

# VORSIM Overview

VORSIM is a set of computer programs that helps build, operate, and manage mathematical models in the Microsoft Excel spreadsheet. Multi-equation simulation models can be constructed in a spreadsheet, however if they are large or complex it is much easier to deal with them using the VORSIM modeling system. With VORSIM, the user specifies the structure and equations for a model as always but VORSIM programs automate the most tedious parts of model building. VORSIM keeps track of variables and equations while retaining all of the convenience and power of spreadsheets. Operations are controlled with buttons placed on appropriate workbook sheets as they are constructed.

VORSIM is especially useful for the construction of large spreadsheet models with standardized equation systems. Once a model is constructed, equation parameters and equations can be quickly changed and re-initialized to fit the base data. Models can be rapidly assembled and can be repeatedly re-simulated for alternative scenarios. Solution values can be viewed immediately with Excel pivot tables and charts. Model components and solutions are accessible for viewing and are pre-formatted for printing. Since a VORSIM-built model is documented as it is constructed, a 'transparent' model and set of associated component and solution outputs can be readily transferred to others who were not involved in the model construction.

# Control, Construction, Operation

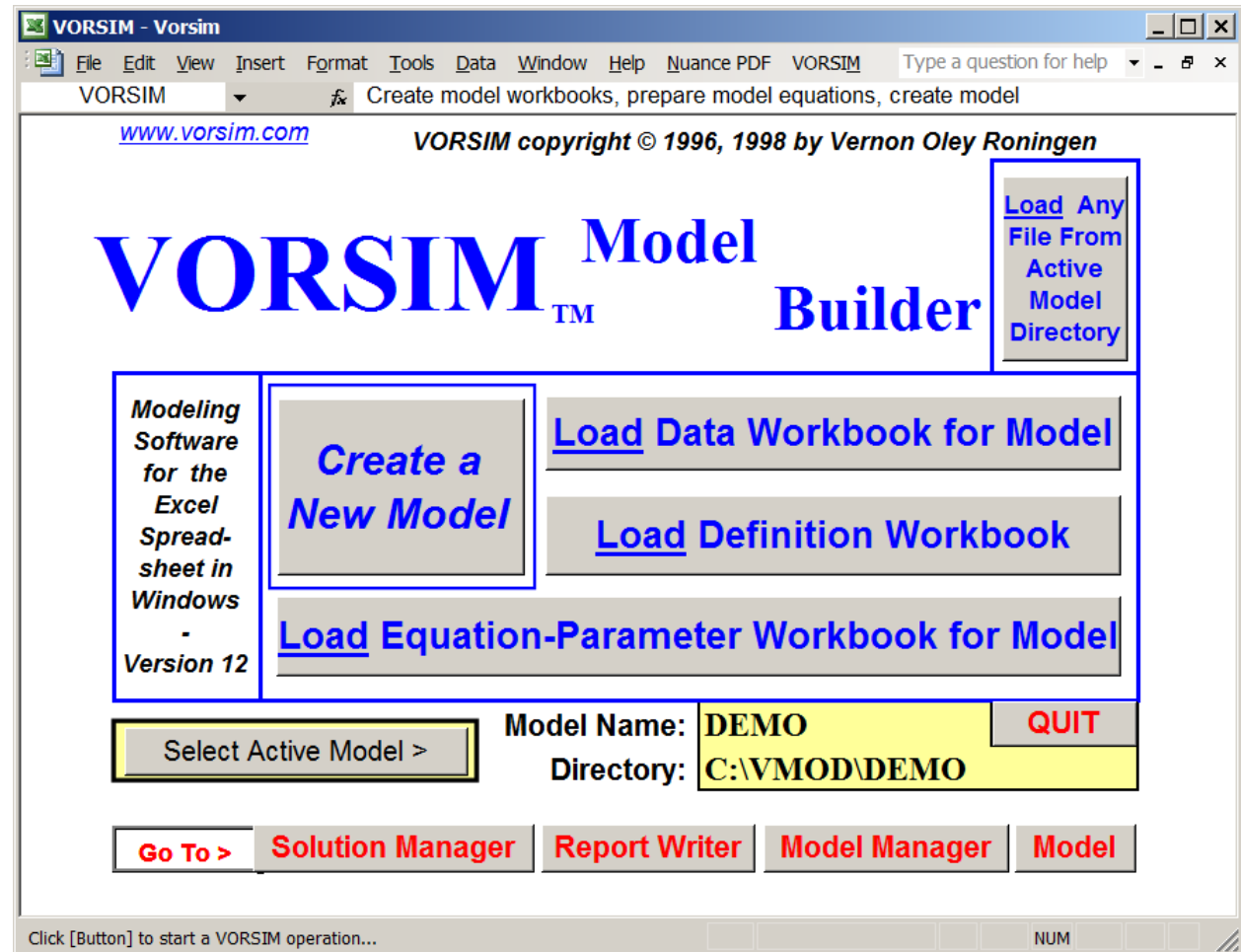
**Control:** VORSIM is a set of programs which perform modeling operations. Program operations are initiated from control screens. As model workbooks are built, buttons are placed on many workbook sheets to control operations relevant to that sheet. Control buttons have explanatory labels and are organized to follow model construction and other operations. The normal Excel spreadsheet menu is still available at all times.

