilibrium, related to output. They are not costless, however, for they tie up financial capital, and so the stock-output relationship is likely to be moderated by the rate of interest. We also allow it to be moderated by a time-trend to reflect the changing technologies of both production and stock management. Hence in the long-run:

$$S = \alpha_1 + \alpha_2 t + \alpha_3 Y + \alpha_4 RINT$$

The Industrial Employment Parameters (BYE or BYEØ)

BYE has dimension (NY, NBYE)

parameter \ option	RECU (1) (2)		LLIN	LLWA	EFIX	PFIX
1	β (r.c.)	(H)	β (r.c.)	β (r.c.)	$\overline{YE}$	β (Y)
2	β (YH - ZSH)	(H)	β (t)	β (t)	-	-
3	β (ZSH_{-1})	(H)	β (Y)	β (Y)	-	-
4	β (UE)	(H)	β (Y_{-1})	β (Y_{-1})	-	-
5	d log ZSH	(H)	β (YE_{-1})	β (YE_{-1})	-	-
6	ZSH_{-1}	(H)	ρ	β (wage)	-	-
7	β (r.c.)	(E)	U_o	ρ	-	-
8	β (Y)	(E)	-	U_o	-	-
9	β (Y_{-1})	(E)	-	-	-	-
10	β (YE_{-1})	(E)	-	-	-	-
11	β (YVØ/Y)	(E)	-	-	-	-
12	β ($(YVØ/Y)_{-1}$)	(E)	-	-	-	-
$NBYE_{-1}$	w.c.	(E)	-	-	-	-
NBYE	w.c.	(H)	w.c.	w.c.	-	-
value of						
JSWYE	1		2	3	4	5
NBYE	14		8	9	1	1

1/ An 'H' in brackets denotes that co-efficients apply to the hours equation,(1)above, and an 'E' denotes that they apply to be employment equations,(2) above.

2/ Most variables in this equation need transforming - see text.

In the shorter-run, two other features may affect the level of stocks. First, they will be sensitive to changes in prices through time - a speculative element - and, second, they will be affected by unanticipated changes in demand or output. We capture these effects crudely by including in the equation a term:

$$+ \alpha_5 (PYS/PYS_{-1})$$

and breaking the income effect into:

$$+ \alpha_3 Y + \alpha_6 Y_{-1}$$

The signs of these coefficients are not well defined by theory, for they represent a series of effects.

Finally, we should recognize that the relationship above represents an ideal, or desired, stock-level. If there are costs to adjustment we might expect lagged responses, and these may be represented by including a Koyck lag variable:

$$+ \alpha_7 S_{-1}$$

in the equation. Note, however, that the interpretation of this lag may convey information on the error structure of the final equation; if it is as a partial adjustment, no error-transformation takes place, but if it is as adaptive expectations, white noise in our basic equation becomes auto-correlated when S_{-1} is added - see Intriligator.

The final estimation equation combines the above effects, and refers to stock-building $(S - S_{-1})$ rather than stock-levels:

$$YS = \beta_1 + \beta_2 t + \beta_3 Y + \beta_4 RINT + \beta_5 (PYS/PYS_{-1}) + \beta_6 Y_{-1} + \beta_7 ZS_{-1}$$

where ZS is the stock-level at the end of period -1.

ZS is stored in BYS (column 8) and is updated automatically by the programme. The updating occurs before the function is calculated, so ZS ($=\beta_8$) should be initialized to ZS_{-2} - i.e. stocks at the end of year -2 - and YS to YS_{-1} - i.e. stock-building in year -1. Then stocks at the end of year -1 are formed $ZS_{-1} = ZS_{-2} + YS_{-1}$ and used in the equation.

PYS_{-1} is also stored in BYS; it is updated after the function has been formed. It may be initialized to the stock price in year -1 or year -2, because the value for -1 (the value in the function) is automatically inserted by the programme, which takes it from the initialized value of PYS.

2. ACC: accelerator model: simple function of the changing output but with a time variant parameter.

   If stock levels are related to output, stock-building is related to the change in output. The ratio, however, is not constant:

$$YS = \beta_1 + \beta_2 t + (\beta_3 + \beta_4(YEAR-1))(Y - Y_{-1})$$

   β_4 is the rate of change of the accelerator coefficient. It is used to update the value stored in β_3 every year. The updating occurs after the function is formed so β_3 should be initialized to the value required for the first year of projection.

3. FIX: fixed stock-building

$$YS = \beta_1$$

4. NULL: no stock-building

$$YS = 0$$

The values of lagged stock and the lagged stock price are also used in these routines to calculate stock-appreciation. (There is no functional control over this operation, however). Stock appreciation is "levied" on stocks at the end of the previous period (beginning of this); no allowance is made for any appreciation of this year's stocks.

$$YSA = ZS_{-1} * (PYS - PYS_{-1})$$

or in full:

YSA (I) = BYS (I, NBYS-2) * (PYS(I) - BYS (I, NBYS-1))

Stock building by industry is translated into the respective commodity demands by means of the converter QYSC.

Finally, an additional exogenous item of stock-building and stock-appreciation is added - that accruing to government. This is not disaggregated at all and does not enter the commodity balances directly (it does so indirectly via QG). It is merely an element of the accounts of the government sector, calculated to render these more easily compatible with published sources.

CONSUMERS' EXPENDITURE (DISAGGREGATED) (Function DC)

Given total consumption by each income group, these functions split it among the various consumer demand categories. There are ND income groups (defined by the ND tax-brackets) and NC consumer goods. For each income group basically four alternatives exist.

1. LES: The linear expenditure system. This is well known in the econometric literature. It presents certain problems because once it is applied to more than four groups it imposes a rough proportionality between price and income (expenditure) elasticities. It also requires estimation as a system and is non-linear. On the other hand, it is a system that exploits the *a priori* features that

The Stock-building Parameters (BYS)

BYS has dimension (NY, NBYS)

parameter \ option	FULL	ACC	FIX	NULL
1	β (r.c.)	β (r.c.)	$\overline{YS}$	–
2	β (t)	β (t)	–	–
3	β (Y)	β ($Y - Y_{-1}$)	–	–
4	β (RINT)	increment for β_3	–	–
5	β (PYS/PYS_{-1})	–	–	–
6	β (Y_{-1})	–	–	–
7	β (ZS_{-1})	–	–	–
NBYS −2	ZS_{-1}	ZS_{-1}	ZS_{-1}	ZS_{-1}
NBYS −1	PYS_{-1}	PYS_{-1}	PYS_{-1}	PYS_{-1}
NBYS	w.c.	w.c.	w.c.	w.c.
value of				
JSWYS	1	2	3	4
NBYS	10	7	4	3

the economic theory of consumer demand provides. The model is discussed by Deaton (1975) and Deaton and Muellbauer (1980). Unless considerable resources are to be devoted to a study of consumption, my advice is not to use this system, for the effort involved in setting up the system from scratch is not justified by the results.

The basic equation for consumer good i is:

$$DC_i = \gamma_i + (\beta_{1i} + \beta_{2i}t)(ZDSPC - \sum_{j \in LES} \gamma_j PC_j)/PC_i$$

where:

DC_i, γ_i, β_{ei} and ZDSPC all vary over income groups, but PC_j does not, and

ZDSPC is total consumption less expenditure on durables.[1/]

γ_i represents "committed" expenditure on good i and is formed by IDIOM as β_{3i} * DPOP/POP, i.e. the parameter given to the mode s scaled by the proportion of the population lying in this income group before use. Hence β_{3i} - the parameter that is input - represents committed expenditure on good i for the whole country, not just this particular income group. The apparent implication is that committed expenditure is the same for every member of the population; hence an income group with x% of the population accounts for x% of the committed expenditure.[2/] This interpretation must be remembered at estimation stage. Although it appears rather clumsy, the reason for it will become clear when we discuss the UNIF option below.

Another feature to note is that super-numerary expenditure, the surplus remaining after committed expenditures have been made, is calculated by removing from total consumption only durables and commodities explained by the LES. It is as if those non-durable

---

1/ Plus a further adjustment in the case of group 1. See option XFT below.

2/ We say "apparent" because β_3 may differ over income groups, it must just be input in this strange form.

commodities that are explained by the other functions in fact have committed expenditure to zero. This is not a problem at estimation time for then one usually estimates the system complete, but in simulation runs it has a possibly awkward consequence. Suppose we were using LES for all commodities, but that we then decide to move to a log-linear function for, say, commodity 1. Suppose also that the log-linear function is chosen so as to produce exactly the consumption that the LES had given for commodity 1. One might expect that the composition of consumption would be unchanged by this; but it is unfortunately not so. The super-numerary income passed to the LES in the new run would be higher then in the original, because $\beta_{3,1}PC_1$ would no longer have been removed from it and so consumption on every other commodity would be predicted to rise relative to before by $(\beta_{1i} + \beta_{2i}t)$ times the change in super-numerary income. When we add the consumption of good 1 (unchanged by construction) the sum of consumptions will now be higher than before. IDIOM then scales the disaggregate consumptions so they add to the given total, DSC (see below), and so all consumptions will fall by the same proportion. The outcome will be consumption of commodity 1 lower than before, and all others higher. Numerically this problem is unlikely to be large unless committed expenditures are small and commodity 1 has a large budget share, but it does prevent the precise fine-tuning of consumption under such circumstances. Finally note that the LES is not in per capita terms.

2. LLIN: Simple log-linear functions explaining expenditure in terms of total expenditure and relative prices. The basic equation is in per capita terms, and the results grossed up by population.

$$\log DC = \beta_1 + \beta_2 t + \beta_3 \log ZDSCH + \beta_4 \log (PC/DPCE)$$
$$+ \log DPOP + \rho^{NTD} U_o$$

where ZDSCH is real expenditure excluding durables per head. All variables except PC vary over income groups.

There is no "system" here; the equations for each commodity are wholly independent.

3. FIX: expenditure fixed in per capita terms:

$$DC = \beta_1 * DPOP$$

4. DUR: a durables equation based on stock adjustment. Basically in per capita terms, but grossed up by population:

$$DC = DPOP * \{\beta_1 + \beta_2 t + \beta_3 ZDRH + \beta_4 PC * HPD/DPCE$$

$$+ \beta_5 (DC/DPOP)_{-1} + \rho^{NTD} U_o$$

where ZDRH is real personal disposable income for this income group.

and HPD is hire purchase variable that modifies the price response. It has dimension (NC x 1) and is not used anywhere else in the model. It is set exogenously by the user and remains unchanged until a new value is read in explicitly. If it is not required, set it to unity.

After the disaggregated consumptions have been calculated for each income group above, they are scaled so that their totals equal the total consumptions for the groups that were calculated by the consumption functions. Hence even the fixed values may get altered slightly.

There are two further functions under this heading which produce special treatment.

5. XFTA: This is the function for foreign tourists' expenditure.

In the national accounts, consumption refers to consumption by domestic residents and any expenditure in our country by foreigners as exports. IDIOM observes this convention, but it is difficult to apply to disaggregated accounts, for although we can measure tourist expenditure in total, we can never get it broken down by consumption good. Hence the data on disaggregated consumption include expenditure by both residents and non-residents. One solution would be to guess the non-resident portion for each good and subtract it before estimating the equations described above. Another, however, is to leave foreigners' expenditure in the consumption vector, but to define an additional class of expenditure - foreign tourists' expenditure - which contains the negative of total tourists' expenditure. In this way the consumption vector sums to the required "residents' consumption", while the disaggregated figures have not been subjected to an arbitrary deduction.

IDIOM approaches the problem this way, and so in addition to the consumption classes normally used, we may also define the extra one, tourist expenditure which receives a negative entry in consumption, and whose absolute value is added into exports. Since it is meaningless to estimate foreign tourists' expenditure by domestic income group, it is possible to use this function only for income group 1.

$$\log XFT = \beta_1 + \beta_2 t + \beta_3 \log WVOL + \beta_4 \log (WPRI * EX/PCE)$$

WVOL is the world income variable:

$$WVOL = \sum_i (\sum_j X_{ij}) AD1_i / \sum_i \sum_j X_{ij}$$

where $AD1_i$ is the income variable for world area i,
and X_{ij} our exports of good j to area i

WPRI is the world price variable:

$$WPRI = \sum_i (\sum_j X_{ij}) PW_i / \sum_i \sum_j X_{ij}$$

where PW is domestic prices in world area i

$$DC_{1,k} = - XFT$$

where $DC_{1,k}$ is expenditure by group 1 on good k, and k is the index for foreign tourists' expenditure.

By placing tourists' expenditure in income group 1, we are implicitly maintaining that the disaggregated consumption vector for that group contains two elements: expenditure by domestic residents in income group 1 and expenditure by foreign tourists. This should be recalled when preparing data for estimation and should also enter consideration when choosing functional forms for group 1. It also means that if we are using the LES for group 1, the total expenditure variable should be adjusted to include foreign tourists' expenditure (XFT * PXDT), as well as group 1's total expenditure, DPSC (1). IDIOM does this, but it must also be done for the estimation stage.

This treatment of tourism is perfectly acceptable for countries where it is relatively small, but where it is a major component of GDP, alterations may be necessary. However, in those cases more data should be available.

6. UNIF: The consumption data are disaggregated by income group using the same classification as the direct tax system. Hence if the latter distinguishes ND groups (because, say, there are ND marginal rates), so too must consumption. Often, however, there are insufficient data (or interest) to estimate consumption behaviour for all these groups, and the UNIF option allows the user to simply constrain all groups to behave identically. Specifically for any income group which uses UNIF, the consumption relationship is precisely that used for group 1. Hence UNIF must not be used for group 1.[1/]

The correspondence refers to the parameters of the consumption relationship, not the result, for each income group still has different independent data. This is straight-forward in the case of the per capita functions LLIN, FIX and DUR but there is a slight complication in the case of LES. The parameters β_1 and β_2, which determine each commodity's share of super-numerary income, transfer readily for they are basically unit free, but β_3, the committed expenditure on each good, does have a dimensionality, for it is an absolute amount. Hence if the sizes of income groups K and 1 differ, so should their

1/ See Manual, version 2, release 1, update No. 3.

β_3's just because more people means more committed expenditure. To overcome this problem β_3 is defined throughout as the committed expenditure of the whole economy on each good. Hence assuming that committed expenditures are the same for everyone, the committed expenditures for group k are β_3 *DPOP(k)/POP. In this way if LES were used for group 1 and UNIF for the rest, the sum over groups or committed expenditures would be β_3.

This same convention applies, however, whether or not UNIF is being used and so, in any application of LES, the β_3 required by IDIOM for any group (k) must be set to POP/DPOP(k) times the absolute committed expenditure for that group.

If UNIF is specified for the consumer commodity "foreign tourists' expenditure" it is ignored. XFT can only ever be handled by income group 1.

After forming DC and scaling it to ensure that it sums to DSC, IDIOM calculates total consumption by consumption commodity (i.e. sums over income groups), and using the converter QCC produces a vector of consumer demand by commodity.

The prices of consumption, including that of foreign tourists' expenditure, are formed by the accounting procedure of passing domestic absorption prices through the consumption - commodity converter.

$$\underline{PC} = [QCC]' * \underline{PQH}$$

The group-specific deflators for total consumers' expenditure are formed by weighting PC by the groups' own expenditure patterns.

The Consumer Expenditure Parameters, BDC

BDC has dimension (ND, NC, NBC)

parameter \ option 1/	LES	LLIN 3/	FIX 3/	DUR 3/	XFT
1	β (share)	β (r.c.)	$\overline{DC/DPOP}$	β (r.c.)	β (r.c.)
2	β (Δ share)	β (t)	–	β (t)	β (t)
3	committed expend. 2/	β (ZDSCH)	–	β (ZDRH)	β (WVOL)
4	–	β (PC/DPCE)	–	β (prices)	β (prices)
5	–	ρ	–	β $(DC/DPOP)_{-1}$	–
6	–	U_o	–	ρ	–
7	–	–	–	U_o	–
NBC	w.c.	w.c.	w.c.	w.c.	w.c.
value of					
JSWDC	1	2	3	4	5
NBC	4	7	1	8	5

1/ Option UNIF requires no parameters

2/ Committed expenditure for whole economy, not merely income group in question

3/ per capita

CONSUMERS' EXPENDITURE IN AGGREGATE

THE CONSUMPTION FUNCTION (Function DSC)

These functions determine total consumers' expenditure on goods and services by income group. They use income and wealth as their main explanatory variables, and operate in per capita terms. The main differences between the options concern the accumulation of wealth and the linear vs. log-linear functional form.

THE COMMON ELEMENT

The consumption function for each income group expresses consumption per capita as a function of the group's real personal disposable income per capita (ZRPH), the group's real wealth per capita (ZZWH) and time:

$$ZDSCH = \beta_1 + \beta_2 t + \beta_3 ZRPH + \beta_4 ZZWH + \rho^{NTD} U_o$$

Wealth is accumulated as β_5 in one of the following ways.

1. LEXW: Wealth per capita in nominal terms is exogenous. Its value is given by DPWPH and β_5 is set equal to it every year in the UPDATE phase.

$$ZZWH = DPWPH/DPCE$$

2. LNMW: Wealth per capita is cumulated in nominal terms in DPWPH and β_5 is set equal to it every year in UPDATE.

$$DPWPH = DPWPH_{-1} + \left[(DPDI - DSPC)/DPOP\right]_{-1}$$

$$ZZWH = DPWPH/DPCE$$

DPWPH is updated before being incorporated into the consumption function (via β_5) so it should be initialized to per capita wealth in the year preceding the first projection year.

3. LRLW: Wealth is accumulated in real terms. DPWPH is accumulated in nominal terms, as before, but now β_5 is accumulated separately:

$$\beta_5 = (\beta_5)_{-1} + [(DPDI - DSPC)/(DPOP * DPCE)]_{-1}$$

$$ZZWH = \beta_5$$

As previously initialize β_5 to real wealth in the year proceeding the first projection year.

4, 5, and 6. LLXW, LLNW, and LLRW: These are log-linear versions of LEXW, LNMW and LRLW respectively. Wealth is cumulated as before but in the equation ZDSCH, ZRPH, and ZZWH are all in logarithms.

The final step is to calculate total expenditure and wealth by income group:

DSC = ZDSCH * DPOP

DPW = DPWPH * DPOP

There is no UNIF facility for aggregate consumption. Separate consumption functions are required for each group.

The Consumption Function Parameters BDSC

BDSC has dimension (ND, NBDSC)

parameter	all options
1	β(r.c.)
2	β(t)
3	β(ZRPH)
4	β(ZZWH)
5	wealth
6	ρ
7	U_o
NDBSC	w.c.
value of NDBSC	8

RECEIPTS AND PAYMENTS (Function R)

These are pseudo-functions rather than functions proper. They offer the user no functional control over the calculation of receipts and payment flows, but rather allow him to define which type of flow appears where in the matrix of flows. The disaggregation of flows is defined by the R-classification ($5 \leq R \leq 7$) and these functions merely define whether, say, profits appear in the 1st, 2nd or 3rd column of the payments and receipts matrices. Profits themselves are defined by the model with the help of other variables and parameter matrices.

From the "pseudo-function" definitions it is clear that the bulk of the flows considered are payments to value-added. They are labelled as such in the IDIOM programmes, and they are calculated in the various price routines. Value-added arises from several activities in the economy, so the value-added calculations are correspondingly widespread. Obviously, however, if profits are the first element of the payments matrix for industry, so must they be for households, government etc. Therefore only one set of pseudo-functions is necessary.

We illustrate the various definitions below for industry. They generalize easily to other activities

Industry Payments

These are calculated in IDMSPY and are disaggregated NY industries by NR payments. Only the row and column sums of the payments matrix are stored, however, as RYØ and YRØ which are NR and NY dimension vectors respectively. The parameters used to calculate industrial payments flows are stored in RYB, which IDIOM treats as a converter. Consequently each flow can have, at most, only one parameter. RYB has dimension (NR, NY).[1/]

We now define for each payment type the nature of its formation for industry, and then briefly consider other generators of payments. Note that at this stage we are not considering any receipts. As noted under "Classifications", at least five entries are required in R, one each to be defined as WAGE, PROF, TAX, IMP and GOOD.

1/ Although it may well be read in and stored sparse.

1. WAGE: (JSWR = 1) Wage and salary flows. For each industry (j)

$$Z_j = \beta_j * YE\emptyset_j * AWY * \emptyset.001$$

where:

Z_j is the wage and salary flow for industry j

β_j is the element for industry j in the wages row of RYB

$YE\emptyset_j$ is employment in j } standard IDIOM names

AWY is the average industrial wage }

and the 0.001 arises because per worker variables (like the average wage) are stored in units 10^3 times smaller than aggregates, see "Units in IDIOM" below.

β_j can play the role of a wage differential.

2. PROF: (JSWR = 2)

$$Z_j = YP_j + YSA_j$$

profits + stock-appreciation

No parameter is necessary, hence the corresponding row of RYB may be set to zero.

3. ITAX: (JSWR = 3) - see separate functions T. No entry is required in RYB here either.

4. VOLS: (JSWR = 4) - (optional flow). Value-added at constant market prices is proportional to the volume of output. Hence the income flow is:

$$Z_j = \beta_j * Y_j * HUC$$

This might be used to represent say, a royalty payment.

5. FIX: (JSWR = 5) - (optional flow). Value-added, or payment, is fixed in nominal terms:

$$Z_j = \beta_j$$

6. IMP: (JSWR = 6) direct imports. Imports direct to industry rather than of a commodity, e.g. business travel:

$$Z_j = \beta_j * Y_j * PSYM$$

Direct imports are proportional to output in constant prices. The flow is the constant price amount times the direct import price (common to all industry direct imports).

7. GOOD: (JSWR = 7) The flows arising from the purchase of goods and services; not an element of value-added. This is built up from the real flows calculated elsewhere in the model and entered into the accounts in subroutine IDMSHR. The user may ignore it (no entry in RYB is necessary): it is included to facilitate the presentation and balancing of the income and expenditure accounts.

During these calculations various macro-variables are also calculated:

WAGY cumulates industry's wage and salary bill
SYM cumulates industry's direct imports
SPYM cumulates industry's direct imports at current prices
SYVA cumulates industry's value-added

Other sources of value-added

1. Consumption: calculated by IDMSPC, in vectors RCØ and CRØ using parameters RCB. All functions are as above except that no profits are assumed to accrue directly from consumption; i.e. PROF is ignored. All other flows are feasible, however.

2. Government: calculated by IDMSPG, in vectors RCØ and GRØ, using RGB. Exactly analogous to consumption.

3. Investment: calculated by IDMSPV, in vectors RVØ and VRØ , using RVB. Exactly analogous to consumption.

Example: the input command

FUNCTION R PROF 2 WAGE 1 ITAX 4 GOOD 3 IMP 5

tells IDIOM that in its payments classification element 1 is to be calculated as wages, element 2 as profits etc.; notice that in this example the optional components of value-added (VOLS and FIX) are suppressed. The paramenter in RYB etc. would be stored as follows, and IDIOM would use the following identifiers in JSWR.

	RYB	JSWR	
row	1 wage parameters	1	
	2 profit parameters	2	all zero
	3 goods parameters	7	all zero
	4 tax parameters	3	all zero
	5 import parameters	6	

EXPORTS BY AREA AND GROUP (Function XA)

Exports are determined disaggregated by area of destination (A-classification) and export commodity (X-classification). The basic model is recursive (determining price prior to and independently of current export quantities) and is better suited to trade in differentiated products under imperfect competition than to other market situations. However, competitive price-taking may be forced into the IDIOM mould. The references covering this sort of export function are Winters (1981) and Barker (ed) (1976).

A particular feature of the export functions, which is discussed at length in Winters (1981), is that although exports are disaggregated in two dimensions, the independent data (of which there are many) are disaggregated by only one or the other, see below. The parameters of the export function, however, vary by both area and export commodity; that is, they are specific to each individual flow.

The Export Parameters (BXA)

	Exports of good J to area I are explained by:	Position of coefficient in BXA
1.	a constant	β_2
2.	time	β_8
3.	two separate measures of aggregate demand in I (AD1 and AD2)	β_3 and β_4 respectively
4.	an institutional variable affecting trade with area I (DI1)	β_5
5.	an institutional variable affecting trade of commodity J (DI2)	β_6
6.	the exporters' internal pressure of demand for sector J (Y/EXP converted by QYC to commodities and then by QXC to export commodities)	β_7
7.	lagged exports of J to I	β_9
8.	the exchange rate (EX, EXL1, etc.)	β_{10} to β_{13} lagged 0 to 3 years respectively
9.	the price of exports of J (in local currency) inclusive of tax	β_{14} to β_{17} lagged 0 to 3 years respectively
10.	the price of exports of J (in local currency) exclusive of tax	β_{18} to β_{21} lagged 0 to 3 years respectively
11.	competitors' prices of exports of J (in foreign currency) (PCLØ, etc.)	β_{22} to β_{25} lagged 0 to 3 years respectively
12.	domestic prices (in general) in I (in foreign currency) (PWLØ, etc.)	β_{26} to β_{29} lagged 0 to 3 years respectively
13.	home prices of good J (in local currency, converted to export classification by QXC)	β_{30} to β_{33} lagged 0 to 3 years respectively

Two functional forms are possible - linear and log-linear - but a wide variety of effects are modelled within each. The variables included and the set out of the parameter matrix BXA is given in the table below.

Matrix BXA has dimension (NX, NA, NBXA), where NBXA is always 33. Column 1 for each flow is used as a working constant. No allowance is made for serial correlation, essentially because of the extensive dynamics permitted in the determinate part of the equation. In the log-linear version all variables are logged except the first two.

The two aggregate demand variables for each area allow, say, agricultural exports to be related to a foreign income variable and manufactured exports to an output variable. The definitions of AD1 and AD2 need not be constant across the NA areas; hence, for example, AD2 might be industrial output when I is a developed region, and real export earnings from primaries if I is less developed. Obviously then, however, the coefficients on AD2 are not comparable between areas.

The two variables AD1 and AD2 are perfectly interchangeable in the export functions, but they are not in the rest of IDIOM. The variable AD2 appears nowhere else in the model, but AD1 is used to represent world demand in the export price and foreign tourists' expenditure equations.

The institutional variables can reflect changes, either at home or abroad. Supposing, for instance, that our country joins with those in, say world area 3 to form a customs union. We could set DI1 (3) to zero until the union was formed and unity thereafter, and for each flow set the coefficient β_5 to the effect that the union has on the flow. Probably for all commodities flowing to area 3 $\beta_5 \geq 0$, while for some other areas at least some J would have $\beta_5 < 0$ (export diversion).

The distinction is made here between competition from indigenous producers in our export markets (via PW) and competition from exporters on the world market (PC). This allows some treatment of tariffs in the foreign markets. A non-discriminatory tariff abroad alters the relative price of the importer's domestic goods and all imports (from both us and other suppliers) and hence may be reflected in PW - by reducing it by the extent of the tariff. A discriminatory tariff on imports from us alone correspondingly reduces PW and PC. Note, however, that if these data are manipulated to reflect tariffs artificially, this will also affect the export price formation routines so that a corresponding change in the constant of those equations needs to be made, to ensure that these "artificial" price reductions do not cause us to reduce our export prices.1/

Often data on PW will not be available. If so, set its coefficients to zero. Also it may sometimes not be necessary. The functions here were established to relate our exports to an area directly to that area's macro-economic variables - AD1, AD2, PW etc. There are reasons for preferring this - see Winters (1981) - but a common alternative is to explain our exports to an area with respect to the area's total imports. Hence the export equation essentially explains our share of the area's market and so appears no longer to require access to their macro-data directly. In these circumstances we use AD1 (or AD2) to represent the aggregate imports and PC to represent competitiveness effects; leaving the other area variables blank.

The two export price variables - inclusive and exclusive of taxes - allow the treatment of export taxes. For instance suppose we wished the export function to be homogeneous of degree zero in prices, and to include a "competitiveness" effect (which would include the effects of taxes) and a "profitability relative to home sales" effect (which would exclude taxes from both price series); we would, in the log-linear function, set:

$$\beta_{14+i} = -\beta_{22+i} \qquad i = 0, 1, 2, 3$$

and

$$\beta_{18+1} = -\beta_{30+i} \qquad i = 0, 1, 2, 3.$$

After determining the area x export group flows of exports, IDIOM sums across areas and then passes the export group sub-totals through the export converter, QXC, to get commodity exports QX.

Finally, foreign tourists' expenditure is added to total exports for use in the macro-economic balances. (Because disaggregated tourists' expenditure

1/ Unless of course you believe that we will bear the tariff.

cannot be separated from residents' consumer expenditure, it finds its way into the commodity balances via QC rather than via QX. For macro-economic purposes, however, we shift it across.) Foreign tourist expenditure is discussed under consumers' expenditure, Function DC, above.

EXPORT PRICES (Function PX)

Export prices are determined on the export classification by a combination of foreign (exogenous) data and home data. They are measured in local currency. The main functions are taken from Winters (1981).

Eight options exist. The first six are rather similar and their common features are explained first. Export prices are formed from: a constant, time, world export prices, local prices in our export markets, the exchange rate, domestic pressure of demand, foreign demand and some domestic cost variables. It is only this last element that varies over the first three options, while the second three options merely replicate the first three, but in log-linear rather than linear form.

The common element

$$ZPX = \beta_{16} + \beta_2\ PC + \beta_3\ PC_{-1} + \beta_4\ PW + \beta_5\ PW_{-1} + \beta_6\ EX$$

$$+ \beta_7\ EX_{-1} + \beta_{12}\ PX_{-1} + \beta_{13}\ (\widetilde{Y/YEXP}) + \beta_{14}\widetilde{AD1} + \beta_{15}t$$

where ZPX is the common component of the export price functions

$(\widetilde{Y/YEXP})$ is formed first on an industry basis and then converted via QYC and QXC to the export classification; and

$\widetilde{AD1}$ is converted to the export classification from the area classification using current export weights. Hence:

$$\widetilde{AD1}_j = \sum_i X_{ij}\ AD1_i / \sum_i X_{ij}$$

where X_{ij} is current exports of good j to area i.

The rationale of this part of the function is that export prices are partly determined by their competitive environment the latter comprising competition on world markets from other exporters (PC) and competition from indigenous producers in export markets (PW). The exchange rate merely converts these prices from foreign to domestic currency. In the logarithmic form one would expect:

$$\beta_6 + \beta_7 = - (\beta_2 + \beta_3 + \beta_4 + \beta_5)$$

which would ensure that over the long-run foreign inflation and a devaluation are equivalent.

The demand term reflects the possibility that export prices respond to excess foreign demand, while the local pressure of demand reflects the possibility that, ceteris paribus, when local markets boom export prices either rise (because of the rising opportunity cost of exporting) or fall (because of the spreading of fixed costs).

The lagged endogenous variable is best thought of as reflecting partial adjustment.

1. LPH: Turning now to the specific elements of each function, LPH forms export prices by adding a home price effect to ZPX. Hence:

$$PX = ZPX + \beta_8 \widetilde{PQHH} + \beta_9 \widetilde{PQHH}_{-1}$$

Note that it is the price of domestic sales of domestic output that is used here - the opportunity cost of exporting. It may be justified either as the alternative destination for the output exported, or as a proxy costs variable if domestic prices are cost determined.