**Construction:** There is a logical progression in the construction of a model in VORSIM. First the model is defined in terms of spreadsheet organization and an accompanying variable nomenclature. From the resulting model definition workbook, data and equation workbooks are created. Then data is entered into the data workbook and equations are constructed in the equation workbook. Many operations along the way are guided by menus that appear when an operation is begun. When model equations have been written, the model itself is created in an independent model workbook. The model is configured for solution and saved.

**Operation:** The model itself is simulated and operated from the model workbook. As solutions are generated, they can be saved using names selected by the user. After solutions have been saved, indicators (post solution calculations) can be added. Then solution and indicator values can be compared or can be used in the generation of user-built spreadsheet reports. In addition, graphic and other analytical programs allow immediate analysis of solution results.

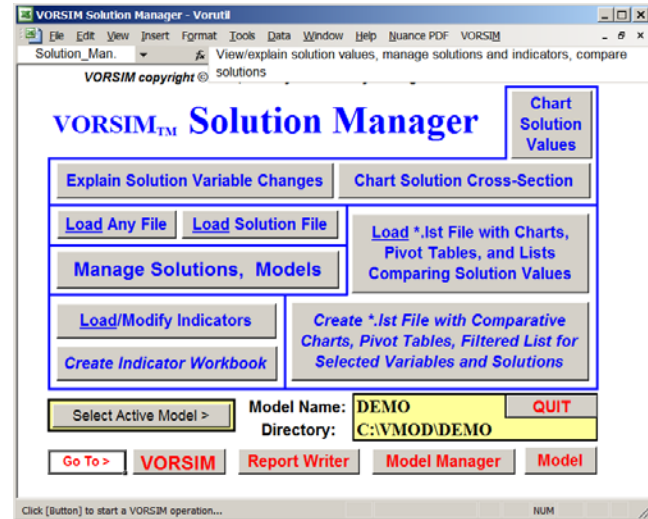
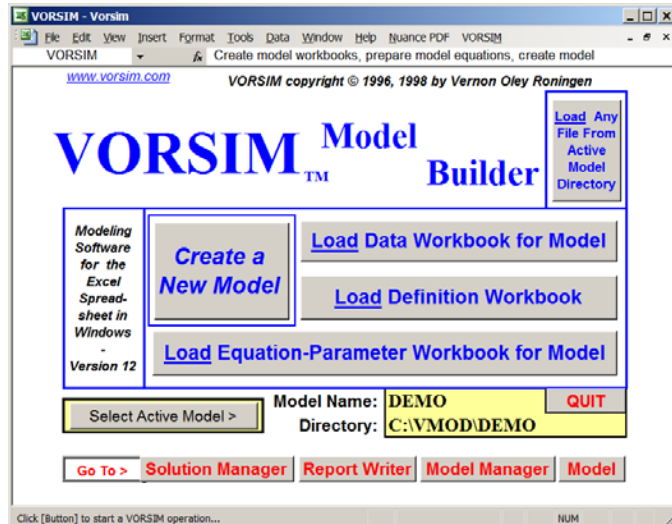
# A Control Screen

Operations are initiated from four control screens that are focused on (use) the model shown in the yellow area. Buttons with blue type initiate VORSIM modeling operations while buttons with red type close screens or workbooks or quit or transfer you to other control screens.



There are four control screens – VORSIM Model Builder (above), Solution Manager, Report Writer, and Model Manager. In addition, several utility programs which come with the system.