2. LUM: This adds costs data to the common element:

$$PX = ZPX + \beta_8 \widetilde{ULC} + \beta_9 \widetilde{ULC}_{-1} + \beta_{10} \widetilde{ZMAT} + \beta_{11} \widetilde{ZMAT}_{-1}$$

where $\widetilde{ZMAT}$ is material input costs for each export good.

$$\tilde{ZMAT} = [QXC]' * \underline{ZMAT}_Q$$
$$\underline{ZMAT}_Q = Q\{[QYC]' * \underline{PQH}\}$$

In words: first, material costs are calculated by industry from the absorption matrix QYC and absorption prices PQH. These are then mapped into commodity space using the function Q{}, which associates each commodity with the material costs of its "principal producer". Finally, the resulting commodity vector $\underline{ZMAT}_Q$ is converted to the export basis by QXC.

LUK: This is as LUM except that key input prices are used for materials rather than all material prices. The key inputs hypothesis is explained under function PY above. If it is used for an export commodity, it must also be used for every domestic commodity that enters that export commodity, as defined by the columns of QXC. Also both export and all domestic prices concerned must use either the linear or the log-linear formulation; mixtures are not permitted:

$$PX = ZPX + \beta_8 \tilde{ULC} + \beta_9 \tilde{ULC}_{-1} + \beta_{10} \tilde{ZKEY} + \beta_{11} \tilde{ZKEY}_{-1}$$

where $\tilde{ZKEY}$ is the key input price indicator by commodity as defined for PQ by functions PY above.

4, 5, 6. LLPH, LLUM, LLUK: These are merely log-linear versions of LPH, LUM and LUK respectively, with all variables converted to logarithms (except time and the constant), after conversion to the export basis.

The log-linear forms allow the simple imposition of homogeneity constraints: Unless there are very strong reasons to the contrary, I recommend imposing homogenity of degree zero in the exchange rate and foreign prices:

$$\sum_{i=2}^{7} \beta_i = 0$$

and homogeneity of degree one in foreign and domestic prices

$$\sum_{i=2}^{5} \beta_i + \sum_{i=8}^{11} \beta_i = 1$$

In general, I should also require β_i $i=2,\ldots 5$ and $i=8,\ldots 11$ to be non-negative.

The estimation of these functions requires considerable amounts of data. In general some a priori restrictions may be required, as in Winters (1981), where:

$$\beta_4 = \mu\beta_2 \text{ and } \beta_5 = \mu\beta_3 \quad \text{with } \mu = 0 \text{ or } 1/2 \text{ only}$$

and $$\beta_{i+1} = \frac{\mu i}{1+\mu_i}\, i \qquad i = 2,4,6,8,10$$

with $\mu_i = 0$, 1/2 or 1 only

The homogenity restrictions can be imposed without the loss of linearity, provided that the data are suitably transformed. Similarly the restriction using μ may be imposed for any exogenously given μ by suitable transformation.

These functions are most suitable for differentiated products where some market power abroad exists, but it is moderated by competitive pressures.

7. PQHH: Export prices equal home sales prices. The case of full market power - i.e. price-making. Because we are using price indices this case covers discriminating monopoly with constant elasticities. Facing two markets, a monopolist sets his absolute prices in year t so that $MC = MR = (1-e^{-1})\, P$, in each market. Hence:

$$P_1(t) = (1-e_1^{-1})^{-1}\, MC(t)$$

$$P_2(t) = (1-e_2^{-1})^{-1}\, MC(t)$$

Now consider price indices $P_i(t)/P_i(b)$ with base year b. Assuming the e_i are constant, the price index in market i is:

$$R_i(t) = P_i(t)/P_i(b) = \frac{(1-e_i^{-1})^{-1}\, MC(t)}{(1-e_i^{-1})^{-1}\, MC(b)} = \frac{MC(t)}{MC(b)}$$

The Export Price Parameters (BPX)

BPX is (NX, NBPX)

parameter \ option	LPH and LLPH	LUM and LLUM	LUK and LLUK	PQHH	PC
1	w.c.	w.c.	w.c.	–	–
2	$\beta(PC)$	$\beta(PC)$	$\beta(PC)$	–	–
3	$\beta(PC_{-1})$	$\beta(PC_{-1})$	$\beta(PC_{-1})$	–	–
4	$\beta(PW)$	$\beta(PW)$	$\beta(PW)$	–	–
5	$\beta(PW_{-1})$	$\beta(PW_{-1})$	$\beta(PW_{-1})$	–	–
6	$\beta(EX)$	$\beta(EX)$	$\beta(EX)$	–	–
7	$\beta(EX_{-1})$	$\beta(EX_{-1})$	$\beta(EX_{-1})$	–	–
8	$\beta(\widetilde{PQHH})$	$\beta(\widetilde{ULC})$	$\beta(\widetilde{ULC})$	–	–
9	$\beta(\widetilde{PQHH}_{-1})$	$\beta(\widetilde{ULC}_{-1})$	$\beta(\widetilde{ULC}_{-1})$	–	–
10	–	$\beta(\widetilde{ZMAT})$	$\beta(\widetilde{ZKEY})$	–	–
11	–	$\beta(\widetilde{ZMAT}_{-1})$	$\beta(\widetilde{ZKEY}_{-1})$	–	–
12	$\beta(PX_{-1})$	$\beta(PX_{-1})$	$\beta(PX_{-1})$	–	–
13	$\beta(\widetilde{Y/YEXP})$	$\beta(\widetilde{Y/YEXP})$	$\beta(\widetilde{Y/YEXP})$	–	–
14	$\beta(\widetilde{AD1})$	$\beta(\widetilde{AD1})$	$\beta(\widetilde{AD1})$	–	–
15	$\beta(t)$	$\beta(t)$	$\beta(t)$	–	–
16	β(r.c.)	β(r.c.)	β(r.c.)	–	–
value of					
JSWPX	1 and 4	2 and 5	3 and 6	7	8
NBPX	16	16	16	∅	∅

That is, the price index is the same for each market.

8. PC: Pure price-taking. The export price equals the competitors' price. If this option is used, and if the data series PW are not defined, the competitiveness effect in the export functions (XA) will not be operative.

EARNINGS (Function PE)

These functions determine the average renumeration of each of NE types of labour. Four options are available.

1. EXGI: The rate of wage inflation is set exogenously:

$$\log PE = \log PE_{-1} + \beta_1$$

β_1 is the logarithmic rate of change of the average wages of the type of labour concerned.

2. EXGW: The wage rate is exogenous.

$$PE = \beta_1$$

3. PHIL: Earnings are determined by a Philips curve relationship. this has a chequered history in econometric forecasting; it has frequently proved to be a very poor forecasting tool despite excellent fit over the sample period. Its use in simulation, however, does permit the response of wages to excess supply and demand in the labour market.

$$d \log PE = \beta_1 + \beta_2 t + \beta_5 \left(\frac{100*UE}{UE + E} + \beta_4\right)$$

$$+ \beta_6 (\log PCE_{-1} - \log PCE_{-2}) + \rho^{NTD} U_o$$

$$d \log PE \geq \beta_3$$

The employment and unemployment variables are both specific to the types of labour under consideration.

The price-term is essentially a catch-up variable. To the extent that last year's prices increased (implicitly unanticipated), labour demands compensation this year.[1/]

The coefficient β_4 may be used to allow for breaks in the wage/unemployment relationship, by reading in a new parameter matrix with a different β_4 when breaks are desired.

The parameter β_3 imposes a lower limit on the rate of change of wages.

The Philips curve has been extensively discussed in the literature: Henry, Sawyer and Smith (1975) discuss the practicalities of estimation and use, while Hansen (1970), among others, considers its theoretical standing.

4. SARG: The "Sargan" equation. This combines Philips-curve types of effect with unions' efforts to maintain an expected real wage. The most accessible explanation is probably Henry, Sawyer and Smith.

$$d \log PE = \beta_1 + \beta_2 t + \beta_5 \left(\frac{100*UE}{UE + E}\right) + \beta_4) + \beta_6 \log (RET/RET_{-1})$$
$$+ \beta_7 \log (PCE_{-1}/PCE_{-2}) + \beta_8 ZPROD_{-1} + \beta_9 \log$$
$$(PE*RET/PCE)_{-1} + \rho^{NTD} U_o$$
$$d \log PE \geq \beta_3$$

RET is the retention ratio. It is defined as (1 - proportionate rate of tax on labour earnings) and is exogenous to IDIOM. A rise in RET seems likely to boost wage claims:

ZPROD is industry-wide productivity

$$ZPROD = \sum_{i=1}^{NY} Y_i \Big/ \sum_{i=1}^{NY} YE_i$$

1/ These lags prevent an instability that might arise if wages were a function of current prices.

It seems likely to boost wage claims by reducing cost pressures on employers.

The term $\beta_9 \log$ (PE * RET/PCE)$_{-1}$ represents a catch-up term in real take-home pay. Its coefficient seems likely to be negative.

In most applications of IDIOM, it seems likely that the exogenous treatment of PE will be most useful, given the appalling history of forecasting wage movement.

Given the disaggregated wages from these functions, IDIOM then calculates the average wages of all employees (AW), all employees in industry(AWY), all in government (AWG) and all employed directly by consumers (AWC). This is done by weighting the values in PE by their shares in the total employment in each of these categories.

INDIRECT TAXES (Function T)

These are pseudo-functions - like functions R. They control not the functional specification of tax collection, but rather the identity of particular entries in the tax matrix. The user may use all or none of the available options, but the classification T must correspond to the functions T to the extent that each item of the classification must be assigned a method of tax formation from the eight options. It is possible, however, to have two (or more) items in the classification determined by the same tax method.[1/]

As with the R-functions, the order of the T-classification is the same for all activities that give rise to taxes. The calculations are carried out in the price sub-routines of IDIOM and the resulting tax takes stored by type of tax and paying entity (industry, consumption good etc.). As before, we illustrate the use of the T-functions by looking at industrial taxes.

1/ Although why this should be desirable is not immediately obvious to me.

The Average Wage Parameters (BPE)

BPE has dimension (NE, NBPE)

parameter \ option	EXGI	EXGW	PHIL	SARG
1	$\overline{d \log PE}$	$\overline{PE}$	β(r.c.)	β(r.c.)
2	–	–	β(t)	β(t)
3	–	–	min. change	min. change
4	–	–	unemploy. const.	unemploy. const.
5	–	–	β(UE)	β(UE)
6	–	–	$\beta(PCE_{-1}/PCE_{-2})$	$\beta(RET/RET_{-1})$
7	–	–	–	$\beta(PCE_{-1}/PCE_{-2})$
8	–	–	ρ	β(ZPROD)
9	–	–	U_o	$\beta(PE * RET/PCE_{-1})$
10	–	–	–	ρ
11	–	–	–	U_o
NBPE	w.c.	w.c.	w.c.	w.c.
value of				
JSWPE	1	2	3	4
NBPE	1	1	10	12

Industrial Taxes

There are NT taxes levied on NY industries. The sub-totals are stored by tax-type in TYØ (NT) and paying industry in YTØ (NY). Parameters (tax-rates mostly) are taken from TYB, which is (NT, NY),[1] and which is treated by IDIOM as a converter.

1. ADVA: Ad valorem taxes on current values.

$$ZT_j = \beta_j * PY_j * Y_j$$

where ZT_j is the payment of these taxes by industry j
β_j the j'th element of the corresponding row of TYB
and PY and Y are standard IDIOM names.

2. SPEC: Specific taxes, levied per unit of output:

$$ZT_j = \beta_j * Y_j$$

3. SUBS: Fixed in nominal terms. More usually subsidies than taxes, in which case $\beta_j < 0$.

$$ZT_j = \beta_j$$

4. EMPW: Employment taxes, proportionate to the wage bill:

$$ZT_j = \beta_j * ZW_j$$

where ZW_j is industry j's wage bill as calculated by the WAGE option of functions R.

5. EMPE: Employment tax per employee:

$$ZT_j = \beta_j * YEØ_j$$

1/ Although it is probably best read in and stored sparse.

6. PRPP: Tax as a proportion of total industrial output at current prices:

$$ZT_j = \beta_j * SPY$$

This is useful if there is a shortage of information concerning the tax system. It is not very realistic, however.

7. PRP: Tax as a proportion of total industrial output:

$$ZT_j = \beta_j * SY$$

8. ADVI: Ad valorem taxes levied on the value of material inputs:

$$ZT_j = \beta_j * ZMP_j * Y_j$$

where ZMP_j is current cost per unit output of material inputs.

$$\underline{ZMP} = [QYC]' * \underline{PQH}$$

Once calculated, the indirect taxes are added into the vectors of payments by industry in the appropriate place; that is RYØ (3). Indirect taxes are a component of value-added.

Other Indirect Taxes

Indirect taxes may be levied on virtually any transaction in IDIOM.

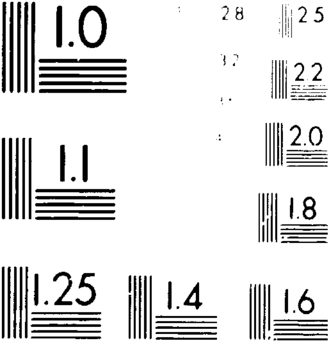
1. Consumption: coefficients TCB, amounts in TCØ and CTØ. All options available but ADVI is levied on the price of the consumption category, regardless of quantities(!)

$$ZT_j = \beta_j * ZMP_j$$

where $\underline{ZMP} = [QCC]' * \underline{PQH}$

B-389

84.04.16



MICROCOPY RESOLUTION TEST CHART

2. Government: coefficients TGB, amounts in TCØ and GTØ. Exactly analogous to consumption.

3. Imports: coefficients TQMB, amounts in TQMØ and QMTØ. Options EMPW, EMPE and ADVI not available; otherwise as for industry.

4. Exports: coefficients TQXB, amount in TQXØ and QXTØ. Exactly analogous to imports.

   Note: taxes are levied on exports on the commodity (Q-) classification rather than the export (X-) classification, and then coverted back to the export classification for use in the export function.

5. Investment: coefficients in TVB and amounts in TVØ and VTØ. Options EMPW and EMPE not available. ADVI is calculated as for consumption, except that industry-specific converters are used for plant (as elsewhere in IDIOM).

Inflation Neutrality

It is possible to inflate specific and nominal taxes by some proportion of the rate of inflation, from year to year. This allows the user to keep the tax system "inflation neutral" if he wishes.

During any year, the entries in the tax matrices corresponding to SPEC, SUBS, EMPE and PRP will be updated to:

$$ZB_t = ZB_{t-1} * (1 + TXPI * (PCE/PCE_{-1} -1)$$

where ZB_t are the tax entries in year t

and TXPI is the exogenous rate of "inflation neutrality" for indirect taxes.

OVERALL INVESTMENT (Function V)

These functions determine non-industrial investment except for social capital formation. The routines concerned also carry out certain house-keeping operations concerning total investment. Four functional options are available:

1. INDL: Industrial investment - determined by functions YV with no further additions at this stage.

2. PLAN: Industrial investment in plant; as above.

3. DWEL: Investment in dwellings. Assuming an institutional framework similar to the U.K.'s, where mortgage payments attract tax relief, and dwelling investment is carried out by the government and personal sectors, this explains personal investment in dwellings by means of income, the "effective" rate of interest, a relative price term, and time. Hence:

$$\log ZVD = \beta_1 + \beta_2 \, RPDI + \beta_3 \, RMORG * (1-SRT) + \beta_4 \, ZPVD/PCE + \beta_5 t + \rho^{YEAR} U_0$$

where ZVD is investment in dwellings, and
ZPVD the price of dwellings (an element of PV).

The semi-log formulation allows for a strong luxary effect. Care must be exercised over the units of RMORG and SRT: SRT is a proportion, $0 \leq SRT \leq 1$; RMORG may be either a percentage or a proportion as long as it is used consistently. (My strong advice is always to work with proportions.)

4. LFEE: Investment in legal fees. Legal fees are a concommitant part of investment, and are assumed to be related to the number and size of transactions taking place. Unfortunately most countrys' data do not allow the identification of gross transfers of property (between and within sectors) and so legal fees are related to gross investment (transfers of new goods). Hence:

$$\log ZVL = \beta_1 + \beta_2 \log V(\beta_3) + \beta_4 \log V(\beta_5) + \beta_6 t + \rho^{YEAR} U_o$$

where ZVL is investment in legal fees.

$V(\beta_3)$ is total investment in asset β_3, etc.

Note:

1. It is perfectly possible to define non-industrial investment assets other than dwelling and legal fees. However, without reprogramming they would need to be explained by one of the above types of functions.

2. For historical reasons, function V does not have an explicit FIX or EXOGENOUS option. However, this effect may be achieved by using either DWEL or LFEE, setting all parameters except β_1 equal to zero and $\beta_1 = \log \overline{ZVD}$.

3. It is perfectly acceptable for industry to invest in dwellings and/or legal fees. This should be done through the YV-functions (remember YV has dimension NV, and thus covers all assets). IDIOM adds industrial and non-industrial investment by asset. Then, except for plant, the resulting totals are converted into commodity demands by the converter QVC. Hence, all buildings - factories, roads and houses - are assumed to require the same commodity inputs.

Social Capital Formation

This is exogenous in IDIOM. The user defines this investment (gross) for each year on the social capital (K-) calssification. IDIOM then converts this to assets, using the converter VKC, adds the result to the other investment by assets, and finally converts to QV by means of QVC.

Investment Parameters VB

VB has dimension (NV,NVB)

parameter \ option	INDL	PLAN	DWEL	LFEE
1	-	-	β(r.c.)	β(r.c.)
2	-	-	β(RPDI)	$\beta(V(\beta_3))$
3	-	-	β(int. rate)	asset for β_2
4	-	-	β(ZPV/PCE)	$\beta(V(\beta_5))$
5	-	-	-	asset for β_4
NVB -3	-	-	$\beta(t)$	$\beta(t)$
NVB -2	-	-	ρ	ρ
NVB -1	-	-	U_o (1)	U_o (1)
NVB	-	-	w.c.	w.c.
value of:				
JSWU	1	2	3	4
NVB	0	0	8	9

(1) Updated once per year - see note (3) to table on BYV.

UNIT LABOUR COSTS (Function YULC)

Unit labour costs are determined for each of NY industries. Total labour costs (ZLC) comprise:

wages and salaries
+ employment taxes levied on number employed
+ employment taxes levied on the wage bill.

All these elements are calculated elsewhere in IDIOM - in the receipts and indirect tax sections - and passed to IDMSLC.

For converting them to unit labour costs two options exist.

1. ACTU: Actual costs. Divide total costs by output:

$$YULC = ZLC/_{Y}$$

2. STAN: Standard, or normal, unit labour costs. Actual total costs are normalized not by actual output, but by expected output - to try to remove the cyclical element from them (see Coutts, Godley and Nordhaus (1978) for an explanation):

$$YULC = ZLC/_{YEXP}$$

The resulting vector, YULC, is used throughout IDIOM wherever unit labour costs are required. It is hence not possible to use standard costs in one application and actual costs in another.

Note: These unit costs are measured in £ per constant price £ of output; not as an index number.

EXPECTED OUTPUT

The standard unit labour costs option depends crucially on the formation of expected output by industry - YEXP. At present there is no functional control over this and YEXP is formed by assuming that the previous four years' outputs lie on a cubic trend, and extrapolating this. The resulting equation is:

$$YEXP = - Y_{-4} + 4\ Y_{-3} - 6\ Y_{-2} + 4\ Y_{-1}$$

However, to help convergence in extreme situations we constrain YEXP by

$$.8\ Y_{-1} \leq YEXP \leq 1.2\ Y_{-1}$$

This is carried out for each industry, and used wherever in IDIOM YEXP occurs.

COMMODITIES and INDUSTRIES (Function Q)

Commodities and industries are distinguished by IDIOM, although obviously they must be closely linked. Function Q informs IDIOM of the nature of that link.

It has already been pointed out that the number of industries must not exceed the number of commodities (NY ≤ NQ). Function Q is defined for each commodity as:

1. <u>IDEN</u>: If this commodity is the principal product of the corresponding industry (the corresponding industry is the "principal producer" of the commodity). "Corresponding" means in these circumstances that the industry and commodity have the same position in their respective vectors. That is industry 3 corresponds to commodity 3. Clearly if the user wishes to suppress the difference between industries and commodities he sets Q to IDEN.

2. DIFF: If the commodity is the product of a different industry. Each commodity must have a "princiapl producer" - an industry whose behaviour represents this commodity - and if Q is set to DIFF, this industry is defined by the entry in vector BQ[1/] referring to the commodity in question.

UNEMPLOYMENT (Function UE)

This is predicted separately for each of the NE types of labour. "Real" unemployment is the difference between the exogenously given labour force (LF) and total employment (E) for each type. There are no labour supply considerations within IDIOM. For fiscal purposes, however, and also for comparing the models' output with published sources, it is not "real" employment, but "registered" unemployment that counts. The relationship between the two will vary according to the costs and benefits of registration, but in the U.K. there is substantial evidence that as "real" unemployment rises, the registration rate (the registered/real ratio) falls. These functions allow the user the choice of whether to use "real" unemployment or "registered" unemployment in the various places unemployment figures in IDIOM.

1. SIMP: "real" unemployment:

$$UE = LF - E$$

2. VARP: Variable registration, with registration rate ZUE given by:

$$ZUE = \beta_1 + \beta_2 t + \beta_3 / \log(\beta_4 ZU)$$

where $ZU = (LF - E)/LF$ is the unemployment rate, and

$$UE = ZUE * (LF - E)$$

Note: It is also possible to justify the variable registration equation in terms of variable participation, if you hold that those people who do not register actually leave the labour force, rather than just become "invisible".

1/ Sometimes expanded to BQC

Employment

IDIOM forms employment by labour type (E) by passing total employment vectors through converters. There are three vectors of total employment:

(i) YE (or YEØ) industrial employment, which has been discussed under functions YEØ. Industrial employment by type (EYØ) is formed as:

$$\underline{EY\emptyset} = [EYEC] * \underline{YE\emptyset}$$

(ii) Employment by consumers directly (e.g. personal servants, charity workers, and possibly subsistence agriculture). First employment by consumer good is formed:

$$CE\emptyset_i = CE\emptyset B_i * C_i \qquad \text{for good } i$$

where CEØB is a vector of multipliers (coefficients) generating direct employment from the volume of consumption. CEØB will normally comprise mostly zeros. Then CEØ is converted to ECØ by:

$$\underline{EC\emptyset} = [ECEC] * \underline{CE\emptyset}$$

(iii) Employment by government. First by government current expenditure category:

$$GE\emptyset_i = GE\emptyset B_i * G_i$$

and then by labour type:

$$\underline{EG\emptyset} = [EGEC] * \underline{GE\emptyset}$$

Finally total employment by type:

$$\underline{E} = \underline{EY\emptyset} + \underline{EC\emptyset} + \underline{EG\emptyset}$$

and $$EMPC = \sum_i EC\emptyset_i$$

$$EMPG = \sum_i EG\emptyset_i$$

THE UNEMPLOYMENT PARAMETERS (BUE)

BUE has dimensioı (NE, NBUE)

parameter	VARP
1	β(r.c.)
2	β(t)
3	β(1/log ZU)
4	β(ZU)
5	-
6	-
NBUE	w.c.

JSWUE	2
NBUE	7

THE INPUT-OUTPUT TABLE (Function A)

IDIOM allows the user four ways of updating the mix matrix during a run (as well as the options of (a) leaving it unchanged, and/or (b) reading in a completely new version periodically). The input-output table, hence the updating, is denominated in constant prices, and refers only to material inputs (value-added being handled elsewhere).

1. NULL: Do nothing to the A-matrix. (default value)

2. CDPF: The input-output coefficients are constant in current prices. This implies that the bundle of material inputs (which is to be combined with the value-added components) is built up from individual materials in a Cobb-Douglas manner.

$$ZA_{ij} = \beta_{ij1} * PY_j/PQH_i$$

where ZA_{ij} is element i,j of the A-matrix (inputs of i into j)

β_{ij1} is element i,j of the A-matrix in the base-year of the price index numbers PY and PQH. This is stored in the first column of BQYC.

This function is executed during the COMPUTE phase, so the input-output matrix, QYC, should be initialized to its value in the year preceding the first projection year in order that any lagged material costs might be correctly built up.

3. <u>CES</u>: The materials bundle is built up as a CES function. The resulting A-coefficients are:

$$ZA_{ij} = \beta_{ij1} \, (PY_j/PQH_i)^{\beta_{ij2}}$$

4. <u>LTIM</u>: The A-matrix elements evolve through time along a logistic curve. These curves involve non-linear estimation, but they appear to be reasonable representations of the evolution of less developed countries' technology.

$$ZA = \beta_1 + \frac{\beta_2 - \beta_1}{1 + \beta_3 e^{\beta_4 * NYEARS}}$$

This functional form offers considerable flexibility, by virtue of its plentiful parameters. In particular, it allows the user to set the upper and lower limits of ZA. For instance, assuming $\beta_4 < 0$ (as is usual), and NYEARS $\geq$ 0, ZA ranges from $\beta_1 + (\beta_2 - \beta_1)/(1 + \beta_3)$ to β_2.

5. <u>LGDP</u>: The A-matrix is a logistic function of GDP, lagged. The reason for lagging GDP is computational effeciency - it allows these calculations to be done during the UPDATE phase.

$$ZA = \beta_1 + \frac{\beta_2 - \beta_1}{1 + \beta_3 e^{\beta_4 (GDP_{-1}/GDPT)}}$$

GDPT is a macro variable, exogenously set by the user, which merely acts as a scale factor in this equation. (It should be the same as is used in the estimation of this function).

THE INPUT-OUTPUT PARAMETERS (BQYC)

BQYC has dimension (NQ, NY, NBQYC)

parameter \ option	NULL	CDPF	CES	LTIM	LGDP
1	-	$\beta(PY/PQH)$	$\beta(PY/PQH)^{\beta_2}$	const.	const.
2	-		power of prices	limit	limit
3	-		-	scale	scale
4	-		-	$\beta(t)$	$\beta(GDP/GDPT)$
value of					
JSWQYC	1	2	3	4	5
NBQYC	0	1	2	4	4

NOTES:

1. Options LTIM and LDGP are executed in the UPDATE phase. In order that lagged values using the A-matrix are correctly calculated elsewhere in that phase, function A should be executed last. This is its default position, and it should not be moved.

2. It is not possible to update zeros in the A-matrix. Attempts to do so will not be faulted, however.

3. The column summs of the A-matrix are not checked. The user must maintain his own checks.

4. Each element of the A-matrix may be updated differently. Each has its own parameters and functional switch.

INSTITUTIONAL SECTORS (Function H)

These are yet another set of "pseudo-functions". The institutional (H-) classification defines the number of sectors and the H-functions merely allocate each one a type. Hence:

```
FUNCTION H HOUS 1 ROW 2 CGOV 4 CORP 3;
```

tells IDIOM that of the four institutional sectors (NH=4) the first is households, the second the rest of the world, the third corporations and the fourth central government. Knowing this, IDIOM is then able to calculate the payments and receipts of these sectors using functions HRP and a lot of accounting identities, and to write the results into the correct element of the payments matrices.

IDIOM uses the switch JSWH to record where each sector is stored. The Ith element of JSWH records the definition of the Ith of the NH institutional sectors, where values of JSWH are:

households	JSWH = 1
non-profit making bodies	JSWH = 2
life assurance funds	JSWH = 3
corporations	JSWH = 4
nationalized industries	JSWH = 5

central government	JSWH = 6
local government	JSWH = 7
rest of the world	JSWH = 8

Hence in the example above JSWH would have dimension 4 and would be filled as follows: (1, 8, 4, 6)

PAYMENTS AND RECEIPTS (Function HRP)

This set of functions builds up the flow of funds matrices necessary to close IDIOM. Payments and receipts are disaggregated by type of payment (receipt) - the R-classification and functions - and the institutional sector involved - the H-classification and functions. The payments are constructed in various ways using either accounting identities or behavioural equations, the parameters of the latter being held in matrix BHR. The results of the calculations are written to two (NH, NR) matrices HRR and HRP, which record respectively receipts and payments disaggregated by institution and type of payment.

It is important to distinguish the receipts and payments under consideration here (HRP) and those calculated under the R functions. With the exception of the goods column, the R-functions are basically value-added flows disaggregated by the originating activity and the nature of the value-added. They are part of the production side of the "real" economy. The present flows, while largely built up from those flows, concern transfers and the distribution of income and expenditure flows. They have little significance for the "real" economy, except in the determination of personal disposable income. Note that there is no separate classification corresponding to the HRP functions: receipts and payments are built up on the R-classification, the HRP functions merely determining how.

To ease the exposition of this sub-section, we note now the dimensions and classifications of various matrices.

HRR (NH, NR) receipts by sector I of type J, I=1,..NH; J=1..NR

HRP (NH, NR) payments by sector I of type J

HRB (NH, NR) a converter (parameter matrix) translating total payments by type (R = RCØ + RGØ + RYØ + RVØ) into receipts by sector and by type. Hence:

$$[ZHRR] = [HRB] * [\hat{R}]$$

where ZHRR is that part of HRR which stems from R. This, in fact, is the majority of HRR - see below, where only the items additional to this calculation are mentioned.

BHR (NH, NR, NBHR) the parameter matrix for payments and receipts equations. The column BHR (I,J) usually refers to <u>payments</u> of type J by sector I. the receiving sector is either obvious, or recorded as BHR (I,J,3). Occasionally, if payment (I,J) is known to be zero, its column of BHR is used to store parameters for flow (K/L). If so BHR (I,J,4) is used to define K (see below). Both these sector definitions should refer to the user's ordering of sectors. Hence, continuing the example used above to illustrate the H-functions, a receipt by central government would be indicated by setting BHR (I,J,3) = 4.

As already noted, the order of institutional sectors and of payment types is not fixed - it is defined by pseudo-functions H and R respectively. This makes the programming of routines IDMSHR and IDMUHR very messy (and does not help our exposition that follows!). During initialization, IDIOM defines the switch vectors JSWH and JSWR from the key-word inputs so that, for example, the value in JSWH (I) records the definition of the Ith of the NH institutional sectors, where the values of JSWH have been defined under functions H above (and those of JSWR under functions R). Frequently in IDMSHR it is necessary to find which row of HRR or HRP corresponds to corporations say, i.e. whether corporations are the first, second, third etc., institutional sector. To do this IDIOM uses a function:

INDLOC (4, JSWH, NH)

which searches JSWH (which has dimension NH) to find which element contains the value 4, the indicator for corporations. When this is found, and written say to I1, operations on HRR (I1,.) build up receipts by corporations.

Below we shall use the notation "CORP" to refer to the index for corporations, "ROW" for the index of the rest of the world etc.; that is, we shall use the key word from the H-functions enclosed in quotes to denote the index of the required sectors. We shall similarly use the key words from the R-functions to denote indexes of various payments; e.g. "WAGE" denotes the index for wages in the R-classification.

We shall also, for this subsection, use the Fortran subscripting notation - enclosing subscripts in brackets. Hence:

HRR ("CORP", "GOOD") denotes the receipts of corporations from the sale of goods and services.

Note again that once the ordering of the H and R classifications has been defined by the respective pseudo-functions, all vectors and matrices using those classifications must stick to that order.

The calculation of payments and receipts is done by a mixture of accounting and functional operations. The following exposition does not separate these, but rather takes the user through the calculations in the order in which they occur. The different nature of the various calculations is, however, obvious.

IDIOM starts its receipt/payment calculations by deriving the basic receipts matrix from the vector of payments to value-added.

The Basic Receipts Exercise

The payments to value-added, including all indirect taxes, i.e. including taxes on exports and imports, are summed to a single vector of total payments by type. This is then "converted" to form the basic receipts matrix, using HRB which gives each sector's share (receipt) of each type of payment.