# The Four Screens



# Program Access

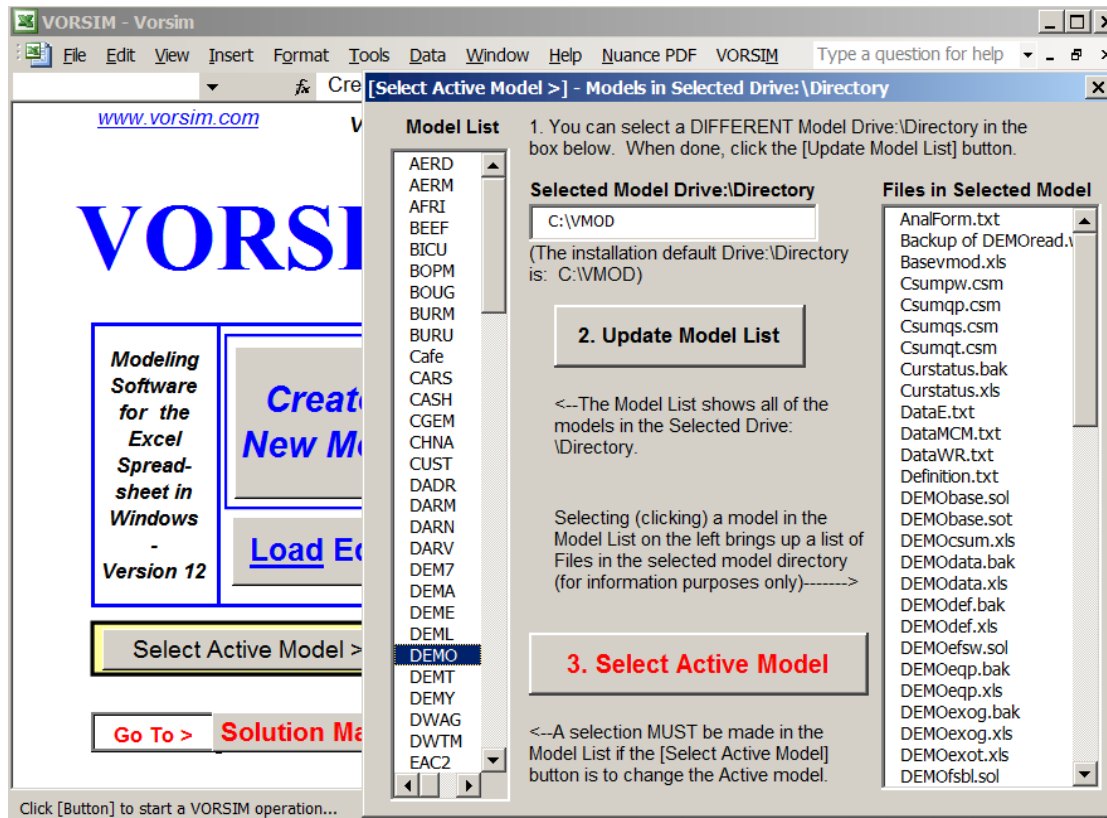


[VORSIM] – This icon loads the VORSIM Model Builder control screen (previous slide). This screen accesses the software for building models. Red buttons at the bottom of the screen close active files and transfer you to other control screens.



[VORSIM Utilities] – This icon opens a window that contains icons for accessing all four control screens as well as icons that bring up utility programs, VORSIM information, the VORSIM web site including the User's Guide and tutorials, and information about demonstration models.

# Model Access



Model Access – Models are all contained in a directory called VMOD. Each model (including all associated files) is kept in a folder with the model name (for example, the model DEMO might be kept in the folder C:\VMOD\DEMO). Models are accessed by the [Select Active Model >] button on a control screen (you select a model from all those shown and it becomes the active model). When a model is constructed, all of the programs necessary for its simulation and for saving solutions are attached to it, making the model workbook independent of VORSIM controls.

# Model Organization

VORSIM uses the structure of the Excel workbook for model organization. The largest model unit is the (workbook) Sheet. Sheets may represent countries or commodities or any other major organizational unit of a model. The next organizational level is the are subdivisions within Sheets which VORSIM labels Categories. For example, if a Sheet represents a country, a Category may represent a product in the country.

The key model unit is the Variable which may be unique for each Sheet or for each Category. Variables are given a code that identifies their coverage. Model variable **names** are created from a concatenation of codes for Sheets, Variables, and Categories. Model variables are placed in specific cells when a model is created (this means that with VORSIM you always know where a variable is located). By imposing this model organization and associated nomenclature, VORSIM keeps track of all model variables at all times while using codes defined by the user.

There are many component files (workbooks) which are created along with the model workbook itself. These include files for the model definition, model data, model equations, model indicators, solutions, reports, and backups of key files. The construction of a model is a sequenced creation of the workbooks leading up to the final model. Model simulations and reports involve the creation of additional workbooks.

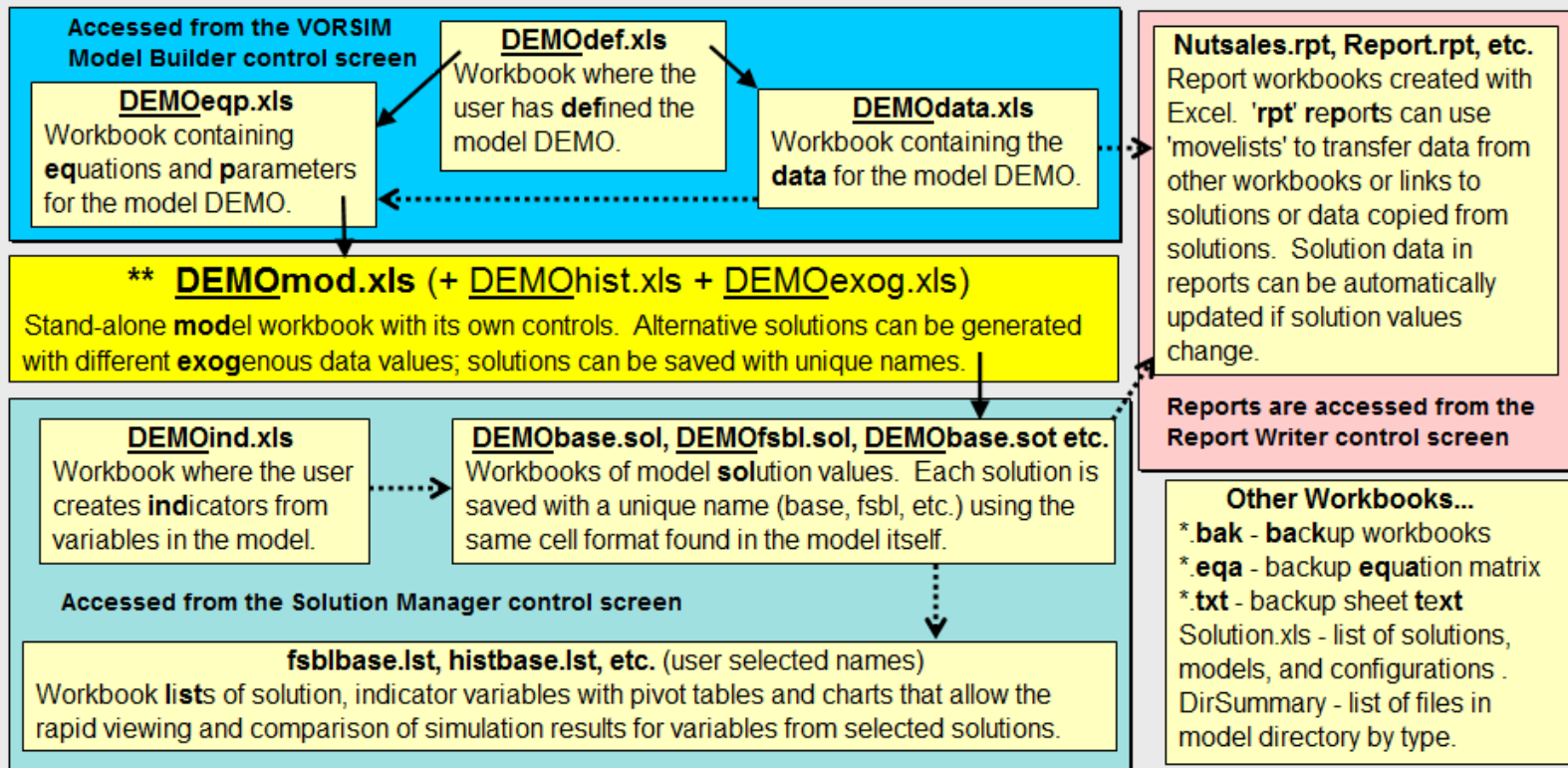
The next slide shows the workbook family associated with the model DEMO along with the order of their creation. There are also many other files residing in the DEMO folder that are created during model building and simulation operations.