$\underline{R} = \underline{RC\emptyset} + \underline{RG\emptyset} + \underline{RY\emptyset} + \underline{RV\emptyset}$

R("ITAX") = R("ITAX") + all taxes on trade

$[ZHRR] = [HRB] * [\hat{R}]$

Note: the converter operates on total payments (or receipts); no distinction is made according to the origin of the payment either by activity (consumption, investment etc.) or by industry or good.

Other Receipts

HRR ("ROW", "GOOD") = SPQM

i.e., receipts by the rest of the world for goods and services are equal to total current price imports.

Payments

HRP ("HOUS", "WAGE") = WAGC
HRP ("CGOV", "WAGE") = WAGG
HRP ("HOUS", "IMP") = SPCM
HRP ("CGOV", "IMP") = SPGM
HRP ("HOUS", "GOOD") = SPC - SPCM - WAGC
HRP ("CGOV", "GOOD") = SPG - SPGM - WAGG

i.e., households' expenditure comprises direct employment costs (WAGE), direct imports (SPCM) and goods and services. Similarly for government.

HRP ("ROW", "GOOD") = SPX

i.e., the rest of the world's payment for goods and services in our exports.

Functional Relationships

The HRP functions permit seven methods of calculating different payments and/or receipts. Not all of these are suitable for all flows, of course, but the user will easily be able to see which functions to use for any flow. We define the functions first, and then examine certain features of their use in IDIOM.

1. DIST: profit distribution

$$\log ZA = \beta_1 + \beta_2 t + \beta_4 \log (HRR\ (I,J) - ZCT) + \rho^{NTD} U_o$$

$$HRR\ (\beta_3,\ J) = HRR\ (\beta_3,\ J) + ZA$$
$$HRP\ (I,J) = HRP\ (I,J) + ZA$$

I and J count directly over 1...NH, and 1...NR respectively

ZA is the amount distributed and it is related exponentially to the sectors' receipts of type J less corporation tax (ZCT)

ZA is then credited to sector β_3 as a receipt and to sector I as a payment. If (HRR (I,J) - ZCT) < 1 it is set to 1.

In concrete terms this function is only suitable for companies' distributions of profit. From the accounting step above, HRR ("CORP", "PROF") contains all profits accumulating to corporations. Corporation tax is paid on this (see below) and of the remainder ZA paid to sector β_3 as dividends.

β_3 is the index, in the user's ordering of H, of the sector to which payment is made.

2. INTE: interest payments

$$\log ZA = \beta_1 + \beta_2 t + \beta_5 \log \beta_6 + \beta_7 \log RINT + \rho^{NTD} U_o$$

$$HRR\ (\beta_3,\ \text{"PROF"}) = HRR\ (\beta_3,\ \text{"PROF"}) + ZA$$
$$HRP\ (\beta_4,\ \text{"PROF"}) = HRP\ (\beta_4,\ \text{"PROF"}) + ZA$$

The interest payment is a simple function of the rate of interest and the cumulated debt (see below), which is stored in β_6.

The interesting feature of this option is that although it is stored in column (I,J,.) of BHR it refers to receipts and payments by other sectors. The type of payment is always profits and the receiving and paying sectors are β_3 and β_4 respectively. The necessary parameters can be stored in any column of BHR that is not required for anything else, so long, of course, that that column is given the key-word INTE during initialization. For example suppose local government is the fifth institutional sector, central government the fourth and corporations the third. Suppose also that imports are the second type of payment in the R-classification. Provided that the payment HRP ("LGOV", "IMP") is not predicted using the functional options, column (5,2,.) of BHR is free. We could then use it for corporations' interest payments to central government by setting BHR (5,2,3) = 4, BHR(5,2,4) = 3 and declaring in the initialization phase:

FUNCTION HRP INTE 5:2 ......

3. GRAN: grants

$$\log ZA = \log PUP + \log \beta_1 + \beta_2 t + \rho^{NTD} U_o$$

The current grant is the product of:

(a) the GDP (factor cost) deflator, presently set equal to the GDP (market prices) deflator for the purposes of this subroutine

(b) a base year value of the grant at constant prices

(c) a modifying term $\beta_2 t + \rho^{NTD} U_o$.

$$HRR (\beta_3, J) = HRR (\beta_3, J) + ZA$$

$$HRP (I,J) = HRP (I,J) + ZA$$

4. FIX: fixed payment

$$\log ZA = \log \beta_1 + \beta_2 t + \rho^{NTD} U_o$$

As above except without the GDP deflator.

The Receipt/Payment Parameters (BHR)

BHR has dimension (NH, HR, NBHR)

parameter \ option	DIST	INTE	GRAN	FIX	SPEC
1	β(r.c.)	β(r.c.)	base value	base value	β(r.c.)
2	β(t)	β(t)	β(t)	β(t)	β(t)
3	index of receiving sector				β(WVOL)
4	β(HRR_{ij}-ZCT)	paying sec.	ρ	ρ	β(WPRI)
5	ρ	β($β_6$)	U_o	U_o	β(EX)
6	U_o	cum. debt	-	-	ρ
7	-	β(RINT)	-	-	U_o
8	-	bond-share	-	-	-
9	-	-	-	-	-
10	-	ρ	-	-	-
11	-	U_o	-	-	-
NBHR	w.c.	w.c.	w.c.	w.c.	w.c.
value of					
JSWHR	1	2	3	4	5
NBHR	7	12	6	6	8

5. <u>SPEC</u>: special function for profit and interest from abroad.

$$\log ZA = \beta_1 + \beta_2 t + \beta_3 \log WVOL + \beta_4 \log WPRI + \beta_5 \log EX$$

$$+ \rho^{NTD} U_o$$

This function is a simple function of foreign variables. If it is to be used it must be stored in column ("ROW", "PROF") of BHR and the key-word SPEC must be specified for flow "ROW" : "PROF".

HRR ("CORP", "PROF") = HRR ("CORP", "PROF") + ZA
HRP ("ROW", "PROF") = HRP ("ROW", "PROF") + ZA

The results are necessarily credited to corporations as receipts. Profits from abroad may alternatively be calculated by the GRAN and FIX options, but the accounting is still as above.

6. <u>NULL</u>: do nothing. The values calculated by the accounting procedures stand.

7. <u>PFA</u>: profit from abroad - currently undefined.

The functions can potentially be used for any flows but their most obvious use is for the distribution of profits. We, therefore, illustrate their use in this particular case.

The total of profits as a value-added payment is stored in R("PROF") and is distributed to sectors as receipts by the converter HRB; see above. This is pre-tax and concerns the sector earning profits, not the sector ultimately benefitting from them. The following stages are then undertaken:

(a) profits from abroad: calculated by GRAN, FIX or SPEC options, credited to corporations;

(b) distribution of equity dividends: payment by companies, determined by DIST option using parameters in column ("CORP", "PROF") of BHR. Only one receiving sector permitted - normally the household sector;

(c) interest payments - determined by INTE option: The column ("CORP", "PROF") of BHR has already been used by the equity equation, so these functions need to be fitted into BHR elsewhere. The key-words for columns used must be set to INTE and the paying sector, defined in β_4, as "CORP".

(d) corporation tax calculated using column ("CORP", "ITAX") of BHR as follows:

$$ZCT = \beta_1 * ZUD + \beta_2 * ZD$$

where:

ZCT is corporation tax payments

ZUD undistributed profits = HRR ("CORP", "PROF") - HRP ("CORP", "PROF"), where HRR and HRP contain the results only of the calculations up to this point.

ZD distributed profits = equity payment calculated above.

NOTE: 1. Interest payments are assumed tax deductable (since they are included in HRP ("CORP", "PROF")).

2. The row "ITAX" of the payments classification has now been widened to include direct taxes.

3. BHR ("CORP", "PROF") must contain these parameters and the key-word NULL must be specified for it to prevent any other interpretation of the column.

Tax is also due on profits paid abroad as follows:

$$ZPA = \beta_3 * HRR\ (\text{"ROW"}, \text{"PROF"})$$

$$HRR\ (\text{"CGOV"}, \text{"ITAX"}) = HRR\ (\text{"CGOV"}, \text{"ITAX"}) + ZPA$$

$$HRP\ (\text{"ROW"}, \text{"ITAX"}) = HRP\ (\text{"ROW"}, \text{"ITAX"}) + ZPA$$

HRR ("ROW", "PROF") has already been calculated in the conversion exercise from R.

Cumulated Debt

The interest payment functions require data on cumulated debt. IDIOM cumulates debt as an accounting operation during the UPDATE phase. Interest is paid on debt outstanding at the beginning of the year in question. The updating takes assets at the beginning of the previous year and using the flow data from that year calculates assets at the beginning of the present year.

The change in net indebtedness by sector over a year is given by savings-investment. Savings is available from the receipt and payments subroutine, but investment by sector has to be specially calculated as follows:

(a) Total industrial investment (at current prices) - the sum of stock-building and fixed capital formation - is allocated across sectors using the same shares as are used to split profits among sectors, i.e. using column (.,"PROF") of HRB.

(b) Government investment on the K-classification is converted to assets (using VKC) and its current price sum found (multiplying by PV). This is allocated direct to central government.

(c) Household investment is calculated as the difference between the total value of investment (V*PV) and the sums of items (a) and (b) above. It is allocated to the household sector.

For any sector paying debt interest, the cumulated debt variable, β_6, is updated as:

$$\beta_6 (t) = \beta_6 (t-1) + \beta_8 \text{ ZDD } (t-1)$$

where ZDD (t-1) is the sector's change in net indebtedness over year (t-1).

β_8 is the proportion of the sector's change in indebtedness that bears interest. Normally this will be unity (given that real assets are treated as investment) but for Central Government the possibility of issuing cash may reduce β_8, so that $0 \leq \beta_8 \leq 1$.

The residual error (RESE) referred to in IDIOM is the error in the financial accounts and is calculated as the sum of the sectors' net indebtedness. Ideally it should be zero, but in IDIOM, as in the national accounts, the independent calculation of different flows frustrates this.

The Personal Tax System

This is incorporated in IDIOM as an accounting relationship, hence there is no option but to use it. If data are not available for it, the user should set the various rates to zero.

The system is based on British personal taxation. It treats self-employment income (which is treated in the national accounts as profits) equivalently to wages. It allows different marginal rates of tax on different sources of income, but defines the tax brackets according to total taxable income. Obviously IDIOM cannot tax individuals so the tax system works on per capita income.

IDIOM requires the following data for the personal tax system; it treats the matrices as parameter matrices rather than converters.

There are ND income brackets:

DBT (ND) — records the lower limit of each bracket in terms of taxable income.

DDB (ND, NR) — contains the marginal rates of tax on each kind of income (NR) for tax-payers whose total taxable income puts them in any particular tax bracket (ND).

DAB (ND, NR) — contains the personal allowances per capita for income J, J=1...NR, in tax-bracket I, I=1...ND.

DRB (ND, NR) — gives the proportion of total income of type J, J=1...NR, accuring to individuals in tax bracket I, I=1...ND. This is an IDIOM converter. (It is also know in some parts of the programme as DRC).

and DN (ND) — gives the distribution of the labour force over tax-brackets. It gives the share of WPOP whose total taxable income lies in each bracket, that is the proportion of WPOP whose top marginal rate of tax is that of this bracket.

IDIOM builds up the following variables:

D (ND) total tax paid by people in each tax bracket

and total tax paid by the personal sector and received by the central government.

The system operates on each tax bracket in turn. First it calculates average taxable income by type of income for the bracket under consideration (K):

$$ZT\ (J) = (1000 * HRR\ (\text{"HOUS"},\ J) * DRB\ (K,J)/ZWP) - DAB\ (K,J)$$

where:

ZT(J) is the average per capital taxable income of type J accruing to people in bracket K[1/].

HRR("HOUS",J) is households' total receipts of income of type J.

DRB(K,J) is the share of the total of income J accruing to workers in bracket K.

ZWP is the number of workers in bracket K

DAB(K,J) is per capita allowances against income type J allowed to people in bracket K.

and the factor of 1000 converts the per capita incomes to more convenient units. (10^{-3} times the units used for aggregate flows).

IDIOM also defines average total per capita taxable income :

$$ZTT = \sum_{j} ZT\ (J)$$

1/ People are allocated to bracket K if their taxable income exceeds K's lower limit but falls short of (K+1)'s lower limit.

The tax system operates by levying the highest of the marginal rates in bracket K on income of the relevant kind, until either that income is exhausted or the part of income remaining untaxed has fallen below the lower limit of bracket K. If the former occurs the next highest marginal rate within bracket K is levied on its respective income until one of the stops is met, while if the latter occurs the tax rates of bracket K-1 are applied to any income that has not been taxed in bracket K. This continues until all taxable income has been taxed. The two guiding principles are that no unit of income is taxed more than once, and tax is always levied at the highest available rate.

Example:

Suppose ZTT is high enough to warrant starting with bracket K. Let DDB(K, MAX) be the highest marginal rate in bracket K; it is, obviously enough, levied on income of type MAX. It is levied on the minimum of :

(i) ZT(MAX)
(ii) ZTT - DBT(K)

That is, if when income MAX has been taxed, the remaining untaxed income still lies in bracket K, tax is levied on all of ZT(MAX). If, however, taxing all of ZT(MAX) would leave remaining income below the limit for bracket K, tax is levied on ZT(MAX) until remaining income just falls below DBT(K). i.e. on ZTT - DBT(K).

If step (i) applies, i.e. ZT(MAX) < ZTT - DBT(K), IDIOM then searches for the next highest marginal rate in bracket K and repeats the above exercise, except that the income already taxed, ZT(MAT), is removed from ZTT. That is ZTT should be interpreted as the total of taxable income not yet taxed.

If step (ii) applies, the tax system now has to consider tax rates in bracket (K-1). Tax has been levied on the amount (ZTT - DBT(K)) of income type MAX; hence [ZT(MAX) - (ZTT - DBT (K))] of MAX remains to be taxed, and total taxable income not yet taxed is equal to DBT(K). With these values substituted for ZT(MAX) and ZTT respectively, the tax system applies the above procedure again, but using the rates from bracket (K-1). Operationally the basic rule is that once a unit of income has been taxed, that unit is removed from (i) the relevant total of that type of income, and (ii) total income for defining the tax brackets.

Finally, the per capita taxes for people in this income bracket are grossed up by numbers in the bracket to get total tax take.

This process is repeated on each tax bracket. It is clear that people in tax-bracket K potentially pay taxes at the marginal rates of all brackets from 1 to K; nevertheless, their tax and their presence is credited to tax bracket K in the vectors D and DN. i.e. DN(K) refers to all people who pay some tax at rates DDB(K,.) but not at rates DDB(K+1,.), and D(K) refers to all personal tax paid by such people.

Inflation Neutrality

It is possible to update the tax brackets and allowances to keep pace with inflation, or to do so partially. In any year:

$$DBT_t = DBT_{t-1} * (1 + TXPD * (PCE/PCE_{-1} - 1))$$

where TXPD is the exogenously set index of inflation neutrality for direct taxes, is treated similarly.

The Final Accounting

Once calculated, total personal tax paid (ZD) is added to the accounts:

HRR ("CGOV", "ITAX") = HRR ("CGOV", "ITAX") + ZD
HRP ("HOUS", "ITAX") = HRP ("HOUS", "ITAX") + ZD

Savings by institutional sectors are then calculated as total receipts less total payments. The negative of Central Government savings is also written to PSFD and the negative of Rest of the World savings to BP.

Personal disposable income (by tax bracket and in total) is calculated as income less tax, and real personal disposable income (in aggregate only) found by deflating by PCE. Expected RPDI is also set, for present, to RPDI.

THE ECONOMIC STRUCTURE - ADDITIONAL NOTES

GOVERNMENT IN IDIOM

The government is exogenous in IDIOM, except possibly for certain payments and receipts. It determines tax rates, brackets etc., and also its own expenditure. The latter is fixed in constant prices in two parts:

G = current expenditure
K = social capital formation.

Each has its own classification (the G- and K-classifications) and IDIOM itself determines the prices of these expenditures. The price of G is PG:

$$\underline{PG} = [QGC]' * \underline{PQH} + [\hat{G}]^{-1} * \underline{GR\emptyset}$$

where $[\hat{G}]$ is a square matrix with the elements of vector $\underline{G}$ down the main diagonal, and $[\hat{G}]^{-1}$ its inverse, and $\underline{GR\emptyset}$ is total value-added payments by government.

Hence, the converter QGC is seen to be a sort of input-output table; it reflects the demands for commodity I arising from government expenditure of type J.

The price of social capital formation is determined on an asset basis. First K is converted into investment assets, using VKC, and then the prices by asset (PV) are applied.

TIME IN IDIOM

Several variables in IDIOM refer to time. All are integers:

(a) <u>In functions</u>: Two variables appear regularly in functions - they are both initialized by the user and care should be taken to ensure that the values used in projection are compatible with those implied by the parameter matrices:

YA70 - appears in function sub-routines as NYEARS used as the time trend variable in functions

YA72 - appears in function sub-routines as NTD used as the power of ρ in determining expected error of projection equation.

(b) In control: It is important to distinguish the first year that is projected in any particular run of IDIOM and the first year that could be projected. They may differ if an IDIOM run is restarted from a dump (using PUTGET ALL and GET ALL commands). Under these circumstances all IDIOM store is filled with the values for the year indicated in the GET ALL command, just as if the model had been solved from the start. Hence, so far as IDIOM is concerned, the "real start" of the run is the first year that was solved in order to produce the dump from which the re-start was initialized.

START The first year of the dump. Measured in actual time e.g. 1973.

FINISH The last year of the dump. Set by IDIOM, the user need not set this.

YEAR The year being solved, relative to START. This is the index variable of the loop in the SOLVE phase. If the model is not restarting from a dump YEAR must start at 1. The results of the first year's calculation will be dumped into the first slot of the dump and labelled as referring to year START. If we are restarting, YEAR = 1 corresponds to the year START, and actual solution may be started for any YEAR $\geq$ 1 provided, of course, that the dump contains information up to the year corresponding to (YEAR-1). (In Cambridge MDM we in fact need a dump for the year corresponding to YEAR. It may be the same here.)

UNITS IN IDIOM

The units in which variables are to be measured are stated in JRNS.IDIOM.UNITS, which may be set by the user. IDIOM obviously requires consistency over the units of variables, and expects the following rules to be adherred to.

- aggregates, whether in constant or current prices, must all be in the same units.

- workers and population may be in any units as long as "per capita" or "per worker" is interpreted to mean per unit as declared in IDIOM. Work-force variables must all have the same units and so must population variables, but the two need no be identical.

- "per worker" variables are measured in units 10^{-3} smaller than the aggregates (i.e. IDIOM requires a 1000 x larger number to denote the same quantity).

- "per capita" variables are not similarly adjusted; they are just aggregate unit per population unit.

Prices

All prices are measured as index numbers, and all should have the same base year. In the hand-book, wherever I write prices I mean price indices.

Also the price indices are current weighted; i.e. they are deflators. The basic variables of IDIOM are mostly measured in constant prices (income flows being the main exception) and so in order to preserve the value identity prices have to be measured as current weighted indices.[1/] Hence:

value = (quantity at constant prices) x (current weighted price index)

1/ A constant price aggregate is merely the product of a base weighted quantity index and a base year value.

Currencies

IDIOM expects all variables to be measured in domestic currency, except for:

foreign prices	PCLØ, etc.
	PWLØ, etc.
	PFM
foreign demands	AD1, AD2, DI1, DI2.

(quite possibly these will be index numbers or quantity indicators rather than constant price amounts. The only rule is that what is input to IDIOM should be commensurate with what was used in the estimation stage):

and the exchange rate EX

Foreign prices occur only in company with the exchange rate. The former should be measured in some foreign currency (or bundle of currencies) and the latter give the number of foreign units per domestic unit; i.e. EX uses the British exchange rate convention of quoting, say, $ per £. Since the foreign prices are all index numbers, EX, too, may be quoted as such (again assuming that the index form has been used in estimation).

For example consider the test job described below. It, crudely, applies to the UK:

- aggregates are measured in £ millions
- work-force statistics in "000"s
- and population in millions.

Consider, now, a "per worker" variable. In the initialization phase:

- direct employment by consumers (EMPC) = 100
- the wage bill paid by consumers (WAGC) = 130
- the average wage (wage per worker) (AWC) = 1300

In natural units this means that 100,000 people were employed and they earned £130,000,000 at an average of £1,300 each. To IDIOM it means £130 millions paid to 100 thousand workers or 1.3 aggregate units per worker unit, which is 1300 in "per worker" units.

Consider also a "per capita" variable. Aggregates are measured in millions and so is population. Hence, in IDIOM "per capita" consumption is interpreted as £ millions consumed per million population, but it is plain that this is just £ per consumer.

For convenience I would recommend adopting the test run's conventions except where there are pressing reasons to the contrary. And that if changes are made you try to keep them to multiplying <u>each</u> unit by 10^i where i is any positive or negative integer.

BASE YEAR

IDIOM does not require a base year in the sense that some known year of the economy is updated in order to get projections of a future year. IDIOM treats each year from scratch to a much greater extent than that process implies. Nevertheless, there are "base year" type concepts required.

(a) <u>The initialization year</u>

IDIOM solves year by year, taking as given the solution for the previous year. To start the model off for year T, therefore, it is necessary to fill out the store of IDIOM as if it has just solved for year (T-1). That is all the scalars, vectors and matrices of variables (and to some extent parameters) have to be filled with values for (T-1).[1/] IDIOM then undertakes such UPDATING as is necessary to prepare for the solution of year T.

1/ IDIOM will flag <u>any</u> store that is not initialized. While many values are not strictly necessary to start the solution, this feature should not be over-ridden for it is surprising how often errors creep in from the failure to initialize some crucial variable.

To the extent that the solution of T requires lagged values of data it is essential that the initialization be to correct, or at least plausible, values. Most cases where this is significant are obvious enough from the functional specifications. However, there are some cases where this is not so. Also (T-1)'s solution provides the starting point for the interations in the solution of T, and so the more realistic are the initializations, the quicker IDIOM is likely to solve year T. Hence from both points of view it is desirable that the initialization values are generally plausible and internally consistent.

In this sense IDIOM requires a base year view of the economy. A list of the "necessary" correct initializations follows under the heading initialization.

(b) Accounting

All real variables are measured in constant prices in IDIOM, and naturally all have to use the same year's prices. Thus, there is a "base year" for the price indices. This is know as PBASE.

Certain data are required for this year - specifically the overall rates of tax on each taxable activity - i.e. government, consumers, industry, etc. These are necessary for the definition of tax-inclusive price index numbers from IDIOM's basically tax-exclusive indices, and also for evaluating the tax component of various transactions at constant prices.

INITIALIZATION

The following variables are used lagged by IDIOM. It is therefore necessary that they be correctly initialized.1/

Variable	Options in which it occurs
PQM	QMØ (LLIN), PX (LUK,LLUK)
PQ	QMØ (LLIN), PX (LUK,LLUK)
QMTØ	QMØ (LLIN), PX (LUK,LLUK)
QMØ	QMØ (LLIN), PX (LUK,LLUK)
QMQ	QMØ (QMQ)
Y	YV (NCL,NCLP,ACC), YE (RECU,LLIN,LLWA), YS (FULL), PE (SARG)
YL1	YV (NCL,NCLP)
YL2	YV (NCL,NCLP)
YL3	YV (NCL,NCLP)
PV	YV (NCL), HR (accounting)3/
PY	YV (NCL,NCLP)
PVA	YV (NCL,NCLP), HR (accounting)3/
PYVP	YV (NCL,NCLP)
PYVL1	YV (NCL,NCLP)
PYVL2	YV (NCL,NCLP)
YE	YEØ (RECU,LLIN,LLWA), PE (SARG)
YV	YEØ (RECU), HR (accounting)
YH	YEØ (RECU)
PYS	YS (FULL), HR (accounting)
DC	DC (DUR)
DPOP	DC (DUR), DSC (LNMW,LRLW,LLNW,LLRW)2/
DPDI	DSC (LNMW,LRLW,LLNW,LLRW)2/
DSC	DSC (LNMW,LRLW,LLNW,LLRW)2/
DPCE	DSC (LRLW,LLRW)2/

1/ During UPDATE, nearly all variables hold the values resulting from the year before. Hence, for example, we need to initialize PQM correctly, because PQM_{-1} is used in the import functions.

2/ These are necessary only if the consumption function includes a wealth effect.

3/ If wealth does not enter the consumption functions and interest flows are not calculated under HR, these may be ignored.

Variable	Options in which it occurs
DPWPH	DSC (all)2/
PCLØ	PX (LUK,LLUK,LUM,LLUM,LPH,LLPH)4/, XA (all)
PWLØ	PX (LUK,LLUK,LUM,LLUM,LPH,LLPH)4/, XA (all)
PX	PX (LUK,LLUK,LUM,LLUM,LPH,LLPH)4/, XA (all)
PQHH	PX (LUK,LLUK,LUM,LLUM,LPH,LLPH)4/, XA (all)
XTØ	XA (all)
PCL1	XA (all)
PWL1	XA (all)
PQH1	XA (all)
PQH1	XA (all)
PXL1	XA (all)
XTØ1	XA (all)
PCL2	XA (all)
PWL2	XA (all)
PXL2	XA (all)
PQH2	XA (all)
XTØ2	XA (all)
PE	PE (EXGI,SARG,PHIL)
QYC	A (all)
YS	YS (all), HR (accounting)3/
V	HR (accounting)3/
H	HR (accounting)3/
K	HR (accounting)3/

2/ These are necessary only if the consumption function includes a wealth effect.

3/ If wealth does not enter the consumption functions and interest flows are not calculated under HR, these may be ignored.

4/ These are necessary only to the extent that the lag facilities are used in the functions. If lags of up to 1 year only are used, only X, PCLØ, PWLØ, PX, XTØ and PQHH need be correctly initialized.

Macro variables	Options where used
EX	XA (all) 1/, PX (LPH,LLPH,LUM,LLUM,LUK,LLUK) 1/
EXL1	XA (all) 1/, PX (LPH,LLPH,LUM,LLUM,LUK,LLUK) 1/
EXL2	XA (all) 1/, PX (LPH,LLPH,LUM,LLUM,LUK,LLUK) 1/
PCE	PE (PHIL,SARG)
PCEL	PE (PHIL,SARG)
RET	PE (SARG)

In addition to the variables both parameters and converters must be considered. The functional notes describe the situation with the parameter matrices - almost invariably initialize them to values for the year preceeding the first projection year, i.e. allow for updating before evaluation of the functions. The converters should also so far as possible refer to the initialization year and not the first projection year. The following converters are used during UPDATE and so for consistency must refer to the initialization year: QYC, MQC, QXC, YQC, VKC, HRC.

1/ If lag facilities are being used.

REDUNDANCY

So far as I can tel, IDIOM Release 3, Version 1 makes no use at all of the following:

classifications	S - stock-building assets F - financial assets	set each to 1
variables	QVAT - rates of value-added tax by commodity S - stock-building by asset	
macro-variables	(each of which is equivalenced to an element of $\bar{0}$)	

GDPB = $\bar{0}$ (131)

SCB = $\bar{0}$ (132)

SGB = $\hat{0}$ (133)

SVB = $\bar{0}$ (134)

SSB = $\bar{0}$ (135)

SXB = $\bar{0}$ (136)

SMB = $\bar{0}$ (137)

EMB = $\bar{0}$ (138)

I also believe that certain lagged macro-variables are also unused at present.

There is nothing to be done about the redundant variables. Merely set them during initialization and then ignore them.

INTERPRETING A DUMP

The dump contains information on:

TABLS which stores information on the location of variables in store. Each quartet of information contains:

xx n1 n2 n3

where:

- xx is the variable name (including converters and parameter matrices)
- n1 is its base address in the store ZZ
- n2 the number of elements it contains
- n3 the first dimension of xx

TABLE dimensioning information. Each quartet contains:

cc Ncc n1 n2

where: cc is the identifier of a classification

Ncc is the name of its associated dimension

n1 is the value of that dimension

n2 is address of corresponding titles

TABLF which gives information on functions. Each line of information contains:

ff n1 n2 n3 a1 a2 b1 b2 ...... h1 h2

where: ff is the function name

n1 the dimensions of the function packed into one figure as (1000 x first dimension + second dimension), or first dimension if there is only one. e.g. 4002 indicates a function for a variable (4 x 2).

n2 the number of parameter spaces reserved for each flow in the parameter matrix for this function. Negative of this if parameter matrix has been compressed.

n3 the base address in JSW of the vector of switches for this function.

a1, b1,...h1 key words denoting function options

a2, b2,...h2 the size of parameter matrix that each key word requires.

TABLG information on converters. Each quintet of information contains:

xx n1 n2 n3 n4

where: xx is the name of the converter

n1 its form of storage
1 for sparse
2 for full
3 for identity
4 for add

n2 its first dimension

n3 its second dimension

and n4 the number of elements

JSW the store for switches interpreted according to TABLF. These are the values of JSWH, JSWQM etc. referred to in the parameter tables earlier in this handbook.

ZZ the main store, interpreted by information in TABLS. Hence for instance if TABLS gives the following line:

DAB 446 10 2

it means that the 10 elements of ZZ starting with element 446 refer to DAB.

Note: (i) the first number in each row is merely a line number to help reading the dump.

(ii) between each pair of vectors/matrices in ZZ, is a delimiting element, set to .927E - 76. It may be used by the user to ensure that no vector has over-run its store.

TABLO gives the values of each macro variable.

THE DATA REQUIREMENTS FOR A RUN

As already indicated, IDIOM allows models of varying complexity to be constructed. The data requirements fairly obviously vary with the complexity of the model, so it is impossible to define absolutely what is required. This subsection, therefore, offers merely a sketch of the sort of information required.

Before embarking on the description of the data, however, a few words should be devoted to the distinction between simulation and estimation models. Simulation models explore the consequences of particular environmental or policy changes on the solution of a set of equations. The model amounts to no more than a machine for generating equation solutions. Obviously the answers produced depend on the input, but subject to a few elementary accounting restrictions, there is no need for the inputs to represent any particular year or, even, economy in reality. Estimation models, or exercises, on the other hand, are designed to exploit (past) data on an economy with a view to learning something about its behaviour. Obviously behaviour can be examined only for those phenomena on which data are available and the quality of the conclusions depend crucially on the quality of the data. However, in a simulation model, although the worth of the results is obviously increased by a close link with reality via good data, there is room for "stylised facts". By the latter I mean information which, while according with the model-builder's intuition or informal knowledge, has no firm basis in data sources. Thus, for instance, we may not know how, say, normal weekly hours evolve through time, but it might still be useful to conduct simulation exercises on the growth productivity under the <u>assumption</u> that they decline by, say, 1/2 per cent per annum.

It is also the case that simulation models allow room for "strategic simplification". (This is also important in estimation.) By the latter I mean the omission of various factors deemed by the model-builders to be of minor significance. For example one might feel that exports of a particular good are small enough to ignore, and one would therefore set them to zero in the simulation exercise.

The significance of the previous two paragrpahs in the present context is that although IDIOM's data requirements may seem formidable, it is not always necessary to have precise information on all factors in order to get useful results.

The best way to introduce the data requirements seems to be to describe a set of increasingly sophisticated models and the sort of information they require. I do not provide an exhaustive list of data, but merely a rough guide. To compile a full list for his own particular model, the user should consult the function specifications, the list of converters, TABLŌ and TABLS.

(a) The Leontief Static Model

The simplest model available under IDIOM is the Leontief Static Model. It requires the following data:

(i) QYC, with identical calssifications for Q and Y;

(ii) at least one final demand catagory, e.g. QC, which is set exogenously to the final demand vector (f, in standard notation)

To achieve this: use the Q classification for C (QCC is IDEN)

calculate C by FIX (fixing DPOP appropriately)

fix convergence on C

set C to total final demand, set SC to its total

additionally: set all prices to 1

set as many other functions to FIX, or very simple forms

set macro-variables to zero.

This may be extended trivially by allowing for more (exogenous) elements of final demand, and/or allowing final demands to have different classifications - this involves additional classification converters. The exogenous data may include government current and capital expenditure.

(b) The Production Sector

A natural extension of the Leontief model is to trace the implications of the gross output Y for employment. This may be simply done through the PFIX option of function YE - which requires estimates of output per head for the projection year. If more complex functional forms are used, lagged values of Y and YE will certainly be necessary plus possibly information on hours etc.

(c) A "Dynamic" Model

We now introduce our first feedback effect - investment. This produces a dynamic model, very similar to Leontief's dynamic model. If one is to use a simple accelerator model, investment by industry and asset is determined by changes in output - hence lagged Y is certainly necessary. To obtain even vaguely acceptable accelerator parameters, some information on YV will be required. More sophisticated investment functions will involve longer lags on Y and possibly price information, which we have so far omitted.

(d) An Income Loop

The other essential feedback to get a realistic model is an income-consumption loop. This involves two further substantial blocks of information. First we need to translate employment into incomes and disposable incomes, and second we need a consumption function.

From employment to income involves fixing a wage for industry, AWY, and also filling out the income flow converters RYB etc. The latter requires information on wage differentials between industries. We then also need to translate receipts (of which wages are so far the only element) into sectoral incomes (households) by filling up HRB. If, furthermore, the user wants a direct tax system, he has to define income classes, class limits etc. This is often a fairly complex task.

Second comes the consumption sector. The consumption function requires population information and, possibly, wealth. Using these it calculates total consumption from PDI etc. To fully exploit the consumption function, one may now wish to free the allocation of expenditure over goods (if we persisted with FIX for C, the fixed values would be pro-rated to sum to the endogenous SC). This requires coefficients but no more data.

(e) Prices

So far we have fixed prices to unity. Clearly this removes an important element of the economy. The easiest prices to introduce are import prices - in foreign currency terms there are exogenous (but are required on the Q classification, remember). Thereafter, one requires an exchange rate, some parameters and possibly local price information. Local price information is constructed from data used so far and requires no further input apart from parameters, and, if they are to be recognized, various taxes. For instance, unit labour costs include taxes on labour.

Once prices are introduced it becomes more important to include indirect taxes. These are entered largely as rates, so values for the projection period must be available. Also, however, implicit rates for the base year of the price indices are required.

(f) Foreign Trade

Import prices have been mentioned already. Imports follow from them and previous information, although possibly tariffs will be needed. Remember QM uses the Q classification.

Exports and export prices will probably require various foreign data on the X and A classifications. These are not always easy to find, but since they affect only exports, the model is not greatly affected by their absence - merely fix the export functions so that they are sensitive to local prices and, with assumed values of foreign data, fix the constants to produce plausible export figures. In general, the export sector is rather greedy for data.

(g) Further Details

Included in "further details" are various "frills", such as the corporated tax system, various payments and receipts functions, the direct imports and employment by consumers, industry and government etc. These are not necessary to the bulk of the model although of course they greatly enhance its applicability.

Some General Observations on Data

The data on economic variables fall into three classes:

(i) initialization values - mainly the year before the first solution year.

(ii) policy variables - for the solution years.

(iii) exogenous data - e.g. foreign data variables fixed exogenously for convenience.

For the initialization year, it is best to start from a full set of disaggregated accounts if possible. If not, a set of macro national accounts plus some ad hoc disaggregation should suffice, for as mentioned above, many of the initial values are not used. Use the information above on initialization to identify crucial variables and concentrate on getting these right.

The remaining two classes are obvious enough to handle. They need to be as accurate as possible if the simulations are to be sound. Whenever disaggregate values are provided, remember also to fill in the corresponsing macrovariables compatibly.

The major data requirements concern the parameter matrices and converters. In many respects the latter are the more difficult for they cannot be avoided nor greatly simplified. In choosing classifications, always bear in mind the need to convert from one to another. The parameter matrices range from simple fixed values or proportions through to the products of sophisticated estimation exercises. After reading the earlier parts of this section, the user will know what he can achieve and what data that he requires. At the very least, one past year's observations will be necessary on each of the variables to be endogenized.

PRACTICAL ADVICE

1. Use all diagnostic switches when starting to set up a run.

2. Initialize all variables, macro and vector. The full list is given in JRNS.IDIOM.TABLS. This is best done by drawing up a full set of accounts (SAM, if possible; certainly national accounts). If this cannot be done for the initialization year, do so for the first projection year and make small adjustments to move back a year. Worry about the accuracy only for the essential initial values (where lagged values are required for functions or updating accounts); otherwise roughly plausible but consistent accounts will do.

3. Annotate input streams in detail. If you cannot read annotated streams into IDIOM, store streams with data in, say, columns 1-70 and comments in columns 71-80. Then temporarily strip the comments with an editor just before reading the input streams. Also leave plenty of space in the input streams to improve legibility.

4. It is easier to interprete IDIOM diagnostics if your command stream has only one command per line. (This means just one command key word per line, not that lists should comprise only one item.)

5. Always start by projecting a year for which you have fairly full data, in order to check the calibration of the model.

6. On starting to set up a job, it is probably best to set SWITCH ABORT -1, so the job can run as far as possible without stopping. However, the corollary of this is that IDIOM gets stopped by a fatal error in the operating system and so command cannot be handed back to IDIOM to print out the dump or the trace. Hence, if you cannot interprete the error message of a non-fatal error, set SWITCH ABORT so that IDIOM stops on that error,[1/] and also set SWITCH TRACE and SWITCH DUMP.

7. Great care must be taken to ensure that the input commands and input streams are perfectly compatible. If they are not the whole internal structure of IDIOM can be corrupted. It is possible to check input at least partially using the DISPLAY and DUMP facilities.

1/ i.e. if the error is the third non-fatal error, set SWITCH ABORT 3.

III. THE STRUCTURE OF IDIOM

This section is designed to give the user a feel for the internal construction of IDIOM. It is not a complete guided tour of the programmes, but rather a sketch map so that the experienced FORTRAN user can find his way around the structure. Doing so will be helpful for certain diagnostic tasks and also if ever the IDIOM user wishes to alter the programme. The bulk of the information required for these tasks appears in the manual - see especially the Appendix - and this section must be taken as complementary to that paper rather than substitutable.

AN OVERVIEW

IDIOM comprises four basic phases: INITIALIZATION, INPUT, SOLUTION and ANALYSIS. The third may be termed its "economic" segment while the rest are basically house-keeping functions. Almost invariably the IDIOM user is going to be primarily concerned with the economic functions, although changes there may involve him in making complementary housekeeping changes.

The INITIALIZATION phase sets up the dimensions of the model to be used, defines functions, etc. The INPUT phase takes the store set up previously and fills it with data, converters, and parameters. The SOLUTION phase is where the economic model is solved and comprises an UPDATE section executed once per year in the solution period and a COMPUTE section over which the programme iterates to solve each year. The ANALYZE phase provides printouts, etc. of the results. Finally, there is also a FINISH phase which merely closes the model down. The structure is illustrated in figrue 3.

Control is maintained by a master programme IDIOM which calls in turn sub-masters for each of the main phases: INITIL, INPUTM SOLVEM (which then calls UPDATE and COMPUT) and ANALYM (which calls TABULA). Data are passed between the master and sub-masters by means of common blocks, but within the phases most information is passed through argument lists. The nature of the common blocks is described below. The relationships between the routines are

Figure 3: <u>The Phases of IDIOM</u>

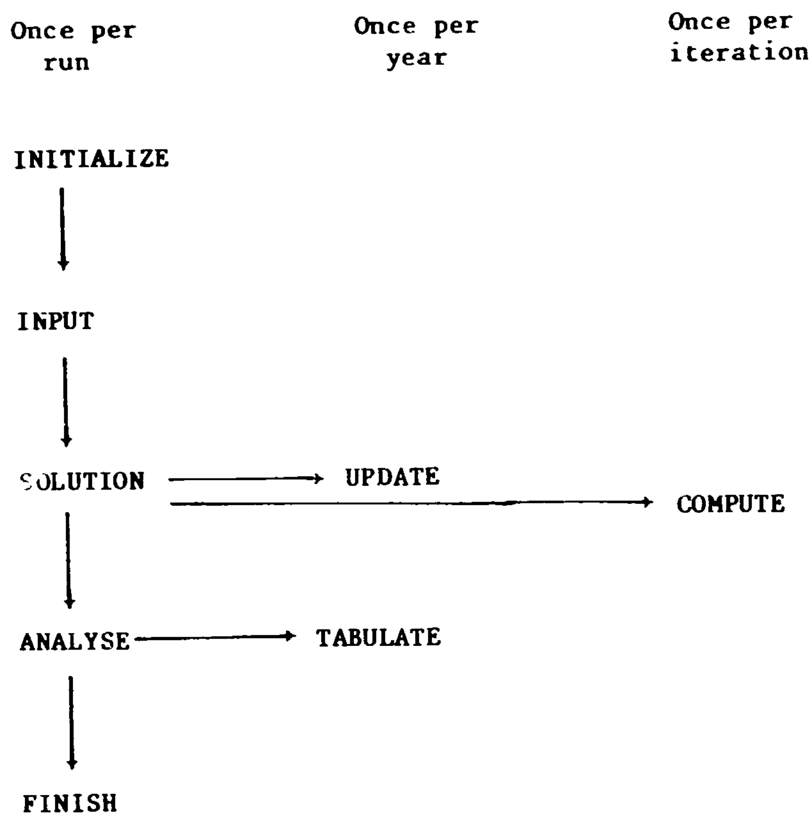


Figure 4: THE PROGRAMMES OF IDIOM

Master: IDIOM (Main) — BLOCKDATA

Sub-masters: INITIL, INPUTM, SOLVEM (UPDATE, COMPUT, (UTILS)), ANALYM (TABULA)

Sub-programmes:

INITIL	INPUTM	SOLVEM	UPDATE	COMPUT	(UTILS)	ANALYM	TABULA
HASHCM	SELECT	CONDIT	IDMUA	IDMSA	CCVERT	AGGRX	IDMTA
SWCHCM	SETMAC	CRITRN	IDMUDC	IDMSDC	IDMUX1	ARITHM	IDMTC
FUNCTN	READHI	LEVEL	IDMUHR	IDMSHR	IDMUX2	DEFINE	IDMTDC
CONVRT	READHX	SETDBG	IDMUPE	IDMSLC	INDLOC	DXFORM	IDMTHR
ALLOC	READLO	SETORD	IDMUPM	IDMSPC	JNDLOC	GETSER	IDMTLC
DSPFNC	GET	STLOOP	IDMUPX	IDMSPE	SPA	IDMGPH	IDMTMC
READER	NEXTBL	TSTCON	IDMUPY	IDMSPG	SPR	MDPNCH	IDMTPE
ERROR	ABCHEK	UPDLAG	IDMUOM	IDMSPM		PLTSFR	IDMTPM
FINISH	DISPLY	WRITHI	IDMUSC	IDMSPV		REVPOL	IDMTPX
NAMECM	DXPVAR	PUTGET	IDMUUE	IDMSPX		TABIDM	IDMTPY
SYMTAB	FREERD	XQLOOP	IDMUV	IDMSPY		TIMSER	IDMTQ
NUMBER		CCNTRL	IDMUXA	IDMSQ		TYPSYM	IDMTQM
MMLIST		DSPMAC	IDMUXB	IDMSQM		WRHEAD	IDMTQ4
TRUPAK		DEBUG	IDMUYE	IDMSSC		WRHEAD	IDTDSC
INTAPH		COMMNT	IDMUYS	IDMST		SETINT	IDMTT
INTAP		DECMOD	IDMUYV	IDMSTA			IDMTSC
REALPH		MODIFY		IDMSUE			IDMTUE
ARREF		IDMADJ		IDMSV			IDMTV
ARREP		DSPMOD		IDMSXA			IDMTXA
COMPRS		CMPFUN		IDMSYE			IDMTYE
				IDMSYS			IDMTYS
				IDMSYV			IDMTYV

described by figure 4 below. The sub-routines specified below each sub-master are those stored with it in the FORTRAN programme files; by and large, sub-routines are used primarily by the sub-master to which they have been allocated, but obviously some are called in several sectors of the programme. The branch headed (UTILS) are utility programmes used mainly by UPDATE and COMPUT, but which have been stored separately in the file JRNS.IDIOM.UTILS. A full guide to the calling hierarchy may be found in the overlay segment JRNS.IDIOM.OVERLAY.

HOUSEKEEPING

This section describes very briefly the main features of the housekeeping sections of IDIOM. It should be read in conjunction with the Users' Manual and the programmes themselves. The latter are extensively commented. This section of the handbook provides merely a broad outline of the structure.

(a) THE STORES

As stated above, information is stored and passed between sub-masters by means of common blocks. The data are all kept in a huge array - called ZZ - in blank common, while information on addresses in ZZ, dimensions, types, units, etc. is stored in other smaller common blocks - most important of which is COMMON/TABLES/. The basic sizes of ZZ and the accounting common blocks are set up in a BLOCK DATA segment attached to the master programme IDIOM; and this also initializes certain variables like key words and counting variables. Changing the size of IDIOM is described in the User's Manual. The macro-economic variables are stored in COMMON/MACRO/ in array O and with mnemonics given in TABLO.

Data are stored in ZZ in the rough order: data variables, parameter matrices, full converters, work-space, and finally sparse converters. The work spaces are four vectors Z1,...Z4, which are used as work-space wherever necessary. Their dimension is the largest of the dimensions declared under the # command. Each array in ZZ is separated from its neighbours by the delimiter 0.927E-76, and major blocks of data start only on "ten word" divisions, i.e. 1, 11, 21, ...

The accounting arrays store the following information. They are built up from the BLOCK DATA segment (defining literals) and the commands.

array	subject
TABLS	information on variables in ZZ[1/]
TABLE	dimensioning information for classifications[1/]
TABLF	function definitions[1/]
TABLG	converter information[1/]
JSW	switches for controlling function specifications[1/]
TABLŌ,	macro information[1/]
DIMENS	dimensioning information for housekeeping processes
CONTRL	control variables
CMCARD	control information interpreted from the directives input
IDM4	scalar work space

(b) INTERPRETING DIRECTIVES

Directives are read at every stage of IDIOM. The subroutine READER reads the card image input and starts the interpretation off by breaking the input up into single commands. Having isolated a command in array COMM, control is then handed back to the calling segment where appropriate action is taken. The appropriate action is determined by firstly converting the command into A4 words and then comparing these with the key words defined in the block data segment. The former operation is done by TRUPAK, while the latter is done by means of the logical operator, .EQ. Obviously any subsequent switching depends on the circumstances of the particular phase and command. READER and TRUPAK use "blank" as a delimiter; any syntactically independent elements should be separated by blank; e.g. IF⌢YEAR⌢=⌢2⌢REAL⌢EX⌢.9

1/ For more information on these arrays see "Interpreting a Dump" above.

(c) READING DATA

During INPUT mainly, but also during SOLUTION, IDIOM reads data from various streams. The facility used by IDIOM in Cambridge to do this is the "list-directed read" supplied by certain compilers which allows various convenient simplifications in input (e.g. the use of n*0. to denote a string of n "Ø."s). This is not universally available, however, and so UNIDO has replaced it by a locally written routine in ASA FORTRAN which has the effect of permitting the simplifications under more restrictive compilers. The routine is called FREERD. It produces substantial amounts of output by which the user may check the accuracy of the reading process; these are illustrated on page 164 below.[1/]

The INPUT routines are necessarily complex because of the large number of directive-determined options that are required. However, the principle is fairly straight-forward. The READ command and its options are identified by the means just described. IDIOM then checks that input of the sort requested is legitimate and assuming that it is, switches control to the requisite routine for that sort. Here the location for the data is found and prepared for input and then the relevant mode of input is summoned, through the routine READLŌ. This fills up the array ARRAY (and its equivalenced integer array IARRAY) with the data from the relevant stream and then hands this back to the calling routine which installs it into ZZ appropriately. Many checks and much switching is necessary to ensure that the data are not subsequently overwritten.

(d) ERRORS

IDIOM traps many errors itself and in each case prints out a unique message. The error directory - stored in JRNS.IDIOM.ERRORS - provides a dictionary of error messages. It comprises the error message and, on the right, a code number. During IDIOM runs, the error code numbers are

---

1/ A new version of the routine has recently been developed which inserts delimiters into the data stream. The annotation can be placed into the delimiter line. (See Appendix D) printed out as four digit numbers; unfortunately the index gives only the first three, the remaining digits at the right of the data set being IBM record card numbers. The messages are arranged, however, in order of ascending code numbers so it is easy to deduce the fourth digit.

The error code numbers are based on the position in the FORTRAN source of the call to subroutine ERROR that outputs the message. The first digit refers to the FORTRAN file, which broadly correspond to phases

0	IDIOM
1	INITIALIZE
2	INPUT
3	SOLVE
4	ANALYSE
5	unused
6	UPDATE
7	COMPUTE
8	TABULATE
9	UTILS

The second and third digits refer to the position of the subroutine within the file: hence

7001 refers to the main subroutine of COMPUTE which is COMPUT

1032 refers to the fourth subroutine of INITIALIZE, since counting is from zero, i.e. FUNCTN.

The fourth digit distinguishes calls from the same subroutine. Hence, the last example was the second error call in FUNCTN.

Error messages are, with one exception, always preceded by ** and they give information on the type of error, its code number, the number of instructions processed, the number of directive cards read and the card current when the error occured. IDIOM also indicates the place in the current command where it has got stuck. This pointer refers to the next character it would have processed if it had been able; hence the error will normally have resulted from the immediately previous element of the command. Hence, if the card

COMPUTE QMØ,PQM

gives trouble because of a mistype in compute; say CØMPUTE was typed, the error message gives

COMPUTE Q MØ,PQM

where < points at the next character for processing. The CØMPUTE has been decoded, as has its delimiting space, but cannot be interpreted. Hence, IDIOM stops, with the pointer showing that Q is the next character for decoding.

There is an unfortunate quirk in the case of the previous example. One would expect it to generate error 3001 "COMMAND CANNOT BE FOUND IN SOLVE PHASE", but in fact, control gets passed back from SOLVE to subroutine IDIOM prematurely and the programme stops with 0000 "ILLEGAL PHASE CHANGE". (This should be corrected at some stage.)

One other idiosyncracy concerns UNIDO's local list-directed READ - FREERD. It does not use IDIOM's error facilities so its errors are not numbered and are not preceeded by **.

(e) TIME-SERIES ANALYSIS

This is carried out under ANALYM, largely by the routine TIMSER. The options are defined in the User's Manual. For most options, after they have been identified by the decoding process, TIMSER calculates the form of the variable to be output (e.g. growth rates) at the end of the stack ZZ. It then calls the printing routine TABIDM to produce the final table. If graphs are required PLTSER constructs the variable(s), whilst IDMGPH produces the output. The DEFINE facility, which involves complex programming, translates define statements into reverse polish logic in REVPOL, checks their legitimacy (TYPSYM) and executes them (DXFORM). The whole is supervised by subroutine DEFINE.

(f) CROSS-SECTION TABULATION

IDIOM has a cross-section tabulation corresponding to each function, which is identified by that function's name, and one for macro-economic variables and condensed national accounts, identified as MACR. These output routines are fairly transparent, since each function has its own routine, and the whole operation is controlled by TABULA. Basically each routine outputs the values of the variable concerned, fully disaggregated, along with as much additional information as is feasible. A list of the variables given in each table appears in table 4 below. The subroutine handling function xx is called IDMTxx.

I should add here a note about the sectoral summary output by the TABULATE MACR or ALL controls. This presents a summary of the full output on a four sector aggregation. Unfortunately, however, the aggregation from the sectors (commodities) in the main printout (NQ) to the four used here is given to IDIOM in a local DATA statement within IDMTQ4. Hence the user has to specify the dimension and contents of integer array IAGG and recompile IDMTQ4. IAGG(I) = 1, 2, 3 or 4 according to whether commodity I falls under primary, manufacturing, construction or service sectors respectively. If IDIOM is being run with fewer than four commodities, I suggest removing the call to IDMTQ4 within TABULA, by inserting a C in column 1 of the cards concerned.

Table 4: DATA OUTPUT BY TABULATE COMMAND

TABULATE	variables output	
QMØ	import vomume	QMØ
	duties at base rates	QMTZ*QMØ
	output vomume	Q
	import ratio	QMØ/Q
	quota rates	QMQ
	import values	QMØ*PQM
	import price	PQM
	production price	PQ
PM	world price	$\widetilde{\text{PFM}}$
	import price	PQM
	production price	PQ
	absorption price	PQH
	tariff rate	QMTØ/QMØ
	import volume	QMØ
PY	production price	PQ
	home prices	PQHH
	import prices	PQM
	export prices	PQX
	absorption prices	PQH
YV	output volume	Y
	price	PY
	output value	Y*PY
	profits	YP
	disaggregate investment (up to 3 asset groups)	$\widetilde{\text{YV}}$
	total investment volume	YVØ
	total investment value	$\sum^{NY}$ YV*PV
	average price	ratio of value/volume
YEØ	output volume	Y
	investment	YVØ
	employment	YEØ
	hours	YH
	wage rate	PYE
	% growth of men	% ΔYEØ
	hours	% Δ(YEØ*YH)
	productivity	% Δ(Y/YEØ*YH)
YS	output	Y
	Δ output	Y-YL1
	stock-building volume	YS } disaggregated by stock type
	stock-building value	YS*PYS } disaggregated by stock type
	value of stocks	BYS(/,NBYS-2) } disaggregated by stock type

Table 4 cont'd

DC	(a) summed over income groups			
		commodities	i [QCC]*C	at constant prices
		imports (direct)	RCB(,"IMP")*C	
		other value added	C - rows 1,2 and 4	
		tax (base year rates)	CTZ*C	
		total volume	C	
		imports (direct)	RCB(,"IMP")*C*PSCM	at current prices
		value-added	RCB(,"WAGE")*CEØ*AWC	
		taxes in total (value)	CTØ	
		taxes disaggregated by tax type		
		total value of consumption	PC*C	
		price	PC	
	(b) consumption by each income bracket		DC	
DSC	(a) by income class			
		volume of consumption	DSC	
		value of consumption	DSPC	
		price	DPCE	
		volume p.c.	DSC/DPOP	
		value p.c.	DSPC/DPOP	
		no. of consumers	DPOP*1000.	
		PDI p.c.	DPDI/DPOP	
		RPDI p.c.	DPDI/(DPOP*DPCE)	
		total wealth	DPW	
		wealth p.c.	DPWOH	
	(b) all consumers			
		volume of consumption	SC	
		value of consumption	SPC	
		price	PCE	
		PDI	PDI	
		RPDI	RPDI	
		expected RPDI	EPDI/PCE	
		% Δ PDI		
		% Δ RPDI		
		% Δ consumption		
		wealth	PW	
R		NO PRINT OUT		
X		exports by group and area	X	
		export prices by group	PX	

Table 4 cont'd

PX	export prices (current, lagged 1 and 2)	PX,PXL1,PXL2
	export taxes	XTØ
	competitors' prices (current, lagged 1,2,3)	PCLØ,PCL1,PCL2,PCL3
	export vomume	$\sum^{NA}$ X
PE	average wage (current, lagged)	PE,PEL1
	total employment	E
	total unemployment	UE
	% Δ	%Δ PCE
	%Δ retentions	%Δ RET
T	taxes by expenditure category (O classification)	TŌ
V	social investment, volume	[VKC]*K
	industrial investment, volume	V - row 1
	total	V
	social investment, value	row (I-3)*PV
	industrial investment value	row (I-3)*PV
	total	row (I-3)*PV
	asset price	PV
	total tax	VTØ
	tax rate	VTØ/row 6
YULC	output, value	Y*PY
	material inputs, value	[QYC] *PQH
	wage bill	RYB(,"WAGE")*YEØ*AWY*0.001
	profit	YP+YSA
	direct imports	RYB(,"IMP")*Y*PSYM
	other inputs	YRØ - YTØ - rows 3,4,5
	taxes on labour	TYB(,"EMPW")* row 3 + TYB(,"EMPE")*YEØ
	taxes on output	YTØ - rows 7,9
	taxes on input	TYB(,"ADVI")*Y*([QYC]*PQH)
	total taxes	YTØ
	unit labour costs	(row 3 and 7)/Y
	average wage	1000.*row 3/YEØ
	industrial price	PY

Q (a) commodity balances

	consumption	QCØ
	government	QGØ
	investment	QVØ
	stock-building	QSØ
	final demand (home)	Σ above

Table 4, Q (a) cont't

	intermediate demand	QYØ
	home demand	rows 5 + 6
	exports	QXØ
	total demand	row 7 + 8
	import supply	QMØ
	import duties	QMØ*QMTZ
	total output	Q
	total supply	rows 10 + 11 + 12

(b) non-commodity demands and supplies

	direct imports indirect taxes other value-added foreign tourist expenditure	disaggregated by the classes of supply and demand used in part (a) of the table
UE	industrial employment	EYØ
	government employment	EGØ
	household employment	ECØ
	total employment	E
	registered unemployment	UE
	unregistered unemployment	LF - E - UE
	total unemployment	LF - E
	labour force	LF
	registered unemployment rate	UE/(UE+E)
	total unemployment rate	row 7/LF
HRP	reciepts by type and sector	HRR
	payments by type and sector	HRP
H	NO PRINT OUT	
QYC	input-output flows at constant prices	$[QYC] * [\hat{Y}]$

THE ECONOMIC ROUTINES

The economics of the economic routines has been described at length already. This section is concerned with their programming.

The SOLUTION phase is controlled by subroutine SOLVEM. It interprets directives and passes control according to them. It is basically a housekeeping routine, having more in common with INPUTM than anything else. Two of the routines it calls should also be thought of as submaster programmes for they control other programmes. These are UPDATE and COMPUT, which preside over the economic solution of the model.

UPDATE is executed once per year; it builds up those components of the endogenous variables that do not change from iteration to iteration. For example, it handles time trends, regression constants, lagged and exogenous variables and cumulated variables. The order in which it treats functions is controllable by the user, although he will rarely wish to alter it.

Part of the UPDATing proceedure required by IDIOM is to update lagged exogenous data; so that what was once PCLO, say, becomes for the next year, PCL1. This process is undertaken by UPDLAG making use of TABLL which contains information on the location and nature of all lagged variables required by IDIOM. It is necessary to execute UPDLAG before reading fresh values of the exogenous data for otherwise the previous current value will be lost.

UPDATE, which handles function updating, presupposes that current variable store contains data from the last solved (or initialized) year, and correspondingly, for lagged values, that, say, PCL1 contains PC from two years ago, etc. Consequently UPDATE must be executed before UPDLAG. This is guarranteed if the directive UPDATE ALL is used, but not otherwise.

The UPDATing of lagged endogenous variables, including lagged macrovariables is undertaken at the end of UPDATE. Again, information in TABLL is utilised.

UPDATE keeps a record of what has been updated in order to prevent things from being treated twice and to flag things that have been omitted. Also where parameter matrices have been compressed, UPDATE "decompresses" them (at the end of the stack, ZZ) before entering the routine that uses/updates them, and then "recompresses" them afterwards. This is done by CMPFUN.

The computations within the iterations are controlled by COMPUT. As with UPDATE, COMPUT tests that every function is computed precisely once, decompresses and recompresses parameter matrices, and allows changes in ordering. It also permits debugging tests and calculates the convergence tests.

MAKING CHANGES

The User's Manual describes the technical aspects of making changes to IDIOM. This section offers some advice of a more homely nature.

Changes to existing functions are by far the easiest to execute. They involve alterations to functions in UPDATE, COMPUT, and conceivably, TABULATE sections of the programme. Before making changes ensure that the existing programme is perfectly understood and ensure that no dimensions are to be changed. Thereafter it is just a question of care and thorough testing.

Next in the order of complexity are changes involving new variables - either exogenous or endogenous. These require, in addition to the steps for the simple changes, alterations to the various tables. This is so that IDIOM knows how much space to devote to the new variable etc. Obviously if the new variable is endogenous, new subroutines in UPDATE and COMPUT are required also. The IDIOM tape contains a preprocessor programme to help with the introduction of new variables.

Finally there are changes to the housekeeping operations. Neither of the other sorts of change is for the beginner to carry out, but these housekeeping changes, in particular, require great experience. The complexity of any change depends on its nature, but one should not consider making any alterations until the present structure is toally understood. The housekeeping programmes are extensively interlinked and one should be sure to have found all uses of a particular routine before editing it. Care should also be taken not to violate the overlay patterns where OVERLAY is used.

It is not worthwhile to give examples of FORTRAN changes - each user's circumstances will be unique - but we might note here that it is possible to check individual economic subroutines using the directive stream alone. For instance, suppose that we wished to test a new formulation of the import functions, which involved changing IDMSQM. We may use the instructions:

```
READ  Q,QM,QX,PQM,QMTØ,PQ
FOR ITER 1 TO 1
COMPUTE QMØ
DISPLAY VARS QMØ
DUMP
ITER
```

This first line initializes <u>current</u> data to convenient values. (Since IDIOM allows one to input values for variables several times over, there would be no need to remove the read statements used in the basic run - i.e. the run without the contemplated changes. The additional read commands and data may just be tacked onto the basic commands). It then enters the iteration loop, solves one iteration, and displays the results. The DUMP is not necessary, but it would allow one to check that BQM and the various data had been correctly constructed. After one interation, one could then check the results by hand.

<u>Note</u>: The data on lagged prices and tariffs would come from the initialization and would have been incorporated into the function during UPDATE. The extra read refers only to current variables.

IV: THE TEST RUN

In this section we introduce the test run. The directives and data for the run are provided on the IDIOM tape, and the first task of any potential user is to ensure that he can (a) reproduce the test run; and (b) understand what the various elements of the input and output mean.

As might be expected by any experienced-model builder, the preliminary experiments with IDIOM, on which this handbook is based, showed up certain infelicities in the test job and in IDIOM itself. Most of these have been corrected, but in order to expedite the production of this document, not all have been so treated. Consequently the test-run has some rather silly aspects, and these are noted as we go through it. Although they render the economics very dubious, however, they do not affect the purely computing aspects of the exercise, and so this section may still be used as a guide to working with IDIOM.

Nevertheless, if the user is to replicate the experiments outlined here, he will need access to the present version, warts and all. This is acceptable for personnel in UNIDO, where the current versions may be permanently stored on tape, but it is not satisfactory if IDIOM is to be transferred to other machines. Hence, for users outside UNIDO, I suggest that the errors known currently be corrected, and that this section of the handbook be rewritten using the correct model. As already indicated, this will change only the results, not the user-processes.

THE TEST RUN DIRECTIVES: TABLE 5

The directives for the test-run are stored on file JRNS.IDIOM.DIRECT on the IDIOM tape. This section describes then in detail - not just so that they may be understood *per se*, but in order that the user may appreciate at least some of the reasons why they are as they are.

Table 5: IDIOM DIRECTIVES

```
                         IDIOM - ORIGINAL - DIRECTIVES
SWITCH ABCHECK ; SWITCH NO ECHO ; SWITCH ABCRT -1 ; SWITCH DUMP
NAME USER AWAP RUN 1 TITLE 'Idiom version 3 final test run'
SETUP DUMP TABLES TITLES ; # C 11
# Q 4 ; # Y 4 ; # C 2 ; # G 1 ; # R 5 ; # H 4 ; # S 1 ; # V 2
# F 1 ; # X 2 ; # M 2 ; # E 1 ; # D 2 ; # T 3 ; # A 3 ; # K 1
FUNCTION QMO LLIN 2 FIXM 3,4 FIXQ 1 ; FUNCTION FQM LLIN ?
FUNCTION PY LMAT ? ; FUNCTION YV NCL ?:1 FIX ?:2
FUNCTION V INDL 1 LFEE 2 ; FUNCTION T AIVA 1 SPEC 2 SUBS 3
FUNCTION YEO RECU 1,2,3 LLIN 4 ; FUNCTION YS FIX ?
FUNCTION DC LES ? ; FUNCTION DSC LNMW ?
FUNCTION R WAGE 1 ITAX 2 PROF 3 IMP 4 GOOD 5 ; FUNCTION XA LLIN ?
FUNCTION PX LLUM ? ; FUNCTION PE EXGW ? ; FUNCTION QYC NULL ?
FUNCTION YULC ACTU 1,3,4 STAN 2 ; FUNCTION Q IDEN ?
FUNCTION UE SIMP ? ; FUNCTION HRP NULL ?
FUNCTION H HCUS 1 CCRP 2 CGOV 3 ROW 4
CONVERT R Y FULL ; CONVERT R C FULL ; CONVERT R G FULL
CONVERT Q Y FULL ; CONVERT C C FULL ; CONVERT M Q FULL
CONVERT Q G FULL ; CONVERT C X FULL ; CONVERT Q S FULL
CONVERT Q V FULL ; CONVERT V K FULL ; CONVERT Q YP FULL
CONVERT E YE ADD ; CONVERT E CE ADD ; CONVERT E GE ADD
CONVERT H R FULL ; CONVERT D P FULL ; CONVERT Q YS IDENTITY
COMPRESS PARS YV,HRP
INPUT
REAL SPX,DFE,WPOP,PW,PWFH,SPC 10000,50000,26000,50000,1000,31472
REAL SPYM,SPCM,SPGM 100,500,100 ; REAL WVCL 100 WFFI 1
REAL EMPY,EMPC,EMPG,EMPL,UNEM,EMB 20274,100,4365,24739,500,24739
REAL WAGY,WAGC,WAGG,WAGE 26000,130,6000,32130
REAL GDP,SC,SG,SV,SS,SX,SM 50785,31472,9095,9390,450,11275,10887
REAL GDPB,SCB,SGB,SVB,SSB,SXB,SMB 50785,31472,9095,9380,450,11275,10887
REAL PUC,PCE,PSG,PSV,PSS,PSX,PSM 1.,1.,1.,1.,1.,1.,1.
REAL PSYM,PSCM,PSGM 1.,1.,1. SYM,SCM,SGM  100.,500.,100.
REAL HUCL,PCEL,PSGL,PSVL,PSSL,PSXL,PSML 1.,1.,1.,1.,1.,1.,1.
REAL EX,EXL1,EXL2 1.,1.,1. ; REAL AW,AWY,AWC,AWG 1300,1300,1300,1300
REAL PDI,PPDI,PDIL,RDI1,RDI2 34622,34622,31612,567,533
INTEGER YA72,YA70 1,1 ; REAL SPM,SM 10887.,10887. FUP 1.
SELECT INPUT 2 CARDS
READ Q,QCO,QGO,QMO,QSO,QVC,QXO,QYO,PC,PCH,PCM,PCX,CMTO,CMTZ,QMQ
READ QXTO,QXTZ,Y,YL1,YL2,YL3,YEO,YEL1,YP
SELECT INPUT 2 GEM ; READ YV
SELECT INPUT 2 CARDS
READ YS,PY,PYVP,PVA,PYV1,PYV2,PYE,YRC,YTO,YTZ,C,PC,CRO,CTO
READ CTZ,G,PG,GRO,GTO,GTZ
SELECT INPUT 2 GEM ; READ HRR,HRP
SELECT INPUT 2 CARDS
READ H,RYO,RCO,RGO,S,V,VKO,PV,VTO,VTZ
SELECT INPUT 2 GEM ; READ XA
SELECT INPUT 2 CARDS
READ PX,M,PPM,E,EYO,ECO,EGO,D,TQMO,TYC,TCC,TGO,TVO,K,TQMB,TYB,TCB
READ TGB,TVB,TQXB
SELECT INPUT 2 GEM ; READ RYB,RCB,RGB,QYC,MCC
SELECT INPUT 2 CARDS ; READ YQC
SELECT INPUT 2 GEM ; READ CCC,CGC,CSC,QVC,QXC
SELECT INPUT 2 CARDS ; READ VKC
SELECT INPUT 2 GEM ; READ CYPC
SELECT INPUT 2 CARDS
READ TITLES Q,Y,C,G,R,H,S,V,F,X,M,E,D,T,A,K 2
READ PARS QMO LLIN 2 FIXQ 2 FIXM 2 QMQ 2
READ PARS FQM LLIN 2 FIX 2 PPM 2
```

Table 5: IDIOM Directives (cont'd)

```
                                    IDIOM - ORIGINAL - DIRECTIVES
READ PARS YV NCL 2 FIX 2 ; READ PARS V LFEE 2
READ PARS PY LMAT 2 LKEY 2 LLM 2 LLK 2 FVA 2 PRS 2
READ YH,YHL1,R,PM,T,TO,YULC,YEXP,CEO,CECE,GEO,GEOB,HPD,VRO,RVO,UE
READ LF,PE,PSL1
SELECT INPUT 2 CARDS ; READ RVB
READ PARS DC LES 2 LLIN 2 FIX 2 DUR 2
READ PARS PE PHIL 2 SARG 2 EXGI 2 EXGW 2
READ PARS DSC LEXW 2 LNMW 2 LRLW 2 LLXW 2 LLNW 2 LLRW 2
READ PARS UE VARP 2 ; READ PARS YEO RECU 2 LLIN 2
READ DN,DBT,H
SELECT INPUT 2 GEM ; READ DAB,DDB,DRC,HFC
SELECT INPUT 2 CARDS ; READ TCXO,PCHH,PYS,YSA
READ PARS YS FIX 2 ; READ AD1,AD2,DI1,DI2
READ PARS XA LLIN 2 ; READ PARS PX LLUM 2
READ PCL0,PCL1,PCL2,PCL3,PWL0,PWL1,PWL2,PWL3
READ PCH1,PCH2,XT0,XT01,XT02,PXL1,PXL2
READ DC,DPOP,DPDI,DSC,DSPC,DPCE,DPW,DFWH
REAL TXPD,TXPI,PCET 1.,1.,1.
REAL RET,RETL .3,.3 POP 55.765 RINT,RPOF .1,.1
INTEGER PBAS 1970 GEAS 1970
SWITCH PRINT ; SWITCH PAGE
INTEGER START 1973
dump
SOLVE
CRITERION Y 10.
FOR YEAR = 1 TO 7
UPDATE ALL
SELECT INPUT 2 CARDS
READ PCL0,PWL0,AD1,AD2,DI1,DI2
FOR ITER = 1 TO 40
COMPUTE ALL
LOOP ITER
SELECT OUTPUT 15 DUMP
PUT ALL
COMMENT ITERATIONS DONE AND RESULTS DUMPED
LOOP YEAR
SWITCH NO DUMP
ANALYSE
SELECT INPUT 15 DUMP
SELECT OUTPUT 3 PRINTER
GET ALL 1973
TABULATE ALL
GET ALL 1979
TABULATE ALL
VALUE C,Q,Y
PLOT SC,SPC
PLOT SPC AGAINST PDI
FINISH
```

Note: Many lines contain several instructions; they are separated by ";".

Line 1: Global switches

ABCHECK: checks array overflows. It is essential when new jobs are being set up, for it traps at least certain common dimensionary errors.

NO ECHO: does not reflect subsequent commands, It is probably best to set ECHO at first, but subsequently NO ECHO economizes on output. Note: NO ECHO is, in fact, the default setting.

ABORT -1: continues execution after any non-fatal error. This allows the job to proceed as far as possible; it will stop only if the operating system finds an error. A consequence of this is that if an error occurs, control is not handed back to IDIOM, and so, IDIOM diagnostic dumps will not be provided (see Practical Advice).

DUMP: provides a dump if IDIOM detects any error. Redundant if ABORT -1 is set, but essential in the early stages if ABORT n(n>-1) is set.

Other switches are set to default values.

Line 2:

Identifying Notes - obvious.

Line 3:

SETUP TABLES: an essential command instructing IDIOM to set up internal tables necessary for data. The key words DUMP and TITLE set up additional auxiliary files (not in IDIOM work-space) for dumping results and storing titles.

#$\bar{0}$ 11: the first of the # commands that set up dimensions. The value for #$\bar{0}$ has to be 11 for IDIOM version 3.

Lines 4-5:

Further # commands: for example, there are four industries, four commodities, two export groups etc. Note, there must be one # command for each IDIOM classification.

Line 6:

Import FUNCTIONS are defined. Commodity 1 has exogenous output, with imports meeting the difference between demand and supply. Commodities 3 and 4 have exogenous imports and commodity 2 a log-linear import function. (Note: the order of these declarations does not matter.) Import price FUNCTIONS are also defined, with log-linear equations for all commodities.

Lines 7-15:

Further function definitions - one for each of the functions defined in the handbook. Features to note include:

FUNCTION YV (line 7): YV is a two dimensional array, (NY, NV). For all industries asset 1 is determined by a neo-classical function, and asset 2 as a fixed value.

FUNCTION T (line 8): defines the order of the indirect taxes in the tax vectors: ad valorem taxes, specific duties and, finally, subsidies. The ways in which each tax type is handled is entirely determined within IDIOM; this function merely specifies their order in the various vectors and printouts.

FUNCTION DC: all commodities for each income group use the linear expenditure system, with parameters specific to each individual flow. This is ensured by the "LES ?:?" part of the command. The subsequent "UNIF ?:?" is entirely redundant in this case - because "?" means "all elements not yet specified", but since the LES takes "?", there are no unspecified elements for UNIF to handle. (If, on the other hand, LES took the control "1:?" - i.e. all commodities for income group 1 - the "UNIF ?:?" would entail using the parameters for income group 1 for consumption by income group 2 also.)

Lines 16-21:

CONVERT commands defining the storage method to be used for each converter. E.g. converter from Y to R, RYB, is to be stored full. Any matrix having fewer than one third of its elements non-zero may be stored in less space in SPARSE form than in FULL, although the packing and unpacking of sparse matrices involves a small increase in execution time. In lines 20 and 21, certain converter matrices are defined as either summation vectors (ADD) or identity matrices (IDENTITY). These require no store within IDIOM or data input, their operations being entirely defined by the nature of the matrix.

Line 22:

This COMPRESSES the parameter matrices YV and HRP. This is of no consequence to the user, but economizes on IDIOM's use of store. Note that no saving of store is made if only one matrix is compressed.

Line 23:

INPUT indicates the close of the INITIALIZE phase and the start of the INPUT phase. It could be followed by SWITCH instructions if different switches were required in the INPUT phase from those defined earlier.

Lines 24-34:

These initialize various scalar variables. Each REAL instruction comprises a list of n names followed by n real numbers, although where free-format directed input is available the decimal point in the latter is optional.

Line 35:

This initializes the integer scalars YA72 and YA70 and then, after ";", some further real variables.

Line 36:

This is the first of the SELECT statements which open data-streams for input or output. In this case stream 2 is opened for input in card image form (80 characters per line, max.). Matrices read under this instruction are read according to FORTRAN conventions; thus matrix X of dimension

(NX,NA), for instance, would be filled up in the order X(1,1), X(2,1), X(3,1)... X(NX,1), X(1,2), X(2,2)... X(NX,2), X(1,3)... X(NX,NA). there is no need to start each new column on a new card, although doing so may improve the legibility of input stream.

Line 37-38:

Read instructins for data in card image.

Line 39:

Stream 2 is selected again, but the argument GEM changes the nature of the reading. Under GEM matrices must be preceded by their dimensions and are read by rows: i.e. X(1,1), X(1,2), X(1,3)... X(1,NA), X(2,1)...etc.

Lines 40-54:

Various SELECT and READ instructions for variables and converters.

Lines 55-56:

Input from stream 2 under card image is selected again, but now titles are read. The "2" at the end of the line is not required here because stream 2 is the current stream, but it would be possible if required to have stream 2 open for data above and yet to take titles from, say, stream M. This would be done by replacing the "2" in line 56 by an "M".

Lines 57-60:

These lines read parameter matrices from stream 2. Again the "2"s are redundant since stream 2 is current. Each instruction involves several key words: first the name of the function then the names of the various functional forms to be read under that function. Note that, for each functional form specified, a complete parameter matrix is required (i.e. parameters for each flow). IDIOM reads these, discarding those rows that it does not require according to the FUNCTION definitions. For example, line 57, tells IDIOM to read four complete matrices of import parameters. It does so, and then selects the parameters for group 1 from the FIXQ matrix, those for groups 3 and 4 from FIXM and those for group 2 from LLIN. The LLIN specification requires 8 parameters, so BQM, the parameter matrix for imports, becomes a (4x8) matrix within IDIOM, with a full second row, but information in only the first column of the other rows. The information read for the QMQ option is discarded entirely.

Lines 61-77:

Further data reading. Note that the SELECT instruction on line 63 is redundant because it is identical to the previous SELECT. As is clear from these lines, the various read options may be intermingled.

Line 78:

Initializes PBS, QBAS

Line 79:

SWITCH commands for the SOLUTION phase: PRINT instructs IDIOM to output details of each iteration and PAGE instructs IDIOM to move to the top of the next page of the currently selected output stream.

Line 80:

Initializes the start year of the run.

Line 81:

This instructs IDIOM to dump the contents of the internal arrays out on stream 6. It allows the user to check that initialization has proceeded correctly. It would not be required in production runs.

Line 82:

SOLVE denotes the end of INITIALISE phase and the beginning of the SOLUTION phase. It could, without harm, have been inserted before the SWITCH commands on line 79.

Line 83:

This defines the convergence criterion. In this case only Y is used and convergence is said to have occurred when:

$$\sum_{I=1}^{NY} (Y_i (I) - Y_{i-1} (I))^2 \leq 10.$$

Where I counts over industries and i over iterations. Note that more variables could have been considered and also that in assessing the convergence criterion no scaling occurs; i.e. the criterion must be measured in the same units as its variable.

Line 84:

The years for solution are defined here. IDIOM is solved for seven years, 1 to 7, where 1 refers to the START year 1973.

Line 85:

UPDATE ALL instructs IDIOM to update all functions and to do so in its default order. Note that this is done before year 1973 is solved. Hence the initialization process should leave the store looking as though IDIOM had just solved 1972.

Lines 86-87:

Certain exogenous variables vary from year to year. IDIOM is instructed to read these from stream 2 in card image. In this case it is only foreign variables and expected output that are so treated, but in principle any variable could be reset here. Of course, stream 2 must contain the requisite information for all the years of the run.

Line 88:

The start of the iteration loop. Up to 40 iterations are to be allowed for convergence.

Line 89:

COMPUTE ALL tells IDIOM to execute all the SOLUTION routines once in the default order.

Line 90:

This denominates the end of the iteration loop. Note that it has the name used in the FOR statement to which it refers; in this case ITER.

Lines 91-92:

This selects an output stream as current. This is stream 15 which has been set-up as a binary dump file in line 3 above; the option DUMP instructs IDIOM that output to it must be in binary form. The actual dumping is initiated by PUT ALL on line 92.

Line 93:

This defines a comment which may be up to 80 characters long and which is output to stream 6 whereever the command is encountered.

Line 94:

This defines the end of the YEAR loop.

Line 95:

This SWITCHES off the dump on stream 6 on the detection of an IDIOM error.

Line 96:

ANALYSE ends the SOLUTION phase and commences the ANALYSE phase.

Line 97:

This SELECTS the binary dump on stream 15 for OUTPUT. The results dumped there, year-by-year, are now to be read in for analysis as required.

Line 98:

This SELECTS output to stream 3 and defines it as a PRINTER stream.

Lines 99-102:

GET ALL 1973 instructs the analysis programmes to read into core the whole of the dump for 1973 and TABULATE ALL requests the fullest set of cross-section tables possible. Lines 101-102 replicate this process for 1979. One could use TABULATE to print results on any particular set of functions or just to produce a table of macro results.

Line 103:

VALUE is a time-series command. For the three variables mentioned it produces a time-series of their value for the whole of the period covered by the dump being used. Tables appear with headings, and with titles as defined in the "read titles" instruction. If none have been defined for a variable that is to be printed, the title space is left blank.

Line 104-105:

This produces a rough line-printer graph of current consumption against current personal disposable income.

Line 106:

FINISH. The end.

THE TEST RUN INPUT

The files required for data input are all contained on the IDIOM tape. They are listed at the start of the Manual as files 12-15, and file 17 of that tape. Their uses are fairly obvious from the descriptions given there and the information given above in this document. One feature requires note, however: the test data are given twice. One version, .FULLDATA, is annotated while the other, .TRUNDATA, has had all explanatory notes removed, ready for input into IDIOM. When the new user is checking through the test model he should use the former, for it is much clearer. Unfortunately - and contrary to the advice given in this handbook - the notes in FULLDATA are not confined to any particular set of columns, so that moving from FULLDATA to TRUNDATA is a very messy process.

Annotation must, however, be removed before an actual run, as otherwise it would be read as data.

The data for the test job contains (at least) one error. The matrix of consumers' expenditure parameters is of dimension (ND,NC,NBDC), with ND=NC=2. Hence, it expects parameters to be offered in the following order for each of the options input:

parameters for flow (1,1)
parameters for flow (2,1)
parameters for flow (1,2)
parameters for flow (2,2)

That is input varies most rapidly over parameters required by the option and then by the first dimension and finally by the second. (Compare input into BXA or BYV.)

Unfortunately, however, the test data appear with the second and third components reversed; i.e. with the second dimension varying more rapidly than the first. For the LES, where there are across-equation constraints - namely that, assuming all commodities use the LES,

$$\left.\begin{array}{l}\sum_J BDC(I,J,1) = 1 \\ \sum_J BDC(I,J,2) = \not p\end{array}\right\} \text{ all I}$$

this reversal leads to nonsense.

THE TEST RUN OUTPUT: TABLE 6

Part 1

The output starts with titling information and then summary statistics of the size of the model. This is followed by the first of the diagnostic prints from FREERD. (Ideally there is no need for these once we are sure of FREERD. It should, therefore, be suppressed.)

Part 2

This is the last of the FREERD information, followed by the start of the DUMP requested at the end of the INPUT segment.

Parts 2-8 (THE DUMP)

This starts with various control data. These are of interest only if one understands the structure of INPUT - especially the parts interpreting directives. If one does, their meanings are clear from the FORTRAN sources.

Following these data come:

	TABLS	Part 2
	TABLE	Part 3
	TABLF	Part 4
	TABLG	Part 5
	JSW	Part 6
	TABLŌ	Part 7
and	ZZ	Part 8

These have been described under "interpreting a dump" on page 126 above. Note that most tables include some spare space, so that additional variables and functions could be incorporated, if necessary. Note also that at the end of ZZ numbers such as 0.515E-84 appear. These are the real representations of the integer values used in the storage of sparse matrices.

Finally in part 8 there is the start of the list of uninitialized variables.

Part 9-10

The list of uninitialized variables is completed and an error message printed out. The severity code is 4 - non-fatal - but if ABORT were set to 1, this would be sufficient to stop IDIOM. One should check the list of uninitialized variables for any important variables, and ultimately attempt to eliminate it.

Following the error message is some more input information. This arises as the exogenous variables PCLO etc. are read. Finally, there is the iteration detail followed by a convergence message. These are, naturally, repeated for every year of the run.

The iteration detail is fairly obvious. Variables are called by their macro-names and the criteria correspond to those declared in the directives. These prints-out can often be useful if the programme crashes, and also if non-convergence appears to arise because of cycling.

The Final Tables

The rest of the specimen output comprises the cross-section tables for 1973 (TABULATE ALL) and the few time-series results requested.

First is the macro-table. It comprises various macro-variables, with growth rates calculated, followed by four sector aggregate results. It is plain from the result that the test run, while being loosely based on the U.K. economy, does not fit very well' This is mainly because of poor parameter estimates rather than any known model defect, although as noted in the introduction to this section, problems do exist.

Next there follow the various cross-section tables for 1973. These are produced by TABULATE ALL. They come in the order of the functions (see Manual, Table 3 or Table 4 above). Table 4 also defines the information generated by each separate TABULATE command. The tables require little comment, being largely self-explanatory. However, we should, perhaps, note the absurdity of the consumption results in which the PDI of the ostensibly

TABLE 6: THE TEST RUN OUTPUT

Part 1

```
**********************************************
** International Dynamic Input-Output Model **
** Program release  3 version  1 : (79320)  **
** Run by AWAF (run  1) at  0  0 :  0  0 56 **
** Idiom version 3 final test run           **
**********************************************

235 symbols read:  172 placed in main symbol table  (126 variables,I3,10H functions 20 classification converters)

 26 symbols read:

 10unit definitions written to file cn stream 16

 4517
47616
 6558
27264
 1518
11442
  776
12156
  133
 1584
  255
 1204
 3586
 7135
    0
 2545
  135
  825
   55
    0
   15
 4032
 4565
 1260
 1265
 8601
   26
 3259
 5058
28152
 1217
11849
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
     22
    327
      2
```

Table 6: The Test Run Output (cont'd)

Part 2

```
24129
10CC0
24129
1.
1.
5CC0
45C00
25C
1200
 NVP:126  NTE2: 17  NTF1: 20  NTF2: 20  NCNV: 26  NFSW: 800  HPOINT: 1391  MXS: 10000
 CIS:  2  CISW:  1  COS:  0  COSW:  0  IABC:  1  MAXE:***  ME4:  0  ME8:  0  ME12:  0  NSW:  8
 FLAG1:F  FLAG2:F  SW:  T  T  T  F  T  F  F  T
```

TABLS

Q	1	4	4	QCO	6	4	4	QGO	11	4	4	QMO	16	4	4	QSO	21	4	4
QVO	26	4	4	QXO	31	4	4	QYO	36	4	4	PQ	41	4	4	PQH	46	4	4
PQHH	51	4	4	PQH1	56	4	4	PQH2	61	4	4	PQM	66	4	4	PQX	71	4	4
QMTO	76	4	4	QMTZ	81	4	4	CMQ	86	4	4	QXTO	91	4	4	OXTZ	96	4	4
Y	101	4	4	YL1	106	4	4	YL2	111	4	4	YL3	116	4	4	YEXP	121	4	4
YEO	126	4	4	YEL1	131	4	4	YH	136	4	4	YHL1	141	4	4	YULC	146	4	4
YF	151	4	4	YV	156	8	4	PVA	165	8	4	PYV1	174	8	4	PYV2	183	8	4
YS	192	4	4	PYS	197	4	4	YSA	202	4	4	PY	207	4	4	PYE	212	4	4
PYVP	217	4	4	YRO	222	4	4	YTO	227	4	4	YTZ	232	4	4	C	237	2	2
PC	240	2	2	CEO	243	2	2	CEOB	246	2	2	CRO	249	2	2	CTO	252	2	2
CTZ	255	2	2	HPD	258	2	2	G	261	1	1	PG	263	1	1	GEO	265	1	1
GEOB	267	1	1	GRO	269	1	1	GTO	271	1	1	GTZ	273	1	1	R	275	5	5
RYO	281	5	5	RCO	287	5	5	RGO	293	5	5	RVO	299	5	5	H	305	4	4
HRR	310	20	4	HFP	331	20	4	S	352	1	1	V	354	2	2	VKO	357	2	2
PV	360	2	2	VRO	363	2	2	VTO	366	2	2	VTZ	369	2	2	XA	372	6	2
PX	379	2	2	PXL1	382	2	2	PXL2	385	2	2	PCLO	388	2	2	PCL1	391	2	2
PCL2	394	2	2	PCL3	397	2	2	DI2	400	2	2	XTO	403	2	2	XTO1	406	2	2
XTO2	409	2	2	M	412	2	2	FFM	415	2	2	PM	418	2	2	E	421	1	1
UE	423	1	1	LF	425	1	1	PE	427	1	1	PEL1	429	1	1	EYO	431	1	1
ECO	433	1	1	EGO	435	1	1	D	437	2	2	DN	440	2	2	DBT	443	2	2
DAB	446	10	2	DDB	457	10	2	DC	468	4	2	DPOP	473	2	2	DPDI	476	2	2
DSC	479	2	2	DSPC	482	2	2	DPCE	485	2	2	DPW	488	2	2	DPWH	491	2	2
T	494	3	3	TO	498	33	3	TCMO	532	3	3	TQXO	536	3	3	TYO	540	3	3
TCO	544	3	3	TGO	548	3	3	TVO	552	3	3	PWLO	556	3	3	PWL1	560	3	3
PWL2	564	3	3	PWL3	568	3	3	AD1	572	3	3	AD2	576	3	3	DI1	580	3	3

Table 6: The Test Run Output (cont'd)

EXA	868	198-10	BPX	1067	32-10	BPC	1100	1-12	BT	1102	0-14	RV	1103	18 -8
BULC	1122	0 -2	BQ	1123	0 -1	BUE	1124	0-12	BHR	1125	0 -6	BH	1126	0 -[illegible]
ECYC	1127	0 -1	TCMB	1341	3-14	TYB	1345	9-14	TCB	1355	12-14	TGB	1368	3-14
TVB	1372	3-14	TQXB	1376	0-14	RYB	1128	20 -5	RCB	1149	10 -5	RGB	1160	5 -5
RVB	1390	0 -5	QYC	1166	16 -1	MQC	1183	8-11	YQC	1377	12 -2	QCC	1192	8 -1
QGC	1201	4 -1	QSC	1206	4 -1	CVC	1211	8 -1	QXC	1220	8 -1	VKC	1229	2 -8
QYPC	1232	16 -1	EYEC	1249	0-12	ECEC	1250	0-12	EGEC	1251	0-12	HRC	1252	20 -6
DRC	1273	10-13	QYSC	1284	0 -1	Z1	1285	11 0	Z2	1297	11 0	Z3	1309	11 0
Z4	1321	11 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0
	0	0 0		0	0 0		0	0 0		0	0 0		0	0 0

TABLE — Part 3

Q	NQ	4	282	Y	NY	4	256	C	NC	2	260	G	NG	1	262	R	NR	6	263
H	NH	4	268	S	NS	1	272	V	NV	2	273	F	NF	1	275	X	NX	2	276
M	NM	2	278	E	NE	1	280	D	ND	2	281	T	NT	3	283	A	NA	3	286
K	NK	1	289	O	NO	11	0												

TABLF — Part 4

CMO	4	8	8	LLIN	8	FIXQ	1	FIXM	1	CMC	6		0		0		0	NULL	0
FCM	4	6	9	LLIN	6	CM	1	FIX	1	PFM	1		0		0		0	NULL	0
PY	4	7	33	PQMF	0	LMAT	7	LKEY	12	LLM	7	LLK	12	PVA	8	PRS	6	NULL	0
YV	4002	-19	21	NCL	19	NCLP	19	ACC	7	FIX	1	PROF	6		0		0	NULL	8
YEO	4	14	13	RECU	14	LLIN	8	LLWA	9	EFIX	1	PFIX	1		0		0	NULL	0

Table 6: The Test Run Output (cont'd)

```
R        5    0   37 WAGE    0 PROF    0 ITAX    0 VOLS    0 FIX     0 IMP     0 GOOD    0         0
XA    2003   33   68 LIN    33 LLIN   33         0         0         0         0         0 NULL    0
PX       2   16   74 LPH    16 LUM    16 LUK    16 LLPH   16 LLUM   16 LLUK   16 PQHH    0 PC      1
PE       1    1   77 EXGI    1 EXGW    1 PHIL   10 SARG   12         0         0         0 NULL    0
T        3    0   24 ADVA    0 SPEC    0 SUBS    0 EMPW    0 EMPE    0 PRPP    0 PRP     0 ADVI    0
V        2    9   66 INDL    0 PLAN    0 DWEL    8 LFEE    9         0         0         0 NULL    0
YULC     4    0   17 ACTU    0 STAN    0         0         0         0         0         0 NULL    0
C        4    0    1 IDEN    0 DIFF    1         0         0         0         0         0         0
UE       1    0   76 SIMP    0 VARP    7         0         0         0         0         0         0
HRP   4005    0   48 DIST    7 INTE   12 GRAN    6 FIX     6 SPEC    8 NULL    0 PPA     8         0
H        4    0   42 HOUS    0 NPMB    0 LASS    0 CORP    0 NATI    0 CGOV    0 LGOV    0 ROW     0
QYC   4004    0   28 NULL    0 CDPF    1 CES     2 LTIM    4 LGDP    4         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0
        -1   -1   -1 LES     5 LLIN    6 FIX     7         0         0         0         0         0

TABLG
TQMB  1  3  4   3  TYB   1  3  4   9  TCB   1  3  2  12  TGB   1  3  1   3  TVB   1  3  1   3  TQXB  1  3  4   0     Part 5
RYB   2  5  4  20  RCB   2  5  2  10  RGB   2  5  1   5  RVB   1  5  2   0  QYC   2  4  4  16  MQC   2  2  4   8
YCC   1  4  4  12  CCC   2  4  2   8  QGC   2  4  1   4  QSC   2  4  1   4  QVC   2  4  2   8  QXC   2  4  2   8
VKC   2  2  1   2  QYPC  2  4  4  16  EYEC  4  1  4   0  ECEC  4  1  2   0  EGEC  4  1  1   0  HRC   2  4  5  20
DFC   2  2  5  10  QYSC  3  4  4   0       -1 -1 -1  -1       -1 -1 -1  -1       -1 -1 -1  -1       -1 -1 -1  -1

JSW
 1 1 1 1 2 1 3 3 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 4 4 4 4 3 3 3 3 2 2 2 2 1 3 2 6 7 1 4 6 8 6 6 6 6 6     Part 6
 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 1 4 2 2 2 2 2 2 5 5 1 2 1 1 1 1 2 2 1 2 3 0 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Part 7

TABLO

DFE	50000.	SYVA	-0.21541E-26	SCVA	-0.21541E-26	SGVA	-0.21541E-26	SYP	100.00
SCM	500.00	SGM	100.00	SPYM	100.00	SPCM	500.00	SPGM	100.00
SP	10887.	GDP	50735.	SC	31472.	SG	9095.0	SX	11275.
SR	-0.21541E-26	FPDI	34622.	SS	460.00	SV	9380.0	WVCL	100.00
SPM	10887.	GDPC	-0.21541E-26	SPC	31472.	SPG	-0.21541E-26	SPX	10000.
SCM	-0.21541E-26	PDI	34622.	SPS	-0.21541E-26	SPV	-0.21541E-26	PPI	1.0000
PEM	1.0000	HUC	1.0000	FCE	1.0000	PSG	1.0000	PEX	1.0000
SFCP	-0.21541E-26	37	-0.21541E-26	PSS	1.0000	PSV	1.0000	SEA	-0.21541E-26
PSCP	-0.21541E-26	PSYM	1.0000	PSCM	1.0000	PSGM	1.0000	FEXM	-0.21541E-26
PCP	55.765	WPOP	26000.	SYS	-0.21541E-26	POPL	-0.21541E-26	SGS	-0.21541E-26
AW	1300.0	AWY	1300.0	AWC	1300.0	AWG	1300.0	55	-0.21541E-26
EPPL	24739.	UNEM	500.00	EMFY	20274.	EMPC	100.00	EMFG	4365.0
RVAT	-0.21541E-26	62	-0.21541E-26	SFT	-0.21541E-26	RET	0.30000	EX	1.0000
FINT	0.10000E 00	RMOR	0.10000E	COSPYS	-0.21541E-26	GDPT	-0.21541E-26	EPGS	-0.21541E-26
BT	-0.21541E-26	BP	-0.21541E-26	PSER	-0.21541E-26	PSFD	-0.21541E-26	EXL1	1.0000
FWPH	1000.0	SRAT	-0.21541E-26	SYSA	-0.21541E-26	RETL	0 30000	SGSA	-0.21541E-26
SY	-0.21541E-26	SVVA	-0.21541E-26	SPVM	-0.21541E-26	SVM	-0.21541E-26	EXL2	1.0000
XFT	-0.21541E-26	EPDI	-0.21541E-26	XPFT	-0.21541E-26	SYV	-0.21541E-26	PUF	1.0000
SPY	-0.21541E-26	WAGY	26000.	WAGC	130.00	WAGG	6000.0	WAGE	32130.

106	-0.21541E-26	107	-0.21541E-26	108	-0.21541E-26	109	-0.21541E-26	110	-0.21541E-26
GDPL	-0.21541E-26	SCL	-0.21541E-26	SGL	-0.21541E-26	SVL	-0.21541E-26	SSL	-0.21541E-26
SXL	-0.21541E-26	SML	-0.21541E-26	EML1	-0.21541E-26	UNL1	-0.21541E-26	PDIL	31612.
HUCL	1.0000	PCEL	1.0000	PSGL	1.0000	PSVL	1.0000	PSSL	1.0000
FSXL	1.0000	FSML	1.0000	AWL	-0.21541E-26	RDI1	567.00	FDI2	533.00
GDPB	50785.	SCB	31472.	SGB	9095.0	SVB	9380.0	SSB	450.00
SXB	11275.	SMB	10887.	EME	24739.	139	-0.21541E-26	140	-0.21541E-26
141	-0.21541E-26	142	-0.21541E-26	143	-0.21541E-26	144	-0.21541E-26	145	-0.21541E-26
STAR	1973	FINI	**********	YEAR	**********	DATE	**********	150	**********
PEAS	1970	QBAS	1970	YA70	1	YA72	1	155	**********
156	**********	157	**********	158	**********	159	**********	160	**********

ZZ

1	0.492E 04	0.476E 05	0.697E 04	0.273E 05	0.927E-76	0.192E 04	0.114E 05	776.	0.122E 05	0.927E-76
11	133.	0.198E 04	285.	0.120E 04	0.927E-76	0.359E 04	0.714E 04	0.0	0.255E 04	0.927E-76
21	135.	825.	99.0	0.0	0.927E-76	15.0	0.403E 04	0.457E 04	0.126E 04	0.927E-76
31	0.127E 04	0.860E 04	26.0	0.330E 04	0.927E-76	0.506E 04	0.282E 05	0.122E 04	0.118E 05	0.927E-76
41	1.00	1.00	1.00	1.00	0.927E-76	1.00	1.00	1.00	1.00	0.927E-76
51	1.00	1.00	1.00	1.00	0.927E-76	1.00	1.00	1.00	1.00	0.927E-76
61	1.00	1.00	1.00	1.00	0.927E-76	1.00	1.00	1.00	1.00	0.927E-76
71	1.00	1.00	1.00	1.00	0.927E-76	22.0	327.	0.0	0.0	0.927E-76
81	0.100E 00	0.200	0.0	0.0	0.927E-76	0.0	0.0	0.0	0.0	0.927E-76
91	0.0	0.0	0.0	0.0	0.927E-76	0.0	0.0	0.0	0.0	0.927E-76
1	0.470E 04	0.493E 05	0.673E 04	0.260E 05	0.927E-76	0.462E 04	0.490E 05	0.665E 04	0.255E 05	0.927E-76
11	0.454E 04	0.485E 05	0.660E 04	0.250E 05	0.927E-76	0.450E 04	0.480E 05	0.650E 04	0.245E 05	0.927E-76
21	0.470E 04	0.490E 05	0.675E 04	0.260E 05	0.927E-76	0.114E 04	0.826E 04	0.179E 04	0.948E 04	0.927E-76
31	0.115E 04	0.825E 04	0.185E 04	0.930E 04	0.927E-76	40.0	42.0	44.0	40.0	0.927E-76
41	40.5	43.0	46.0	40.0	0.927E-76	0.400	0.700	0.600	0.500	0.927E-76
51	0.0	0.0	0.0	0.0	0.927E-76	529.	0.252E 04	176.	0.330E 04	0.0
61	0.0	0.0	30.0	0.927E-76	0.200	0.0	0.200	0.0	0.200	0.0
71	0.100E 00	0.0	0.927E-76	0.800	1.00	0.800	1.00	0.800	1.00	0.900
81	1.00	0.927E-76	0.800	1.00	0.800	1.00	0.800	1.00	0.900	1.00
91	0.927E-76	100.	1000.	-150.	0.0	0.927E-76	1.00	1.00	1.00	1.00
1	0.927E-76	0.0	0.0	0.0	0.0	0.927E-76	1.00	1.00	1.00	1.00
11	0.927E-76	0.110E 04	0.173E 04	0.142E 04	0.141E 04	0.927E-76	1.00	1.00	1.00	1.00
21	0.927E-76	0.201E 04	0.174E 05	0.216E 04	0.179E 05	0.927E-76	-400.	700.	653.	0.120E 04
31	0.927E-76	-0.100E 00	0.100E 00	0.100E 00	0.100E 00	0.927E-76	0.341E 05	1000.	0.927E-76	1.00
41	1.00	0.927E-76	0.0	0.0	0.927E-76	0.100E-03	0.0	0.927E-76	0.790E 04	900.
51	0.927E-76	0.541E 04	0.0	0.927E-76	0.150	0.0	0.927E-76	0.100E-01	0.0	0.927E-76
61	0.957E 04	0.927E-76	1.00	0.927E-76	0.436E 04	0.927E-76	0.450	0.927E-76	0.636E 04	0.927E-76
71	575.	0.927E-76	0.500E-01	0.927E-76	0.480E 05	0.100E 05	0.260E 04	0.125E 04	0.0	0.927E-76
81	0.370E 05	0.460E 04	0.170E 04	150.	0.0	0.927E-76	0.0	0.541E 04	0.317E 04	500.
91	0.0	0.927E-76	0.558E 04	575.	0.0	202.	0.0	0.927E-76	0.0	0.0
1	0.0	0.0	0.0	0.927E-76	0.0	0.0	0.0	0.0	0.927E-76	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.927E-76
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	0.927E-76	1000.	0.927E-76	0.100E 05	30.0	0.927E-76	0.350E 04	0.0	0.927E-76	1.00
61	1.00	0.927E-76	0.0	0.0	0.927E-76	0.0	0.0	0.927E-76	0.0	0.0
71	0.927E-76	0.200E 04	500.	0.300E 04	1000.	0.300E 04	500.	0.927E-76	1.00	1.00
81	0.927E-76	1.00	1.00	0.927E-76	1.00	1.00	0.927E-76	1.00	1.00	0.927E-76

Part 8

Table 6: The Test Run Output (cont'd)

Table 6: The Test Run Output (cont'd)

11	-0.215E-26	1.00	-0.215E-26	0.0	-0.215E-26	0.0	-0.215E-26	0.0	-0.215E-26	0.0
21	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.700	0.0	1.00
31	0.0	0.0	1.10	0.0	1.00	0.0	0.0	1.00	0.0	1.00
41	0.0	0.0	0.800	0.0	1.00	0.500E-01	0.0	0.927E-76	0.0	0.0
51	1.00	0.500E-01	0.0	0.0	0.0	1.00	1.00	0.0	0.927E-76	1.00
61	0.0	1.00	0.0	0.0	0.927E-76	0.200E-01	0.160	0.200E-01	0.900E-01	0.150
71	0.460	0.500E-01	0.800E-01	0.200E-01	0.260	0.100E 00	0.800E-01	0.300E-02	0.150	0.200E-01
81	0.800E-01	0.927E-76	1.00	0.0	1.00	0.0	1.00	0.0	0.0	1.00
91	0.927E-76	0.500E-01	0.350	0.200E-01	0.400	0.0	0.0	0.0	0.0	0.927E-76
1	0.100E-01	0.200	0.300E-01	0.180	0.927E-76	0.130	0.820	0.500E-01	0.0	0.927E-76
11	0.100E-02	0.400	0.450	0.149	0.0	0.0	0.0	1.00	0.927E-76	0.120
21	0.270	0.100E-01	0.0	0.0	0.0	0.0	1.00	0.927E-76	1.00	0.0
31	0.927E-76	0.0	1.00	0.0	0.0	0.0	1.00	0.0	0.0	0.0
41	1.00	0.0	0.0	0.0	1.00	0.0	0.0	0.927E-76	0.927E-76	0.927E-76
51	0.927E-76	1.00	0.0	0.0	0.0	0.0	0.0	1.00	0.0	0.200
61	0.300	0.0	0.0	0.0	0.0	0.0	1.00	0.0	0.0	0.0
71	0.0	0.927E-76	0.600	0.400	0.200	0.800	0.800	0.200	0.0	0.0
81	0.0	0.0	0.927E-76	0.927E-76	0.0	0.0	0.0	0.0	0.0	0.0
91	0.0	0.0	0.0	0.0	0.0	0.927E-76	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.927E-76	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.927E-76
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
31	0.0	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.927E-76	0.927E-76
41	0.515E-84	0.515E-84	0.100E 00	0.927E-76	0.515E-84	0.103E-83	0.200	0.103E-83	0.103E-83	0.500E-01
51	0.154E-83	0.515E-84	-400.	0.927E-76	0.515E-84	0.515E-84	0.100E 00	0.103E-83	0.515E-84	0.100E 00
61	0.154E-83	0.515E-84	-100.	0.515E-84	0.103E-83	0.100E 00	0.927E-76	0.515E-84	0.515E-84	0.100E 00
71	0.927E-76	0.515E-84	0.515E-84	0.100E 00	0.927E-76	0.927E-76	0.515E-84	0.515E-84	1.00	0.103E-83
81	0.103E-83	1.00	0.154E-83	0.154E-83	1.00	0.206E-83	0.206E-83	1.00	0.927E-76	0.927E-76
91	0.0									

Part or all of the following variables are unset

```
Symbol  SYVA
Symbol  SCVA
Symbol  SGVA
Symbol  SR
Symbol  GDPC
Symbol  SPG
Symbol  SCM
Symbol  SFS
Symbol  SPV
Symbol  SFCM
Symbol  37
Symbol  SSA
Symbol  PSCM
Symbol  PSXM
Symbol  SYS
Symbol  POPL
Symbol  SGS
Symbol  SS
Symbol  RVAT
Symbol  62
Symbol  SRT
Symbol  SFYS
Symbol  GDPT
Symbol  SPGS
Symbol  BT
Symbol  BP
Symbol  PSBR
Symbol  PSFD
Symbol  SRAT
Symbol  SYSA
Symbol  SGSA
Symbol  SY
Symbol  SVVA
```

Table 6: The Test Run Output (cont'd)

```
Symbol  XPFT
Symbol  SYV
Symbol  SPY
Symbol  RESE
Symbol  101
Symbol  102
Symbol  103
Symbol  104
Symbol  105
Symbol  106
Symbol  107
Symbol  108
Symbol  109
Symbol  110
Symbol  GDPL
Symbol  SCL
Symbol  SGL
Symbol  SVL
Symbol  SSL
Symbol  SXL
Symbol  SML
Symbol  EML1
Symbol  UNL1
Symbol  AWL
Symbol  139
Symbol  140
Symbol  141
Symbol  142
Symbol  143
Symbol  144
Symbol  145
Symbol  FINI
Symbol  YEAR
Symbol  DATE
Symbol  150
Symbol  155
Symbol  156
Symbol  157
Symbol  158
Symbol  159
Symbol  160

 ** Variables not initialised

 ** Error code 2002   Severity code   4

 ** Total cards read   82     Last card processed was number   82

 ** Card is :

    SCLVE

 ** Commands processed  141

 ** Current command in command format (error in column    6) :

    SCLVE <
```

Part 9

Table 6: The Test Run Output (cont'd)

Part 10

```
IDPUYS  .3
1. 1.
1. 1. 1.
1.1 1.1 1.1
1.1 1.1 1.1
1. 1. 1.
1. 1.
Convergence test on iteration  1 for Y   : Sum of squares = 0.2042383E 09 max absolute change =  10527.68
GDP  44544. SC  27398. SV  12504. SX   7242. SM  12891. PDI  25964. WAG  30704. BT  -5649. EM  25686. HUC  1.152

Convergence test on iteration  2 for Y   : Sum of squares = 0.6956342E 08 max absolute change =  7327.406
GDP  22277. SC  21887. SV   9153. SX   7204. SM  16259. PDI  29884. WAG  32137. BT  -7443. EM  23083. HUC  1.174

Convergence test on iteration  3 for Y   : Sum of squares = 0.1812501E 08 max absolute change =  2840.996
GDP  22538. SC  24856. SV   6630. SX   6043. SM  27281. PDI  25571. WAG  32763. BT -19511. EM  20047. HUC  1.129

Convergence test on iteration  4 for Y   : Sum of squares =  1806307.      max absolute change =  886.7734
GDP  24726. SC  21655. SV   5327. SX   6333. SM  20880. PDI  24067. WAG  29416. BT -12128. EM  18532. HUC  1.126

Convergence test on iteration  5 for Y   : Sum of squares =  883186.6      max absolute change =  684.1836
GDP  27249. SC  20742. SV   5162. SX   6359. SM  17305. PDI  21429. WAG  25491. BT  -8619. EM  18358. HUC  1.087

Convergence test on iteration  6 for Y   : Sum of squares =  184567.2      max absolute change =  354.6859
GDP  24330. SC  19191. SV   4970. SX   6458. SM  18579. PDI  20410. WAG  23486. BT  -9847. EM  18037. HUC  1.053

Convergence test on iteration  7 for Y   : Sum of squares =  241697.0      max absolute change =  365.1055
GDP  25718. SC  18603. SV   4890. SX   6529. SM  16595. PDI  20340. WAG  23260. BT  -7799. EM  17892. HUC  1.032

Convergence test on iteration  8 for Y   : Sum of squares =  88957.56      max absolute change =  249.3750
GDP  26641. SC  18701. SV   4815. SX   6572. SM  15738. PDI  20218. WAG  22846. BT  -6716. EM  17715. HUC  1.024

Convergence test on iteration  9 for Y   : Sum of squares =  8565.473      max absolute change =  75.44141
GDP  27599. SC  18675. SV   4763. SX   6607. SM  14736. PDI  20162. WAG  22662. BT  -5665. EM  17614. HUC  1.018

Convergence test on iteration 10 for Y   : Sum of squares =  879.0701      max absolute change =  18.76172
GDP  28016. SC  18687. SV   4747. SX   6611. SM  14320. PDI  20072. WAG  22435. BT  -5208. EM  17597. HUC  1.014

Convergence test on iteration 11 for Y   : Sum of squares =  278.8511      max absolute change =  12.30078
GDP  27998. SC  18676. SV   4744. SX   6615. SM  14327. PDI  20026. WAG  22302. BT  -5205. EM  17593. HUC  1.010

Convergence test on iteration 12 for Y   : Sum of squares =  160.6572      max absolute change =  10.03516
GDP  28041. SC  18678. SV   4746. SX   6615. SM  14290. PDI  20029. WAG  22279. BT  -5158. EM  17595. HUC  1.007

Convergence test on iteration 13 for Y   : Sum of squares =  117.6953      max absolute change =  8.117188
GDP  28086. SC  18706. SV   4748. SX   6614. SM  14274. PDI  20042. WAG  22273. BT  -5123. EM  17604. HUC  1.006

Convergence test on iteration 14 for Y   : Sum of squares =  201.7256      max absolute change =  9.968750
GDP  28129. SC  18731. SV   4750. SX   6614. SM  14256. PDI  20056. WAG  22281. BT  -5096. EM  17608. HUC  1.005

Convergence test on iteration 15 for Y   : Sum of squares =  161.4170      max absolute change =  9.554688
GDP  28155. SC  18750. SV   4752. SX   6612. SM  14251. PDI  20066. WAG  22287. BT  -5084. EM  17613. HUC  1.005

Convergence test on iteration 16 for Y   : Sum of squares =  113.6610      max absolute change =  8.187500
GDP  28157. SC  18764. SV   4754. SX   6611. SM  14263. PDI  20073. WAG  22292. BT  -5093. EM  17617. HUC  1.005

Convergence test on iteration 17 for Y   : Sum of squares =  63.00650      max absolute change =  6.226563
GDP  28154. SC  18773. SV   4756. SX   6610. SM  14276. PDI  20079. WAG  22298. BT  -5104. EM  17621. HUC  1.004

Convergence test on iteration 18 for Y   : Sum of squares =  30.07323      max absolute change =  4.332031
GDP  28151. SC  18779. SV   4757. SX   6609. SM  14286. PDI  20084. WAG  22304. BT  -5113. EM  17624. HUC  1.004
```

Table 6: The Test Run Output (cont'd)

```
Convergence test on iteration  15 for Y    : Sum of squares =  2.501175     max absolute change =  1.226563
GDP  35836. SC  20273. SV  11281. SX   9169. SM  17178. PDI  20601. WAG  22890. BT  -7486. EM  17586. HUC  0.984

                                   *** Converged after  15 iterations in 1978 ***

 ** Warning : DUMP selected for OUTPUT is not empty
*** ITERATIONS DONE AND RESULTS DUMPED  ***
IDMUYS   3
IDPUYS   3
IDMLYS   3
IDPUYS   3
1. 1.
1. 1. 1.
1.7 1.7 1.7
1.7 1.7 1.7
1. 1. 1.
1. 1.
Convergence test on iteration   1 for Y    : Sum of squares =  610483.1     max absolute change =  615.2070
GDP  35263. SC  20373. SV  12030. SX   9746. SM  17177. PDI  20601. WAG  22893. BT  -6900. EM  17507. HUC  0.985

Convergence test on iteration   2 for Y    : Sum of squares =  500944.1     max absolute change =  648.4766
GDP  35637. SC  20377. SV  12286. SX   9961. SM  17279. PDI  20954. WAG  22797. BT  -7455. EM  17769. HUC  0.962

Convergence test on iteration   3 for Y    : Sum of squares =  207528.5     max absolute change =  411.2070
GDP  35029. SC  20798. SV  12486. SX   9924. SM  17471. PDI  20360. WAG  22707. BT  -7720. EM  17976. HUC  0.950

Convergence test on iteration   4 for Y    : Sum of squares =  85360.06     max absolute change =  213.6797
GDP  35416. SC  20360. SV  12609. SX   9895. SM  17739. PDI  20613. WAG  23054. BT  -7945. EM  18096. HUC  0.948

Convergence test on iteration   5 for Y    : Sum of squares =  9306.828     max absolute change =  57.46484
GDP  35591. SC  20573. SV  12700. SX   9865. SM  17837. PDI  20815. WAG  23330. BT  -7991. EM  18180. HUC  0.950

Convergence test on iteration   6 for Y    : Sum of squares =  1839.458     max absolute change =  41.09766
GDP  35520. SC  20732. SV  12688. SX   9864. SM  18055. PDI  20943. WAG  23492. BT  -8175. EM  18173. HUC  0.952

Convergence test on iteration   7 for Y    : Sum of squares =  1524.295     max absolute change =  30.42969
GDP  35804. SC  20807. SV  12695. SX   9861. SM  17850. PDI  21011. WAG  23606. BT  -7976. EM  18176. HUC  0.954

Convergence test on iteration   8 for Y    : Sum of squares =  1003.101     max absolute change =  24.64844
GDP  35737. SC  20849. SV  12707. SX   9856. SM  17966. PDI  21005. WAG  23596. BT  -8098. EM  18187. HUC  0.955

Convergence test on iteration   9 for Y    : Sum of squares =  640.3406     max absolute change =  20.16016
GDP  35632. SC  20839. SV  12717. SX   9853. SM  18068. PDI  21001. WAG  23600. BT  -8215. EM  18196. HUC  0.955

Convergence test on iteration  10 for Y    : Sum of squares =  81.91780     max absolute change =  7.031250
GDP  35579. SC  20831. SV  12725. SX   9851. SM  18119. PDI  21005. WAG  23614. BT  -8272. EM  18204. HUC  0.955

Convergence test on iteration  11 for Y    : Sum of squares =  6.226120     max absolute change =  1.683594
GDP  35554. SC  20832. SV  12727. SX   9850. SM  18147. PDI  21012. WAG  23626. BT  -8301. EM  18206. HUC  0.956

                                   *** Converged after  11 iterations in 1979 ***

 ** Warning : DUMP selected for OUTPUT is not empty
*** ITERATIONS DONE AND RESULTS DUMPED  ***
```

Table 6: The Test Run Output (cont'd)

Idiox version 3 final test run — 1973 Use of resources and condensed tables — AWAP 1 0 0: 0 0 56

	GDP	C'EXP	GOV'T	INV'T	STKB'G	EXPORTS	IMPORTS
1970 prices	28147	18788	9666	4759	328	8608	14299
Current prices	28268	17522	10914	4672	288	8966	14090
Price indices	1.004	0.933	1.098	0.982	0.876	1.042	0.985
Inflation rates	0.43	-6.74	9.51	-1.83	-12.36	4.16	-1.46
% growth volume	-44.58	-40.30	9.58	-49.27	-27.78	-23.68	31.34
Growth from 1973	-44.58	-40.30	9.58	-49.27	-27.78	-23.68	31.34

TRADE		EMPLOYMENT	
Balance of trade	-5128	Employment	17627
Export index	104.2	Unemployment	8401
Terms of trade	105.7	Unem rate	32.28
% change in tt	5.70	Growth emplt	-28.75

Part 1	Consum -ption	Govern -ment	Invest -ment	Stock -bd'ng	Final demand	Interm demand	Home demand	Export demand	Total demand	Import supply	Import duties	Total output
1. Primary	501	133	8	20	658	4183	4841	730	5571	610	61	4901
2. Manufacturing	3505	1984	1857	200	7546	15683	23229	5292	28520	1754	351	26416
3. Construction	200	285	2089	5	2580	2084	4664	61	4724	150	0	4574
4. Services	4005	1204	808	100	6117	3752	9869	2525	12394	2000	0	10394
Total commodities	8211	3606	4759	328	16901	25701	42602	8608	51210	4513	412	46285
Direct imports	9269	0	0	0	9269	0	9269	0	9269	9269	0	0
Indirect taxes	876	1091	456	0	2422	0	2422	0	2422	0	0	2422
Other value-added	0	6054	0	0	6054	0	6054	0	6054	0	0	6054
Foreign tourists	0	0	0	0	0	0	0	0	0	0	0	0
Grand total	18355	10782	5218	328	34646	25701	60347	8608	68955	13782	412	54761

Part 2	Total supply
1. Primary	5571
2. Manufacturing	28520
3. Construction	4724
4. Services	12394
Total commodities	51210
Direct imports	9269
Indirect taxes	2422
Other value-added	6054
Foreign tourists	0
Grand total	68955

Table 6: The Test Run Output (cont'd)

Idios version 3 final test run — 1973 Commodity imports and import ratios — AWAP 1 0 0: 0 0 56

	Import volume	Duties (base)	Output volume	Import ratio	Quota rates	Import value	Import price	Prodn price
1. Primaries	610	61	4901	12.78	0.0	610	1.000	0.684
2. Manufactures	1784	351	26416	7.68	0.0	1572	0.897	0.886
3. Construction	150	0	4674	3.22	0.0	141	0.940	0.883
4. Services	2000	0	10394	20.27	0.0	1982	0.991	0.991
Total	4513	412	46288	10.70	0.0	4304	0.954	0.885

Idios version 3 final test run — 1973 World, import and domestic prices — AWAP 1 0 0: 0 0 56

	World price	Import price	Prodn price	Absprn price	Tariff rate	Import volume
1. Primaries	1.000	1.000	0.654	0.641	0.100	610
2. Manufactures	1.000	0.857	0.886	0.853	0.0	1754
3. Construction	1.000	0.940	0.883	0.883	0.0	150
4. Services	1.000	0.991	0.991	0.970	0.0	2000
Total	1.000	0.954	0.885	0.860	0.014	4513

Idios version 3 final test run — 1973 Commodity, import and export prices — AWAP 1 0 0: 0 0 56

	Prod'n prices	Home prices	Import prices	Export prices	Absp'n prices
1. Primaries	0.654	0.588	1.000	1.029	0.641
2. Manufactures	0.886	0.849	0.897	1.029	0.853
3. Construction	0.883	0.881	0.940	1.029	0.883
4. Services	0.991	0.965	0.991	1.072	0.970

Idios version 3 final test run	1973 Industrial output and investment									AWAP 1 0 0: 0 0 56
	Output	% change	Price	Output value	Profit	Fixed inv	Legal fees	Total	Total value	Av. price
1. Primary	4901	6.072	0.654	3203	1967	386	0	386	379	0.982
2. Manufacturing	26416	-46.090	0.874	23089	-4195	543	0	543	533	0.982
3. Construction	4574	-31.213	0.883	4039	481	212	0	212	209	0.982
4. Services	10394	-59.239	0.991	10298	1148	1	0	1	1	0.982
Total	46285	-46.036	0.878	40630	-600	1143	0	1143	1122	0.982

Idios version 3 final test run	1973 Industrial output, employment & wages								AWAP 1 0 0: 0 0 56
	Output	% change	Invest -ment	Employ -ment	Hours	Wage rates	% g r Men	% g r Hours	% g r Prodty
1. Primary	4901	6.07	386	1212	39.1	1095	5.41	1.66	4.34
2. Manufacturing	26416	-46.09	543	4609	40.0	1730	-44.40	-48.22	4.11
3. Construction	4574	-31.21	212	1357	42.8	1424	-26.63	-32.26	1.88
4. Services	10394	-59.24	1	5963	40.0	1413	-35.88	-35.88	-36.43
Total	46288	-46.04	1143	13142.	40.2	1496	-36.17	-28.60	-12.10

Idios version 3 final test run	1973 Industrial stockbuilding & stock appn					AWAP 1 0 0: 0 0 86
	Output	Change	% change	Con pr stkbdg	Cur pr stkbdg	Val of stocks
1. Primary	4901	281	6.07	20	13	2100
2. Manufacturing	26416	-22584	-46.09	200	171	11000
3. Construction	4574	-2076	-31.21	5	4	900
4. Services	10394	-15106	-59.24	100	97	5000
Total	46285	-39485	-46.04	325	285	19000

Table 6: The Test Run Output (cont'd)

Table 6: The Test Run Output (cont'd)

Idics version 3 final test run — 1973 Consumers' expenditure cons & curr price — AWAP 1 0 0: 0 0 56

Part 1	Commod -ities	Import	Other val.ad	Tax (base)	Total volume	Import	Value added	Taxes	Advel -orem	Spec -ific	Subs -idies	Total value
1. Domestic	8211	501	-200	1502	10014	501	0	875	875	1001	-100	8748
2. Exp. abroad	0	8771	0	0	8771	8771	0	0	877	0	0	8771
Total	8211	9271	-200	1502	18784	9271	0	875	1752	1001	-100	17519

Part 2	Price
1. Domestic	0.874
2. Exp. abroad	1.000
Total	0.933

	< tax limit	> tax limit
1. Domestic	5482	4532
2. Exp. abroad	4799	3972
Total	10280	8504

Idics version 3 final test run — 1973 Aggregate consumption, disposable income — AWAP 1 0 0: 0 0 56

	Cons volume	Cons value	Cons price	Percap volume	Percap value	No of consum	Percap PDI	Percap RPDI	Total wealth	Percap wealth
1. < tax limit	10282	9590	0.933	514	479	20000	614	658	5000	250
2. > tax limit	8505	7932	0.933	238	222	35765	219	234	43411	1214
Total	18788	17522	0.933	337	314	55765	360	386	48411	868

	Cons volume	Cons value	Cons price	Curr PDI	Real PDI	Exp'td RPDI	% g r PDI	% g r RPDI	% g r Cons	Curr wealth
All consumers	18788	17522	0.933	20090	21541	23097	-41.97	-37.76	-40.30	48411

Idios version 3 final test run	1973 Exports by commodity and area	AWAP 1 0 0: 0 0 56

	N. America	Europe	R.O.W.	Total	Price
1. Visibles	1111	2733	2238	6082	1.029
2. Invisibles	754	1018	754	2526	1.072
Total	1866	3751	2992	8608	1.042

Idios version 3 final test run	1973 Export prices & competitors' prices	AWAP 1 0 0: 0 0 56

	Export prices	Lagged 1 year	Lagged 2 year	Export taxes	Compet prices	Lagged 1 year	Lagged 2 year	Lagged 3 year	Export volume
1. Visibles	1.029	1.000	1.000	0.0	1.000	1.000	1.000	1.000	6082
2. Invisibles	1.072	1.000	1.000	0.0	1.000	1.000	1.000	1.000	2526
Total	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8608

Idios version 3 final test run	1973 Wage rates and inflation	AWAP 1 0 0: 0 0 56

	Av'ge wage	Lagged 1 year	Money change	% change	Total emp't	Total unem't	Unem't rate	% g r prices	'% g r ret'ns
Total	1350	900	450	50.00	17627	8401	0.32	-6.74	0.0

Idios version 3 final test run	1973 Indirect taxes by expenditure category	AWAP 1 0 0: 0 0 56

	Commod -ities	Indus -tries	Consum -ption	Govern -ment	Rec -eipts	Inst'l sector	Stock -bld'g	Invest -ment	Financ assets	Export	Import	Total
1. Ad val.	0	4617	875	1091	0	0	0	456	0	0	61	7100
2. Specific	0	1321	0	0	0	0	0	0	0	0	0	1321
3. Subsidies	0	-400	0	0	0	0	0	0	0	0	0	-400
Total	0	5538	875	1091	0	0	0	456	0	0	61	0

Table 6: The Test Run Output (cont'd)

Idicm version 3 final test run — 1973 Aggregate investment by asset — AWAP 1 0 0: 0 0 56

	Social (vol)	Ind'l (vol)	Total (vol)	Social (val)	Ind'l (val)	Total (val)	Asset price	Total tax	Tax rate
1. Fixed inv	3500	1143	4643	3437	1122	4559	0.982	456	0.100
2. Legal fees	0	116	116	0	112	112	0.970	0	0.0
Total	3500	1259	4759	3437	1235	4672	0.982	456	0.098

Idiom version 3 final test run — 1973 Industrial wages,profits and taxes — AWAP 1 0 0: 0 0 56

Part 1	Output value	Mats input	Wage bill	Profit	Direct import	Other inputs	Tax on labour	Tax on output	Tax on input	Total taxes	Unit 1 cost	Av'ge wages
1. Primary	3203	1246	1145	1212	0	0	0	0	-400	-400	0.234	945
2. Manufacturing	23089	16120	6844	-5812	0	-1	0	0	5939	5938	0.259	1485
3. Construction	4035	1832	1832	375	0	0	0	0	0	0	0.401	1350
4. Services	10298	2340	6435	999	520	-1	0	0	0	0	0.620	1080
Total	40630	21538	16261	-3226	520	-3	0	0	5539	5538	0.361	1363

Part 2	Ind'l price
1. Primary	0.654
2. Manufacturing	0.874
3. Construction	0.883
4. Services	0.991
Total	0.878

Idice version 3 final test run   1973 Commodity demands and supplies   AWAP 1 0 0: 0 0 56

Part 1	Consum -ption	Govern -ment	Invest -ment	Stock -bd'ng	Final demand	Interm demand	Home demand	Export demand	Total demand	Import supply	Import duties	Total output
1. Primaries	501	133	5	20	658	4183	4841	730	5571	610	61	4901
2. Manufactures	3505	1584	1857	200	7546	15683	23229	5292	28520	1754	351	26416
3. Construction	200	285	2089	5	2580	2084	4664	61	4724	150	0	4574
4. Services	4005	1204	806	100	6117	3752	5869	2525	12394	2000	0	10394
Total	8211	3606	4759	325	16901	25701	42602	8608	51210	4513	412	46285

Part 2	Total supply											
1. Primaries	5571											
2. Manufactures	28520											
3. Construction	4724											
4. Services	12394											
Total	51210											
Direct imports	9269	0	0	0	9269	0	9269	0	9269	9269	0	0
Indirect taxes	875	1091	456	0	2422	0	2422	0	2422	0	0	2422
Other value-added	0	6054	0	0	6054	0	6054	0	6054	0	0	6054
Foreign tourists	0	0	0	0	0	0	0	0	0	0	0	0
Grand total	18355	10752	5215	325	34646	25701	60347	8608	68955	13782	412	54751
Direct imports	9269											
Indirect taxes	2422											
Other value-added	6054											
Foreign tourists	0											
Grand total	68955											

Idice version 3 final test run   1973 Input-output flows at constant prices   AWAP 1 0 0: 0 0 56

	Primar y	Manufa cturin	Constr uction	Servic es
1. Primaries	98	2962	91	31
2. Manufactures	784	12151	1185	1559
3. Construction	98	1321	457	208
4. Services	441	2113	366	832
Total	1421	19548	2104	2630

Table 6: The Test Run Output (cont'd)

Table 6: The Test Run Output (cont'd)

Idiom version 3 final test run 1973 Sector incomes and expenditures AWAP 1 0 0: 0 0 56

	Households	Companies	Government	R.O.W.
1. Wages etc	22313	0	0	0
2. Taxes	0	0	7017	0
3. Profits etc	-645	-2581	0	0
4. Imports	0	0	0	9789
5. Goods & services	0	0	0	4305
Total	21667	-2581	7017	14093

	Households	Companies	Government	R.O.W.
1. Wages etc	0	0	6054	0
2. Taxes	1578	-2581	0	0
3. Profits etc	0	0	0	0
4. Imports	9269	0	0	0
5. Goods & services	8253	0	4860	8966
Total	19100	-2581	10914	8966

	Households	Companies	Government	R.O.W.
Sector saving	2568	0	-3897	5128

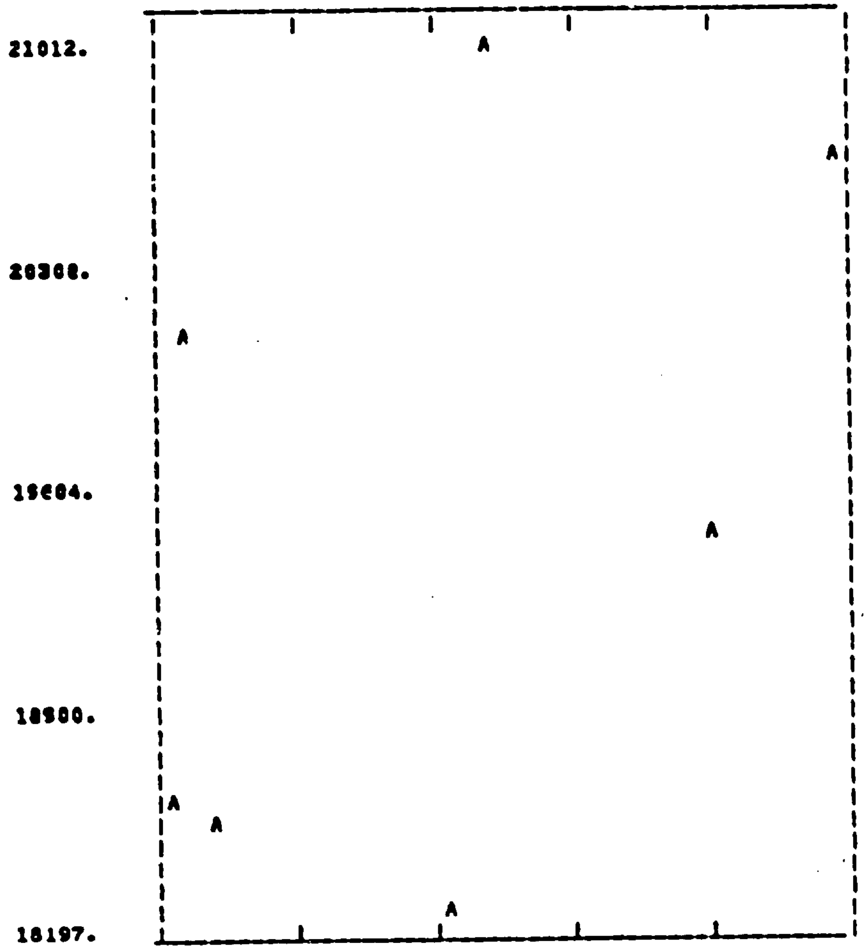
Idiom version 3 final test run 1973 Aggregate employment & unemployment AWAP 1 0 0: 0 0 56

	Ind'l empl't	Gov't empl't	H'hold empl't	Total empl't	Reg'rd unem't	Unregd unem't	Total unem't	Labour force	Reg'rd U rate	Total U rate
Total	13142	4485	1	17627	8401	0	8401	26028	32.28	32.28

Table 6: The Test Run Output (cont'd)

```
Plot of IDIOM variables - up to 9 Y-variables and 1 X-variable can be plotted
X-variable is on vertical axis, Y-variables on horizontal axis
If two or more Y-variables coincide, they are plotted as *

PDI          18196.    18900.    19604.    20308.    21012.
SPC        A 16388.    16851.    17314.    17776.    18235.    18701.
```



poorer members of society (i.e. those below the tax threshold) exceeds that of the richer members' Whether this is because of the problems with BDC - see above - or a separate problem remains to be determined, but clearly it must be resolved before IDIOM is to be trusted.

The results of the TABULATE 1979 instruction are not reproduced here - they are identical in form to those for 1973. Hence, following the cross-section tables here are some specimen time-series results. Those for C, Q and Y were requested, and the resulting tables are self-explanatory.

Finally we have a printer-graph. It plots PDI against SPC. Notice that the figures at the top of the figure define the scales on the axes, <u>not</u> the data. Note also that X is plotted vertically. At present there seems to be an error in the plotting of SPC.

ALTERATIONS TO THE TEST-RUN

This sub-section illustrates the flexibility of IDIOM by cataloguing a few very simple changes to the test-run. The changes are not of much interest in themselves, but they will give the user some feel for the programme.

(a) CONDITIONAL DIRECTIVES - ALTERING THE EXCHANGE RATE

The test run maintains the same exchange rate for all years, it is never redefined after the initialization process. In this example we modify this so that from 1974 to 1978, inclusive, EX is at the depreciated level of 0.9, while in 1979 it is dropped further to 0.8. The devaluation has fairly predictable effects on the economy: it raises prices, especially in the traded sectors, it stimulates activity and improves the trade balance.

The changes required are:

Directives: insert before line 89,

```
IF YEAR = 2  REAL EX .9
IF YEAR = 7  REAL EX .8
```

Note that spaces must appear between the logically separate entities:

YEAR, = , and 2, etc.

Data: none

Output: no changes in output structure occur, but, of course, the results are different. the user may perform this run himself if he wishes to examine fully the test economy's response to depreciation. Here we merely reproduce, in Table 7, the summary - MACR - table and the price - PY - table for 1979. The effects of the 1974 change have virtually all worked through by this time, while those of the 1979 change are far from complete.

Among the features to note, apart from the overall effects mentioned already are:

(i) the import price of primaries is inflated by the full extent of the depreciation - because the country is assumed to be a price-taker in this product. All other trade prices only partially reflect the exchange rate change;

(ii) the stock building figures are unaffected - because they are fixed;

(iii) consumption falls - as a result of the terms of trade effect;

(iv) the effects vary considerably according to industry - industries serving consumers suffer relatively (primaries - agriculture), those serving industry and/or exports benefit (manufacturing);

Table 7: The Consequences of Reducing Exports

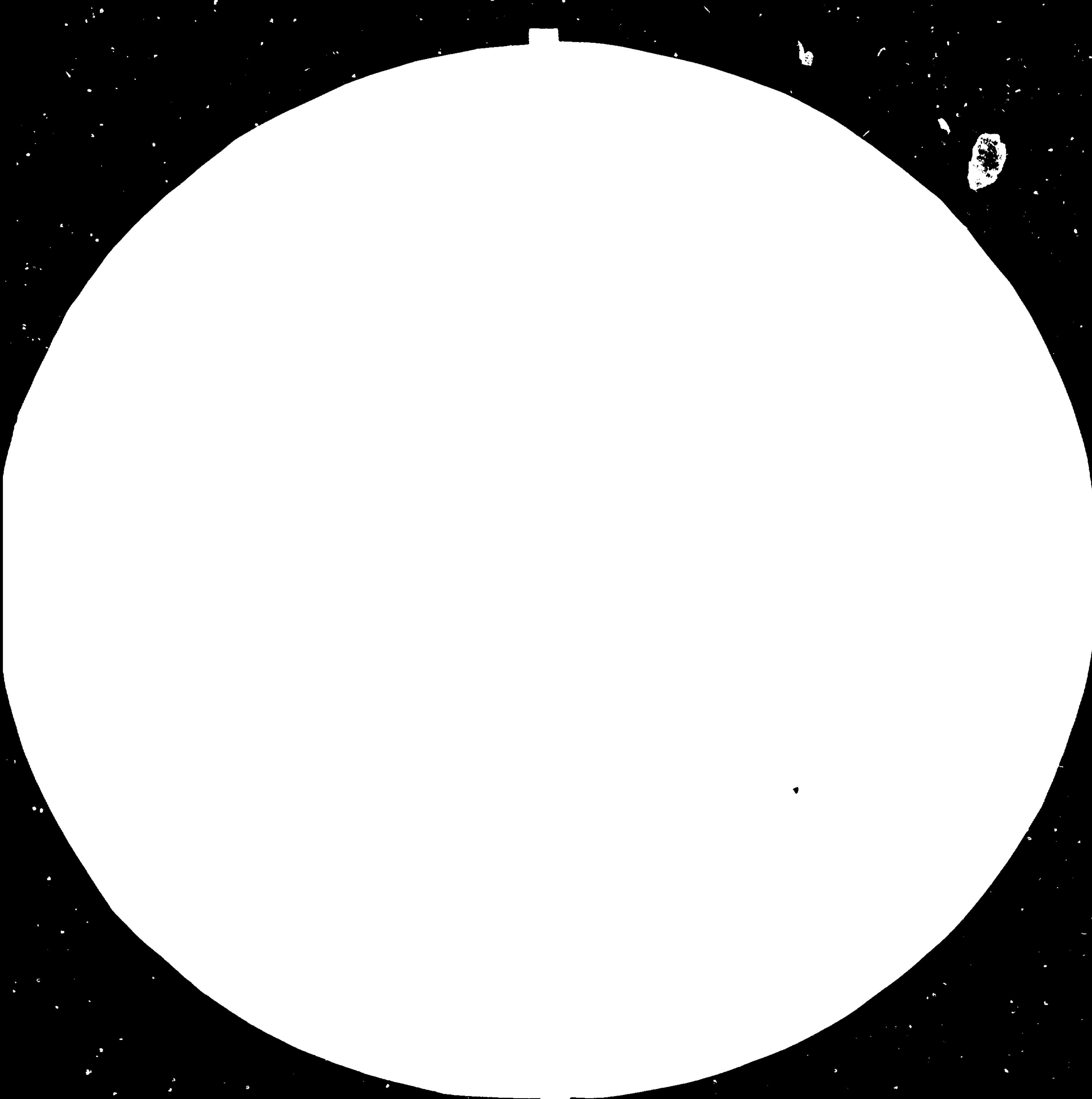
IDIOM VERSION 3 FINAL TEST RUN — 1973 USE OF RESOURCES AND CONDENSED TABLES — AWAP 1 0 0: 0 0 56

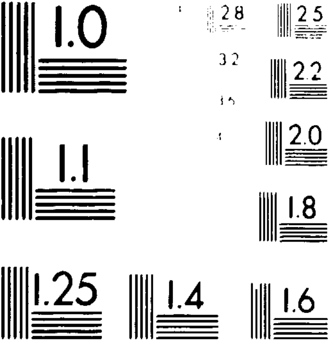
	GDP	C+EXP	GOVT	INVT	STKBG	EXPORTS	IMPORTS	TRADE		EMPLOYMENT	
1970 PRICES	20924	15904	9966	3755	325	2106	11132	BALANCE OF TRADE	-9133	EMPLOYMENT	13574
CURRENT PRICES	16222	14830	6781	3481	266	2052	11185	EXPORT INDEX	97.4	UNEMPLOYMENT	12454
PRICE INDICES	0.775	0.932	0.680	0.927	0.813	0.974	1.005	TERMS OF TRADE	97.0	UNEM RATE	47.85
INFLATION RATES	-22.47	-6.75	-31.96	-7.31	-18.67	-2.59	0.47	% CHANGE IN TT	-3.05	GROWTH EMPLT	-45.13
% GROWTH VOLUME	-58.80	-49.47	9.58	-59.96	-27.78	-81.32	2.25				
GROWTH FROM 1973	-58.80	-49.47	9.58	-59.96	-27.78	-81.32	2.25				

PART 1	CONSUM-PTION	GOVERN-MENT	INVEST-MENT	STOCK-BUING	FINAL DEMAND	INTERM DEMAND	HOME DEMAND	EXPORT DEMAND	TOTAL DEMAND	IMPORT SUPPLY	IMPORT DUTIES	TOTAL OUTPUT
1. PRIMARY	424	133	0	20	577	2558	3135	182	3317	0	0	3317
2. MANUFACTURING	2968	1984	1466	200	6617	9118	15735	1320	17055	829	166	16061
3. CONSTRUCTION	170	285	1649	5	2108	1245	3353	15	3368	150	0	3218
4. SERVICES	3392	1204	637	100	5333	2021	7354	589	7944	2000	0	5944
TOTAL COMMODITIES	6953	3606	3752	325	14636	14942	29577	2106	31683	2979	166	28539
DIRECT IMPORTS	7853	0	0	0	7853	0	7853	0	7853	7853	0	0
INDIRECT TAXES	741	678	337	0	1756	0	1756	0	1756	0	0	1756
OTHER VALUE-ADDED	0	6054	0	0	6054	0	6054	0	6054	0	0	6054
FOREIGN TOURISTS	0	0	0	0	0	0	0	0	0	0	0	0
GRAND TOTAL	15547	10338	4089	325	30300	14942	45241	2106	47347	10832	166	36350

IDIOM VERSION 3 FINAL TEST RUN — 1973 EXPORTS BY COMMODITY AND AREA — AWAP 1 0 0: 0 0 56

	N. AMERICA	EUROPE	R.O.W.	TOTAL	PRICE
1. VISIBLES	276	685	556	1517	0.947
2. INVISIBLES	176	237	176	589	1.045
TOTAL	452	923	731	2106	0.974





MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS

STANDARD REFERENCE MATERIAL 1010a

(ANSI and ISO TEST CHART No. 2)

(v) the extra demand for primaries (generated by industrial current demand) is met by imports - because imports are determined by FIXQ. Imports of construction and services are, on the other hand, fixed, being generated by option FIXM.

Although it is not evident from the print-outs it should be noted that certain traded-services prices are fixed in local currency by IDIOM, and hence are quite unresponsive to depreciation. These are:

PSYM, PSCM, PSGM, PSXM, PSQM

If it is desired that these reflect the exchange rate, it should be done explicitly - although redefining these variables to refer to foreign currency would be both a useful and relatively simple edit to perform.

(b) PSEUDO-FUNCTIONS - THE INSTITUTIONAL SECTORS

The pseudo-functions are those listed as R, T and H, which determine the order of, but not the calculation of, receipts and payments, indirect taxes and sectoral income flows respectively (see above). Hence, we merely reverse the order of the household and corporation sectors of the H classification - mainly to show just how easy it is. Obviously if it is done correctly re-ordering sectors should have no effect on the results.

The changes required are:

Directives: replace line 15 by:

FUNCTION H HOUS 2 CORP 1 CGOV 3 ROW 4

Input: change input lines 322 and 323 to:

1 CORPORATIONS
2 HOUSEHOLDS

(The line numbers are stored on the right-hand side of .FULLDATA preceded by four zeros and succeeded by one.

As stated previously, the results are unchanged by this alteration, but we present in Table 8 the income-expenditure accounts for 1973 on the revised basis.

(c) MAKING VARIABLES EXOGENOUS - EXPORTS

It is often useful to make certain potentially endogenous variables exogenous. This might be the case during testing or if there is insufficient information available to estimate even the simplest endogenous function. Most functional forms permit an explicit "FIX" option which allows the user to fix values by merely redefining the FUNCTION directive (provided, of course, that the desired fixed values are supplied in input and read by a READ PARS "FIX" directive). Where this facility is not provided, the easiest approach is to fix the regression constant to the desired level and fix remaining parameters to zero. We illustrate this by reference to exports.

The required changes are:

Directives: none

Input: replace lines 552 to 563 by:

```
Ø. log XA (1,1)     31*0
Ø. log XA (2,1)     31*0
Ø. log XA (3,1)     31*0
Ø. log XA (1,2)     31*0
Ø. log XA (2,2)     31*0
Ø. log XA (3,2)     31*0
```

The desired values are entered as logs because we have not bothered to redefine the functions from LLIN to LIN. The regression constant for flow (I,J) appears in BXA (I,J,2), hence the leading zero. As noted before, parameters are entered with the first classification index varying more quickly than the second.

Table 8: Reversing the Order of Sectors

IDIOM VERSION 3 FINAL TEST RUN 1973 SECTOR INCOMES AND EXPENDITURES

	COMPANIES	HOUSEHOLDS	GOVERNMENT	R.O.W.
1. WAGES ETC	0	21754	0	0
2. TAXES	0	0	8451	0
3. PROFITS ETC	-1795	-449	0	0
4. IMPORTS	0	0	0	9693
5. GOODS & SERVICES	0	0	0	3906
TOTAL	-1795	21305	8451	13599

	COMPANIES	HOUSEHOLDS	GOVERNMENT	R.O.W.
1. WAGES ETC	0	0	6054	0
2. TAXES	0	1451	0	0
3. PROFITS ETC	0	0	0	0
4. IMPORTS	0	9197	0	0
5. GOODS & SERVICES	0	8135	736	9016
TOTAL	0	18783	6790	9016

	COMPANIES	HOUSEHOLDS	GOVERNMENT	R.O.W.
SECTOR SAVING	-1795	2522	1661	4583

In our case we decided to reduce exports to 25 per cent of their test-run levels. The resulting table of exports and the macro-economic consequences are shown in Table 9. The latter are pretty predictable.

It may seem that this alteration could have been made as simply by other means, but this is not the case. It might seem that we could have used the MODIFY command: VIZ

```
COMPUTE  QMØ,PQM,PY,YV,YEØ,YS,DC,DCS,XA
MODIFY   XZ  1:1  554.
              .
              .
              .
COMPUTE ALL
```

However, there is a problem. MODIFY can only be obeyed at the end of the COMPUTE subroutine dealing with the function to be modified. If that function - IDMSXA in this case - calculates the function and then calculates something else from it, that something else will be calculated on the basis of the <u>endogenous</u> values of the function rather than the <u>exogenous</u> values. In the case of IDMSXA not only is XA calculated but QX, SX, SPX are also derived from it.

$$QX = [QXC] * (\sum^{NA} XA)$$

$$SX = \sum^{NA} \sum^{NX} XA$$

$$SPX = \sum^{NA} \sum^{NX} XA*PX$$

Hence, if we merely alter XA via MODIFY, it becomes inconsistent with these other export quantities. Also, in the rest of IDIOM it is QX rather than XA which is important (in constructing commodity balances, and hence industry outputs, etc.) and so the overall solution to IDIOM is hardly affected by the MODIFY XA command.

Table 9: <u>Some Results of Changing the Exchange Rate</u>

IDIOM VERSION 3 FINAL TEST RUN — 1979 USE OF RESOURCES AND CONDENSED TABLES — AWAP 1 0 01 0 0 56

	GDP	C'EXP	GOV'T	INV'T	STKB'G	EXPORTS	IMPORTS
1970 PRICES	38153	21922	9966	13009	325	11881	18950
CURRENT PRICES	31342	19245	6824	12222	269	11720	18937
PRICE INDICES	0.821	0.878	0.685	0.939	0.829	0.986	0.999
INFLATION RATES	1.62	1.34	0.13	0.94	2.72	5.15	2.96
% GROWTH VOLUME	2.09	-0.19	0.0	4.39	0.0	6.93	2.68
GROWTH FROM 1973	-24.87	-30.34	9.58	38.69	-27.78	5.37	74.06

TRADE	
BALANCE OF TRADE	-7218
EXPORT INDEX	98.6
TERMS OF TRADE	98.7
% CHANGE IN TT	2.13

EMPLOYMENT	
EMPLOYMENT	18989
UNEMPLOYMENT	7039
UNEM RATE	27.04
GROWTH EMPLT	0.72

PART 1	CONSUM -PTION	GOVERN -MENT	INVEST -MENT	STOCK -BD'NG	FINAL DEMAND	INTERM DEMAND	HOME DEMAND	EXPORT DEMAND	TOTAL DEMAND	IMPORT SUPPLY	IMPORT DUTIES	TOTAL OUTPUT
1. PRIMARY	614	133	0	20	767	5810	6578	1055	7632	2484	248	4900
2. MANUFACTURING	4299	1984	5077	200	11560	21369	32929	7648	40577	3343	669	36565
3. CONSTRUCTION	246	285	5711	5	6247	3038	9285	88	9373	150	0	9223
4. SERVICES	4913	1204	2208	100	8425	4814	13239	3090	16329	2000	0	14329
TOTAL COMMODITIES	10072	3606	12996	325	26999	35032	62031	11881	73912	7978	917	65017
DIRECT IMPORTS	10255	0	0	0	10255	0	10255	0	10255	10255	0	0
INDIRECT TAXES	961	682	1199	0	2842	0	2842	0	2842	0	0	2842
OTHER VALUE-ADDED	0	6054	0	0	6054	0	6054	0	6054	0	0	6054
FOREIGN TOURISTS	0	0	0	0	0	0	0	0	0	0	0	0
GRAND TOTAL	21287	10343	14196	325	46150	35032	81182	11881	93063	18232	917	73914

IDIOM VERSION 3 FINAL TEST RUN — 1979 COMMODITY, IMPORT AND EXPORT PRICES — AWAP 1 0 01 0 0 56

	PROD'N PRICES	HOME PRICES	IMPORT PRICES	EXPORT PRICES	ABSP'N PRICES
1. PRIMARIES	0.464	0.334	1.250	0.938	0.693
2. MANUFACTURES	0.899	0.888	0.929	0.938	0.893
3. CONSTRUCTION	0.854	0.854	1.033	0.938	0.857
4. SERVICES	0.800	0.711	0.800	1.124	0.724

Hence, to use MODIFY it would be necessary to MODIFY not only XA but also QX, SX and SPX. However, as presently constituted, MODIFY cannot be used on macro-economic variables, so this path is blocked. This represents a considerable restriction on the usefulness of the MODIFY command and should be edited out at some time.

Similar problems appear to afflict the following functions:

Subroutine	Additional variables determined *after* main function
IDMSDC	DSC,DSPC,SC,SPC,C,SPC,PCE,QCO
HR	D,DPDI,PDI,RPDI,H,PSFD,BP,SRAT,EPDI
PE	AW,AWY,AWC,AWG
PG	SPG,PSG
PM	QMTO,PM
PV	SPVM,PSVM
PX	PQX
PY	PQH,PQHH,YP
Q	Y
QM	SQM,SPQM,PSQM,SM,PSM,SPM,PSXM,SPYM, etc.
SC (DSC)	DSPC,SC,SPC
T	T,TO
UE	EMP,UNEM
V	SPV,SV,QV,PSV
XA	SX,SPX,QX,PSX
YEO	EMPY
YS }	{ QS,SYS,SPYS
(YSA) }	{ SYSA,SPS,SSA,PSS,SS
YV }	{ SVY

REFERENCES

BARKER, T.S. (1970): "The Determinants of Britain's Visible Imports, 1949-66", Volume 10 of Stone JRN (ed.), "A Programme for Growth", Chapman and Hall, London.

BARKER, T.S. (ed.): "Economic Structure and Policy", No. 2 of Cambridge Studies in Applied Econometrics, Chapman and Hall, London.

BARKER, T.S. (1977): "International Trade and Economic Growth: An Alternative to the Neo-Classical Approach", Cambridge Journal of Economics, Vol. 1, pp. 153-172

BARKER, T.S; BOROOAH, V; van der PLOEG, F; and WINTERS, L.A. (1980): "The Cambridge Multisectoral Dynamic Model: An Instrument for National Economic Policy Analysis", Journal of Policy Modelling, Vol. 2, pp. 319-344.

COUTTS, K.; WAH, Godley; and NORDHAUS, W. (1978): "Industrial Pricing in the U.K.", Cambridge University Press, Cambridge, U.K.

DEATON, A.S. (1975): "Models and Projections of Demand in Post-War Britain", No. 1, of Cambridge Studies in Applied Econometrics, Chapman and Hall, London, U.K.

DEATON, A.S. and MUELLBAUER, J. (1980): "Economics and Consumer Behaviour", Cambridge University Press, Cambridge, U.K.

HENDRY, D.F. and MIZON, G.E. (1978): "Serial Correlation as a Convenient Simplification", Economic Journal, Vol. LXXXVIII, pp. 549-563.

HENRY, S.G.; SAWYER, M.C., SMITH, P (1976): "Models of Inflation in the U.K.: An Evaluation", National Institute Economic Review, No. 77, August, pp. 60-71.

JORGENSON, D.W. (1963): "Captial Theory and Investment Behaviour", American Economic Association Papers and Proceedings, pp. 247-259

LLEWELLYN, G.E.J. (1974): "The Determinants of United Kingdom Import Prices", Economic Journal, Vol. LXXXIV, pp. 18-31.

PETERSON, A.W.A. (1978): "A Recursive Model of Employment and Hours Worked", Cambridge Growth Project Paper GPP 461, June 1978, Cambridge, U.K.

WINTERS, L.A. (1981): "An Econometric Model of the Export Sector", No. 4 of Cambridge Studies in Applied Econometrics, Cambridge University Press, Cambridge, U.K.

APPENDICES

a) TABLS

b) TABLO

c) UNITS

d) RUNDATA prepared for input into UNIDO reading Routine DLIMIT

e) ERROR MESSAGES

f) PREPROCESSOR

```
              IDIOM - TABLS
    ** LIST OF IDIOM SYMBOLIC NAMES **

    The following list contains the names of variables,
    parameters, switches & converters for IDIOM.  The
    columns of the list are
    First column  : symbolic name
    Second and third columns : identifiers of internal
           IDIOM classifications for rows & cols
    Fourth column : units of measurement
    Fifth column  : identifier for type of IDIOM store
                    0 no store
                    1 fixed store for variables
                    2 fixed store for parameters
                    3 store for sparse matrices
                    4 store for switches
Q     1  0 1 1  commodity outputs
QCO   1  0 1 1  commodities purchased by consumers
QGO   1  0 1 1  commodities purchased by government
QMC   1  0 1 1  commodity imports
QSO   1  0 1 1  stockbuilding by commodity content
QVO   1  0 1 1  commodities investment by source
QVAT  1  0 5 0  value-added tax rates on commodity outputs
QXO   1  0 1 1  commodity exports
QYO   1  0 1 1  commodities absorbed by industries
PQ    1  0 3 1  prices of commodities
PQH   1  0 3 1  prices of home sales
PCHH  1  0 3 1  prices of home sales by home producers
PQH1  1  0 3 1  PCHH lagged 1 year
PCH2  1  0 3 1  PCHH lagged 2 years
PQM   1  0 3 1  prices of imported commodities
PQX   1  0 3 1  prices of export sales
QMTO  1  0 2 1  tariffs on imported commodities
QMTZ  1  0 5 1  tariff rates on imported commodities base year
QMQ   1  0 6 1  quotas on commodity imports
QXTO  1  0 2 1  indirect taxes on exported commodities
QXTZ  1  0 5 1  indirect tax rates exports base year
Y     2  0 1 1  industry outputs
YL1   2  0 1 1  industry outputs lagged 1 year
YL2   2  0 1 1  industry outputs lagged 2 years
YL3   2  0 1 1  industry outputs lagged 3 years
YEXP  2  0 1 1  expected level of industry output
YEO   2  0 4 1  industry employment
YEL1  2  0 4 1  industry employment lagged 1 year
YH    2  0 2 1  average hours worked per week in industry
YHL1  2  0 2 1  average hours worked lagged 1 year
YULC  2  0 6 1  industrial unit labour costs (£ / unit output)
YP    2  0 2 1  industry profits
YV    2  8 1 1  industrial investment by destination and asset
PVA   2  8 3 1  present value of investment incentives
PYV1  2  8 3 1  incentive inclusive asset prices lagged 1 year
PYV2  2  8 3 1  incentive inclusive asset prices lagged 2 years
YS    2  0 1 1  stockbuilding by industry of destination and asset
PYS   2  0 3 1  price indices for industry stocks
YSA   2  0 2 1  industry stock appreciation
PY    2  0 3 1  prices of industry outputs
PYE   2  0 3 1  average earnings by industrial employment
PYVP  2  0 3 1  prices of equipment by destination
YRO   2  0 2 1  value-added in industries
```

APPENDIX A

IDIOM - TABLES

YTO	2	0	2	1	indirect taxes on industry	00000590
YTZ	2	0	5	1	indirect tax rates industries base year	00000600
C	3	0	1	1	consumers' expenditures	00000610
PC	3	0	3	1	prices of consumers' expenditures	00000620
CEO	3	0	4	1	employment by consumers	00000630
CEOB	3	0	8	1	labour required per £ mn consumers' expenditure	00000640
CRO	3	0	2	1	value-added in consumers' expenditures	00000650
CTO	3	0	2	1	indirect taxes on consumers' expenditure	00000660
CTZ	3	0	5	1	indirect tax rates consumers' exp. base year	00000670
HPD	3	0	5	1	hire purchase deposits as proportion of cost	00000680
G	4	0	1	1	government current expenditure	00000690
PG	4	0	3	1	prices of gov. cur. exp.	00000700
GEO	4	0	2	1	employment by government	00000710
GEOB	4	0	8	1	labour required per £ mn government expenditure	00000720
GRO	4	0	2	1	value-added in government current exp.	00000730
GTO	4	0	2	1	indirect taxes on gov. cur. exp.	00000740
GTZ	4	0	5	1	indirect tax rates government exp. base year	00000750
R	5	0	2	1	receipts and payments in aggregate	00000760
RYO	5	0	2	1	vector of factor and tax payments by industry	00000770
RCO	5	0	2	1	vector of factor and tax payments in consumers' exp.	00000780
RGO	5	0	2	1	vector of factor and tax payments in government exp.	00000790
RVO	5	0	2	1	vector of factor payments investment goods	00000800
H	6	0	2	1	vector of institutional financial surpluses	00000810
HRR	6	5	2	1	matrix of receipts by institutions	00000820
HRP	6	5	2	1	matrix of payments by institutions	00000830
S	7	0	1	1	stockbuilding by asset	00000840
V	8	0	1	1	investment by asset	00000850
VKO	8	0	1	1	government capital expenditure by asset	00000860
PV	8	0	3	1	price of investment by asset	00000870
VRO	8	0	2	1	factor payments for investment by asset	00000880
VTO	8	0	2	1	indirect taxes on investment by asset	00000890
VTZ	8	0	5	1	indirect tax rates investment exp. base year	00000900
XA	10	15	1	1	export matrix export group by area	00000910
PX	10	0	3	1	prices of export groups	00000920
PXL1	10	0	3	1	export price lagged 1 year	00000930
PXL2	10	0	3	1	export price lagged 2 years	00000940
PCL0	10	0	3	1	competitors export price $	00000950
PCL1	10	0	3	1	competitors export prices lagged 1 year	00000960
PCL2	10	0	3	1	competitors export prices lagged 2 years	00000970
PCL3	10	0	3	1	competitors export prices lagged 3 years	00000980
DI2	10	0	3	1	institutional dummy for exports by export	00000990
XTO	10	0	5	1	tax rates on exports by export	00001000
XTC1	10	0	5	1	tax rates on exports lagged 1 year	00001010
XTC2	10	0	5	1	tax rates on exports lagged 2 years	00001020
M	11	0	1	1	imports by import group	00001030
PFM	11	0	3	1	prices of import groups in foreign currencies	00001040
PM	11	0	3	1	prices of import groups in sterling	00001050
E	12	0	4	1	employment by type in total	00001060
UE	12	0	4	1	unemployment by type of labour	00001070
LF	12	0	4	1	labour force by type of labour	00001080
PE	12	0	2	1	wage rates by type of labour	00001090
PEL1	12	0	2	1	wage rates lagged 1 year	00001100
EYO	12	0	4	1	industrial employment by type	00001110
ECO	12	0	4	1	personal employment by type	00001120
EGO	12	0	4	1	government employment by type	00001130
D	13	0	5	1	tax receipts for direct tax brackets	00001140
DN	13	0	6	1	proportion of labour force by tax bracket	00001150
DBT	13	0	6	1	income limits by tax brackets	00001160

IDIOM - TABLS

DAB	13	5	6	1	matrix of household tax allowances by income type	00001170
DDB	13	5	6	1	matrix of marginal tax rates by income type	00001180
DC	13	3	1	1	consumption by tax bracket and functional category	00001190
DPOP	13	0	4	1	population by tax bracket of household	00001200
DPDI	13	0	2	1	disposable income by tax bracket	00001210
DSC	13	0	1	1	aggregate consumption by tax bracket	00001220
DSPC	13	0	2	1	value of consumption by tax bracket	00001230
DPCE	13	0	3	1	price index of consumption by tax bracket	00001240
DPW	13	0	2	1	aggregate wealth by tax bracket	00001250
DPWH	13	0	2	1	per capita wealth by tax bracket	00001260
T	14	0	2	1	indirect tax revenues by type	00001270
TO	14	17	2	1	indirect tax revenues for each SAM group	00001280
TQMO	14	0	2	1	indirect tax revenues imports	00001290
TCXO	14	0	2	1	indirect tax revenues exports	00001300
TYO	14	0	2	1	indirect tax revenues industries	00001310
TCO	14	0	2	1	indirect tax revenues consumers' exp.	00001320
TGC	14	0	2	1	indirect tax revenues government exp.	00001330
TVO	14	0	2	1	indirect tax revenues investment	00001340
PWLO	15	0	3	1	domestic prices by export area $	00001350
PWL1	15	0	3	1	domestic prices by export area lagged 1 year	00001360
PWL2	15	0	3	1	domestic prices by export area lagged 2 years	00001370
PWL3	15	0	3	1	domestic prices by export area lagged 3 years	00001380
AD1	15	0	3	1	aggregate demand measure I	00001390
AD2	15	0	3	1	aggregate demand measure II	00001400
DI1	15	0	3	1	institutional dummy for exports by area	00001410
K	16	0	1	1	government capital expenditures	00001420
BQM	1	0	8	2	import-output ratio functions	00001430
BPCM	1	0	8	2	import price functions	00001440
BPY	2	0	8	2	mark-up price equations or profit share functions	00001450
BYV	2	8	8	2	investment functions for industries and fixed asset	00001460
BYE	2	12	8	2	employment functions for industries	00001470
BYS	2	7	8	2	function for stockbuilding by industry and asset	00001480
BDC	13	2	8	2	consumers' expenditure functions by tax bracket	00001490
BDSC	13	0	8	2	aggregate consumption function by tax bracket	00001500
BF	5	0	8	2	receipts and payments definitions	00001510
BXA	10	15	8	2	export functions by export groups and areas	00001520
BPX	10	0	8	2	export price functions by export commodity	00001530
BPE	12	0	8	2	wage function for each employment groups	00001540
BT	14	0	8	2	indirect tax definitions	00001550
BV	8	0	8	2	investment functions for non-industrial assets	00001560
BULC	2	0	8	2	unit labour cost functions	00001570
BC	1	0	8	2	functions relating commodity prices to industries	00001580
BUE	12	0	8	2	unemployment/participation functions	00001590
BHF	6	5	8	2	functions for receipts and payments by sector	00001600
BH	6	0	8	2	institutional sector definitions	00001610
BCYC	1	2	8	2	input-output coefficient functions	00001620
JBCM	1	0	9	4	switch for import ratio functions	00001630
JBPM	1	0	9	4	switch for import price functions	00001640
JBPY	2	0	9	4	switch for industry price functions	00001650
JBVY	2	8	9	4	switch for industry investment functions	00001660
JBYE	2	12	9	4	switch for industry employment functions	00001670
JBYS	2	7	9	4	switch for industry stockbuilding functions	00001680
JBDC	13	3	9	4	switch for consumers' expenditure functions	00001690
JBSC	13	0	9	4	switch for aggregate consumption functions	00001700
JBF	5	0	9	4	switch for receipts & payments functions	00001710
JBXA	10	15	9	4	switch for export functions	00001720
JBPX	10	0	9	4	switch for export price functions	00001730
JBPE	12	0	9	4	switch for wage functions	00001740

```
                           IDIOM - TABLS
JBT   14  0 9 4   switch for indirect tax definitions                       00001750
JPV    8  0 9 4   switch for non-industrial investmaent functions           00001760
JBLC   2  0 9 4   switch for unit labour cost functions                     00001770
JBC    1  0 9 4   switch for commodity price functions                      00001780
JBUE  12  0 9 4   switch for unemp.oyment/participation functions           00001790
JBHF   6  5 9 4   switch for sectoral receipt functions                     00001800
JBH    6  0 9 4   switch for institutional definitions                      00001810
JBQY   1  2 9 4   switch for input-output coefficient functions             00001820
TCME  14  1 5 3   indirect tax rates on imports                             00001830
ITMB  14  1 9     identifier TCMB                                           00001840
TYB   14  2 5 3   indirect tax rates on industries                          00001850
ITYB  14  2 9     identifier TYB                                            00001860
TCB   14  3 5 3   indirect tax rates on consumers' exp.                     00001870
ITCB  14  3 9     identifier TCB                                            00001880
TGB   14  4 5 3   indirect tax rates on government curr. exp.               00001890
ITGB  14  4 9     identifier TGB                                            00001900
TVB   14  7 5 3   indirect rax rates on investment                          00001910
ITVB  14  7 9     identifier TVB                                            00001920
TCXE  14  1 5 3   indirect tax rates on exports                             00001930
ITXB  14  1 9     identifier TCXB                                           00001940
RYB    5  2 8 3   parameters for industry payments                          00001950
IRYB   5  2 9     identifier RYB                                            00001960
RCB    5  3 8 3   parameters for consumer payments                          00001970
IRCB   5  3 9     identifier RCB                                            00001980
RGB    5  4 8 3   parameters for government current payments                00001990
IRGB   5  4 9     identifier RGB                                            00002000
RVB    5  8 6 3   parameters for payments on investment goods               00002010
IRVB   5  8 9     identifier RVB                                            00002020
QYC    1  2 6 3   input-output matrix                                       00002030
IQYC   1  2 9     identifiers for QYC                                       00002040
MCC   11  1 6 3   classification converter commodities to imports           00002050
IMCC  11  1 9     identifiers for MCC                                       00002060
YCC    2  1 6 3   classification converter commodity to industrial outpu00002070
IYCC   2  1 9     identifiers for YQC                                       00002080
QCC    1  3 6 3   classification converter consumers' exp. to commoditie00002090
ICCC   1  3 9     identifiers for QCC                                       00002100
QGC    1  4 6 3   classification converter gov. cur. exp. to commodities00002110
IQGC   1  4 9     identifiers for QGC                                       00002120
QSC    1  6 6 3   classification converter stockbuilding to commodities 00002130
IQSC   1  6 9     identifiers for QSC                                       00002140
QVC    1  7 6 3   classification converter fixed investment to commoditi00002150
IQVC   1  7 9     identifiers for QVC                                       00002160
QXC    1  9 6 3   classification converter exports to commodities           00002170
IQXC   1  9 9     identifiers for QXC                                       00002180
VKC    8 16 6 3   classification converter gov. capital exp. to assets      00002190
IVKC   8 16 9     identifiers for KVC                                       00002200
QYPC   1  2 6 3   classification converter industrial plant to commoditi00002210
IQYP   1  2 9     identifiers for QPY                                       00002220
EYEC  12  2 6 3   classification converter industrial employment to type00002230
IEYC  12  2 9     identifiers for EYEC                                      00002240
ECEC  12  3 6 3   classification converter consumer employment to type      00002250
IECC  12  3 9     identifiers for ECEC                                      00002260
EGEC  12  4 6 3   classification converter government employment to type00002270
IEGC  12  4 9     identifiers for EGFC                                      00002280
HRC    6  5 6 3   classification converter receipts to institutions         00002290
IHFC   6  5 9     identifiers for HRB                                       00002300
DRC   13  5 6 3   classification converter households to tax brackets       00002310
IDFC  13  5 9     identifiers for DRB                                       00002320
```

APPENDIX A

```
                        IDIOM - TABLS
QYSC  1   2 6 3   classification converter industry stocks to commoditie00002330
ICYS  1   2 9     identifiers for QYSC                                   00002340
QTI   1   0 7     titles of commodities                                  00002350
YTI   2   0 7     titles of industries                                   00002360
CTI   3   0 7     titles of consumers' expenditures                      00002370
GTI   4   0 7     titles of government current expenditures              00002380
RTI   5   0 7     titles of receipts and payments                        00002390
HTI   6   0 7     titles of institutions                                 00002400
STI   7   0 7     titles of stockbuilding assets                         00002410
VTI   8   0 7     titles of fixed investment assets                      00002420
FTI   9   0 7     titles of financial investment assets                  00002430
XTI  10   0 7     titles of export groups                                00002440
MTI  11   0 7     titles of import groups                                00002450
ETI  12   0 7     titles of employment                                   00002460
RTI  13   0 7     titles of tax brackets for income tax                  00002470
TTI  14   0 7     titles of indirect taxes                               00002480
ATI  15   0 7     titles of areas                                        00002490
                  ** CONTENTS OF 'O' STORE **                            00002500
                  The store with symbolic name O contains macrovariables00002510
                  arranged as follows:-                                  00002520
                        Q    Y    C    G    X    R    H    S    V    F   00002530
                        1    2    3    4    5    6    7    8    9   10   00002540
                    1 DFE  SYVA SCVA SGVA SYM  SCM  SGM  SPYM SPCM SFGM 00002550
                   11 SM   GDP  SC   SG   SX   SR   RPDI SS   SV   WVCL 00002560
                   21 SPM  GDPC SPC  SPG  SFX  SCM  PDI  SPS  SPV  WPRI 00002570
                   31 PSM  HUC  PCE  PSG  PEX  SPCM  37  PSS  PSV  SSA  00002580
                   41 PSQM PSYM PSCM PSGM PSXM POF  WPOP SYS  POPL SGS  00002590
                   51 AW   AWY  AWC  AWG   EE  EMPL UNEM EMPY EMPC EMPG 00002600
                   61 RVAT  62  SRT  FET  EX   FINT RMCF SFYS GDPT SPGS 00002610
                   71 BT   BP   PSER PSFD EXL1 FWPH SRAT SYSA RETL SGSA 00002620
                   81 SY   SVVA SPVM SVM  EXL2 XFT  EPDI XFFT SYV  PUP  00002630
                   91 SPY  WAGY WAGC WAGG WAGE TXPD TXPI FCET RESE PW   00002640
                  101 101  102  103  104  105  106  107  108  109  110  00002650
                  111 GDPL SCL  SGL  SVL  SSL  SXL  SML  EML1 UNL1 PDIL 00002660
                  121 HUCL PCEL PSGL PSVL PSSL PSXL PSML AWL  RDI1 RDI2 00002670
                  131 GDPI SCB  SGB  SVB  SSB  SXB  SMB  EMB  139  140  00002680
                  141 141  142  143  144  145  STAR FINI YEAR DATE 150  00002690
                  151 PBAS QBAS YA70 YA72 155  156  157  158  159  160  00002700
OTI  17   0 7     titles of overall SAM classification                   00002710
```

```
            IDIOM - TABLE
  1 DFE  SYVA SCVA SGVA SYM  SCM  SGM  SPYM SPCM SPGM 00000010
 11 SM   GDP  SC   SG   SX   SF   RPDI SS   SV   WVOL 00000020
 21 SPM  GDPC SPC  SPG  SPX  SQM  PDI  SPS  SPV  WPRI 00000030
 31 PSM  HUC  PCE  PSG  PSX  SPCM  37  PSS  PSV  SSA  00000040
 41 PSQM PSYM PSCM PSGM PSXM POP  WPOP SYS  PCPL SGS  00000050
 51 AW   AWY  AWC  AWG   55  EMPL UNEM EMPY EMPC EMPG 00000060
 61 RVAT  62  SRT  RET  RX   RINT RMCR SPYS GDPT SPGR 00000070
 71 BT   BP   PSIR PSPD EXL1 FWPH SRAT SYSA RETL SGSA 00000080
 81 SY   SVVA SPVM SVM  EXL2 XPT  EPDI XPPT SYV  PUP  00000090
 91 SPY  WAGY WAGC WAGG WAGE TXPD TXPI FCET RESE PW   00000100
101 101  102  103  104  105  106  107  108  109  110  00000110
111 GDPL SCL  SGL  SVL  SEL  SXL  SPL  EML1 UNL1 PCTL 00000120
121 HUCL PCEL PSGL PSVL PSEL PSXL PSML AWL  RDI1 RDI2 00000130
131 GDPB SCB  SGB  SVB  SSB  SXB  SMB  EPB  139  140  00000140
141 141  142  143  144  145  STAR FINI YEAR DATE 150  00000150
151 PBAS QBAS YA70 YA72 155  156  157  158  159  160  00000160
```

APPENDIX B

APPENDIX C

PAGE 0001

IDIOM - UNITS

1£(1970)M	£million constant 1970 prices	F8.0	00000010
2 £M	£million current prices	F8.0	00000020
370=1.000	unit-value index 1970=1.000	F8.3	00000030
4 000	thousands	F8.0	00000040
5tax rate	tax rates,ratios or allowances	F8.3	00000050
6 ratios		F8.3	00000060
7 titles		6A4	00000070
8 params	estimated parameters	F8.4	00000080
9 ident.	identifiers or switches	I6	00000090

APPENDIX D

```
                    IDIOM - C3 - RUNDATA                                  PAGE 0001
 4517
47616
 6568
27264
@
 1518
11443
  776
12156
@
  133
 1584
  288
 1204
@
 3586
 7135
    0
 2545
@
  138
  825
   55
    0
@
   15
 4032
 4565
 1260
@
 1265
 8601
 26
 3295
@
 5058
28192
 1217
11849
@
1
1
1
1
@
1
1
1
1
@
1
1
1
1
@
1
1
1
```

APPENDIX D

```
                    IDIOM - C3 - RUNDATA                                        PAGE 0002
1
@
  22
 327
   0
   0
@
  .1
  .2
   0
   0
@.
0
0
0
0
@
0
0
0
0
@
0
0
0
0
@
 4703
49255
 6735
26041
@
 4520
 45000
 6650
 25500
@
 4540
 48500
 6500
 25000
@
 4500
 46000
 6500
 24500
@
 1139
 8263
 1733
 9477
@
 1150
 5290
 1850
 9300
@
 0
```

APPENDIX E

IDIOM - ERROR MESSAGES

IDIOM program errors

ERROR	Illegal PHASE change

INITIALISE phase errors

ERROR	Command not found in INITIALISE phase
ERROR	SETUP may only be invoked once
ERROR	Dimension/symbol table not initialised
ERROR	Unrecognised identifier
ERROR	Duplicated identifier
ERROR	Negative dimension
ERROR	Dimension too large
ERROR	Unrecognised option
ERROR	Unrecognised SWITCH-name
ERROR	Too late to SWITCH ABCHECK
ERROR	Error-number omitted
ERROR	Unrecognised identifier
ERROR	Unrecognised function definition
ERROR	Function switch out of range
ERROR	Function switch unset
ERROR	Unrecognised first identifier
ERROR	Unrecognised converter-identifier
ERROR	Duplicated converter-identifier
ERROR	Conversion type omitted
ERROR	Unrecognised conversion type
ERROR	Unrecognised second identifier
ERROR	Unset function definition
ERROR	Model too big : pars overflow
ERROR	Model too big : conv overflow
ERROR	Symbol pointer fault
ERROR	Unset dimension
ERROR	Model too big :stack overflow
ERROR	Model too big :switch overflow
ERROR	Unrecognised identifier in list
ERROR	Fatal error
ERROR	Error count exceeds maximum
ERROR	Too many continuations
ERROR	NAME option not recognised
ERROR	NAME option used twice
ERROR	NAME not fully specified
ERROR	Unrecognised SETUP option
ERROR	Too many symbols declared
ERROR	Too many function symbols
ERROR	Too many classification converters
ERROR	Lagged value not found
ERROR	Not enough workspace to SETUP DUMP
ERROR	Member-list too long
ERROR	Char string not found
ERROR	Char string terminator not found
ERROR	No integer found
ERROR	Invalid integer
ERROR	No digits found
ERROR	Invalid real
ERROR	Illegal decimal point
ERROR	Omitted decimal point
ERROR	Invalid exponent

APPENDIX E

IDIOM - ERROR MESSAGES

ERROR	No digits found
ERROR	Invalid real
ERROR	Illegal decimal point
ERROR	Omitted decimal point
ERROR	Invalid exponent
ERROR	Faulty integer
ERROR	FUNCTION name not found
ERROR	Illegal COMPRESS directive
ERROR	Illegal COMPRESS option

INPUT phase errors

ERROR	Command not found in INPUT phase
ERROR	Variables not initialised
ERROR	Invalid option
ERROR	Command stream reserved
ERROR	Verification stream reserved
ERROR	Unrecognised descriptor
ERROR	Attempt to read OUTPUT stream
ERROR	Only units 13 & 14 can be accessed as DATABANKs
ERROR	Not enough workspace to read db index
ERROR	DATABANK index cannot be read
ERROR	DUMP selected for INPUT is empty
ERROR	Attempt to write to INPUT stream
ERROR	OUTPUT to DATABANK not implemented
ERROR	Warning : DUMP selected for OUTPUT is not empty
ERROR	Terminator not found
ERROR	Specification too long
ERROR	Attempt to declare DA stream sequential
ERROR	Attempt to declare sequential stream DA
ERROR	Only unit 15 can be selected as DUMP
ERROR	Not enough workspace to process DUMP
ERROR	DUMP cannot be opened
ERROR	Omitted list
ERROR	Wrong no of items in list
ERROR	Unrecognised macro variable
ERROR	Only 1 identifier permitted
ERROR	Omitted unit-number
ERROR	Omitted code-number
ERROR	Attempt to READ illegalstream
ERROR	Attempt to GET from illegal file
ERROR	Unrecognised identifier
ERROR	Attempt to update lagged variable illegally
ERROR	Illegal attempt to read parameters
ERROR	Only 1 identifier permitted
ERROR	Omitted unit-number
ERROR	Omitted code-number
ERROR	Attempt to READ illegalstream
ERROR	Attempt to GET from illegal file
ERROR	Unrecognised identifier
ERROR	Stack overflow
ERROR	Too many BASE variables
ERROR	READ BASE used before ANALYSE phase
ERROR	Unrecognised title classification
ERROR	Stack overflow
ERROR	Stack overflow
ERROR	Unrecognised identifier
ERROR	Only 1 identifier permitted
ERROR	Attempt to READ illegal stream

APPENDIX E

IDIOM - ERROR MESSAGES

ERROR	Attempt to GET from illegal file
ERROR	Unrecognised function definition
ERROR	Stack overflow reading function
ERROR	READ ALL forbidden
ERROR	DUMP is not current INPUTstream
ERROR	GET MODIFICATIONS not permitted
ERROR	Attempt to READ illegal stream
ERROR	Too many modifications
ERROR	Unexpected end of MODIFICATION packet
ERROR	Input stream not open
ERROR	GEM index NR not found
ERROR	GEM index NR invalid
ERROR	GEM index NC invalid
ERROR	Array bound error
ERROR	Illegal READ attempted
ERROR	Transmission error
ERROR	Stack overflow reading sparse matrix
ERROR	End of file encountered on input
ERROR	Matrix not on databank
ERROR	Matrix is full on db : cannot be SPARSE in IDIOM
ERROR	Attempt to GET from illegal file
ERROR	Attempt to GET row/col of sparse array
ERROR	Syntax error in row/col reference
ERROR	Faulty dimensions of matrix on db
ERROR	Error in dimensions
ERROR	Not enough store for GET workspace
ERROR	Row/col reference outside matrix
ERROR	Zero index in sparse matrix
ERROR	Faulty index in sparse matrix
ERROR	Store overwritten
ERROR	Unrecognised display specification
ERROR	Unrecognised identifier in list

SOLVE phase errors

ERROR	Command not found in SOLVE phase
ERROR	Command not allowed inside ITER loop : ignored
ERROR	Command not allowed inside YEAR loop : ignored
ERROR	Unrecognised key word
ERROR	Unrecognised operand
ERROR	Unrecognised operator
ERROR	Omitted command
ERROR	Omitted list
ERROR	Too many convergence tests
ERROR	Wrong no of items in list
ERROR	Unrecognised variable
ERROR	Stack overflow
ERROR	ON/OFF omitted
ERROR	Illegal state
ERROR	Unrecognised function name
ERROR	Unrecognised function name
ERROR	Function specified twice
ERROR	Illegal FOR-loop
ERROR	Illegal FOR-loop variable
ERROR	Illegal FOR-loop syntax
ERROR	Unexpected LOOP termination
ERROR	LOOP logic fault
ERROR	Omitted unit-number

APPENDIX E

IDIOM – ERROR MESSAGES

ERROR	Unrecognised identifier
ERROR	Incompatible identifiers
ERROR	Earlier results for current year still active
ERROR	Not enough workspace for PUT/GET command
ERROR	DUMP not initialised
ERROR	Illegal PUT option
ERROR	Not enough space left in DUMP file
ERROR	Year not specified in GET ALL command
ERROR	Year specified not found on DUMP
ERROR	Dimensions of model on DUMP are wrong
ERROR	Illegal FREE option
ERROR	Not enough space in DUMP index
ERROR	Attempt to use sequential stream for PUT/GET
ERROR	DUMP not SELECTed
ERROR	Attempt to create untitled DUMP file
ERROR	Heading truncated
ERROR	PAGE/NEWLINE/HEADING ignored for non-printer stream
ERROR	Attempt to write negative number of NEWLINES
ERROR	Unrecognised identifier in list
ERROR	Unrecognised identifier
ERROR	Fault in array reference
ERROR	Operator not found
ERROR	Fault in IDMADJ call
ERROR	Insufficient space to unpack FUNCTION matrix

ANALYSE phase errors

ERROR	Command not found in ANALYSE phase
ERROR	Aggregation fault
ERROR	Operation fault
ERROR	Dictionary full
ERROR	Name not given
ERROR	Symbol has not been found
ERROR	Dimensions do not match
ERROR	No definition given
ERROR	Symbol already used
ERROR	Stack overflow
ERROR	Definition nested too deep
ERROR	DEFINE stack overflow
ERROR	Faulty dimension
ERROR	Stack underflow
ERROR	Stack pointer fault
ERROR	Insufficient workspace to read time series
ERROR	Time series date not found
ERROR	Insufficient variation in X-variable
ERROR	Member-list missing
ERROR	Identifier not found
ERROR	2-dimensional array
ERROR	Not enough stack for table
ERROR	Y and X lists different length
ERROR	Output not to PRINTER : PLOT ignored
ERROR	Y-variable and X-variable are incompatible
ERROR	DEFINE statement too long
ERROR	Missing definition
ERROR	Table fault
ERROR	FORMAT code error
ERROR	Member-list missing
ERROR	No terminal year

APPENDIX E

IDIOM - ERROR MESSAGES

ERROR	Identifier not found
ERROR	2-dimensional array
ERROR	Not enough stack for table
ERROR	Illegal decimal point
ERROR	Output not to printer or punch
ERROR	Not enough workspace to print table
ERROR	Wrong dimension for titles
ERROR	Titles not available

UPDATE routine errors

ERROR	Unrecognised function name
ERROR	Attempt to update function twice
ERROR	Logic fault
ERROR	Lag identifier not found
ERROR	Illegal QYC switch
ERROR	Illegal DC function
ERROR	UNIFORM specification not allowed for income group 1
ERROR	Illegal HR function
ERROR	Illegal FCM function
ERROR	Illegal price function
ERROR	Illegal price function
ERROR	Illegal combination of PQ and PY functions
ERROR	Illegal CM function
ERROR	Illegal UE function
ERROR	Illegal V switch
ERROR	Illegal YE function
ERROR	Illegal YS switch
ERROR	Illegal YV switch

COMPUTE routine errors

ERROR	Unrecognised function name
ERROR	Attempt to compute function twice
ERROR	Logic fault
ERROR	Logic fault
ERROR	Illegal QYC switch
ERROR	Illegal DC function
ERROR	Illegal HR function
ERROR	Illegal HR function
ERROR	Illegal ULC switch
ERROR	Illegal v.a. switch
ERROR	Illegal tax switch
ERROR	Illegal v.a. switch
ERROR	Illegal tax switch
ERROR	Illegal FCM function
ERROR	Illegal tax switch
ERROR	Illegal v.a. switch
ERROR	Illegal tax switch
ERROR	Illegal price function
ERROR	Illegal commodity function
ERROR	Illegal export price function
ERROR	Illegal tax switch
ERROR	Illegal v.a. switch
ERROR	Illegal tax switch
ERROR	Illegal commodity function
ERROR	Illegal price function
ERROR	Illegal M function

APPENDIX E

IDIOM - ERROR MESSAGES

ERROR	Negative output set to zero
ERROR	Negative imports set to zero
ERROR	Illegal DSC switch
ERROR	Illegal tax switch
ERROR	Illegal UE function
ERROR	Illegal V switch
ERROR	Illegal commodity function
ERROR	Illegal YE function
ERROR	Illegal YS switch
ERROR	Illegal YV switch

TABULATE routine errors

ERROR	Unrecognised function name
ERROR	Logic fault
ERROR	Illegal v.a. switch
ERROR	Illegal tax switch
ERROR	Illegal v.a. switch
ERROR	Illegal tax switch
ERROR	Workspace too small to print table

Utility routine errors

APPENDIX F

IDIOM - PREPFOCESSCR

```
SECTION "LREQV"
GET "LIBHDF"

GLOBAL $(
ID :      151
CH :      152

LENGTH:   154
PURGELINE:155
READLINE: 156
$)

LET PURGELINE() BE $(
CH:=RDCH() REPEATUNTIL CH = '*N'
$)

LET READLINE() = VALOF $(
LET P=?
LENGTH:= 1 ; ID!0 := CH
UNTIL CH = '*S' DO $( CH:=RDCH() ; ID!LENGTH:= CH ; LENGTH +:= 1 $)
READN() ; READN() ; READN() ;  M := READN()
PURGELINE()
RESULTIS ¬(M = 0 | M = 4)
$)

LET START(P) BE
  $(  LET INPUT = FINDINPUT("FROM")
      LET OUTPUT= FINDOUTPUT("IC")
      LET SYSPR = FINDOUTPUT("SYSPRINT")
      LET IDV = VEC 5
      LET FIRST = TRUE AND EQNEEDED = TRUE
      AND CURSOR =0 AND LINENUMBER = 1 AND NN = 1

      IF SYSPR = 0 WRITETOLOG("NO SYSPRINT*N") <> STOP(16)
      SELECTOUTPUT(SYSPR)
      IF OUTPUT = 0 WRITES("No output stream*N") <> STOP(16)
      IF INPUT = 0 WRITES("No input stream*N") <> STOP(16)
      SELECTOUTPUT(OUTPUT) ; SELECTINPUT(INPUT)
      ID:= IDV
      FOR K = 0 TO 3 ID!K := '*S'
      WRITES("C* EQUIVALENCE statement for symbol table TABLS*N")
   $( CH := RDCH()
      SWITCHON CH INTO
      $(

   CASE ENDSTREAMCH :UNLESS EQNEEDED NEWLINE() ; WRITES("      EQUIVALEN
                    FOR K = 1 TO 4 DO
                      $( IF K = 3 WRITES("*N     1    ")
                         WRITES("(L2") ; WRITED(K,1) ; WRITES(",TABLS(2
                         WRITEN(NN) ; WRITES(")"); ; NN+:= 1
                         UNLESS K = 4 WRCH(','); $)
                    FINISH

         CASE '*S':PURGELINE() ; ENDCASE

         DEFAULT  :IF READLINE() THEN $(
                   IF EQNEEDED THEN $( WRITES("      EQUIVALENCE ")
                                 EQNEEDED:=FALSE ; LINENUMBER:=1 ;FIRST
```

APPENDIX F

```
                    IDIOM - PREPROCESSOR
                         CURSOR := 18 $)
          TEST FIRST THEN FIRST:=FALSE ELSE WRCH(',')
          IF (LINENUMBER > 17 & CURSOR > 29) THEN EQNEEDED :=
          IF CURSOR > 51 THEN $( NEWLINE() ; WRITES("     1")
                               CURSOR := 7 ; LINENUMBER +:= 1 $)
          WRITES("(L") ; FOR K=0 TO LENGTH-1 DO
         $( WRCH(ID!K) ; ID!K:='*S' $)
          WRITES(",TABLS(2,") ; WRITEN(NN) ; WRITES("))") ; NN
          CURSOR +:= 22 ; IF EQNEEDED THEN NEWLINE() $)
          ENDCASE

    $) // End of SWITCHON
  $) REPEAT
$)
```