# Model File Structure

## Workbook (File) Structure for a VORSIM-Built Model (Examples From The Demonstration Model DEMO)

Solid arrows show the workbook that creates another file, dotted arrows show the movement of data/formulas between workbooks.

All files associated with a model including the model itself, model building blocks, solutions, reports, and backup files are kept in a folder with the same name as the model, e.g. C:\VMOD\DEMO. Key files contain the model name DEMO.



\*\* The model (DEMOmod) can be accessed from any control screen (VORSIM Model Builder, Solution Manager, Report Writer, or Model Manager).

# Naming Conventions

In order to keep models fairly simple and to facilitate the programming of VORSIM, brief but flexible naming conventions are used for models and model variables. Model names are confined to 4 letters and the model files are stored in a folder with the model name.

Model variable names are more complex. They are made up of user defined codes which represent the organization of the model as well as the types of variable used. A model variable name includes the Variable type as well as the Sheet and/or Category which it is associated with. This is all done when the model definition workbook is created. Model variables names so constructed can apply to the model as a whole or they can be associated with major (Sheets) or major and minor (Categories) units in a model. However constructed, each variable is unique in a model and occupies its own cell in the model and associated workbooks. Since the cell location of all model variables and their nomenclature is set, the model builder always knows where each variable is located in the model and associated workbooks.

The nomenclature for model variables is used in the equation writing process so that equations are readable in terms of user defined codes and Excel math notation.. For efficiency purposes, equations are converted to a simpler notation in the model itself, substituting model variable cell locations for model variable names.

The nomenclature used in VORSIM models is illustrated in the next slide for the model DEMO.

Codes for the model DEMO

DEMO
Sheet
E
WR
MCM
Category
BL
NUT
W
Variable
fs
pw
qp
qs
new
t

## Variable Code Nomenclature and Model Structure

- Sheet** The MAJOR division of information in a VORSIM model in the Excel workbook structure. Model component workbooks are organized by Sheets, coded as 1-5 digit upper case letters, e.g. **E**, **WR**, and **MCM**.
- Category** Subdivisions of model information within a Sheet. The Category code allows the same subdivision set to be used in multiple sheets. The Category is coded as 1-5 digit upper case letters, e.g. **BL**, **NUT**, **W**.
- Variable** Variables are organized by their type of association with Sheets and Categories. Variable type codes can be from 1-5 lowercase letters. 2 digit codes are Category specific (a variable with a 2 digit code is created for each category) e.g. **qs**; 1 and 3-5 digit codes are Sheet specific (one variable created for each Sheet), e.g. **t** or **new**.



A **full variable name** is a concatenation of Sheet, Variable, and Category codes, e.g. **EqsBL**, **Enew**, **WRqsNUT**. The full variable names are created from the Sheet, Variable, and Category codes during the creation of the model definition workbook. A 2 digit **qs** code would result in variables with the Sheet code in front and the Category code behind, such as **EqsBL** or **WRqsBL**. Codes of other length such as **t** or **new** would result in variables such as **MCMt**, **Enew**, and **WRnew**. A list of some variables for a model (e.g. DEMO) might be the following where **E** and **WR** Sheets have **qs** type variables for each Category but only one variable each for **new** type variables.

E sheet

<b>EqsBL</b>
<b>EqsNUT</b>
<b>EqsW</b>
<b>Enew</b>

WR sheet

<b>WRqsBL</b>
<b>WRqsNUT</b>
<b>WRqsW</b>
<b>WRnew</b>

# Nomenclature

# Model Equations

Equations come in three forms in model workbooks. The model itself contains equations constructed from Excel math and cell notation – the most efficient form of an equation from the spreadsheet viewpoint. However when equations are constructed in the equation workbook, they are written using model variable names rather than cell locations. This provides readable documentation of model equations.

The model equations themselves are created from equation matrices where matrix columns represent equation terms and matrix rows represent categories represented by the equation. Creating an equation (or equations) begins with the construction of the equations matrices, from which the equations are generated. Equation banking procedures bank these equation matrices. There are several reasons for the use of equation matrices to generate equations rather than create all individual equations themselves. First, by putting equation parameters in a cell, it is simple to change an equation simply by changing a matrix parameter in that cell and re-generating the equations; one does not have to manually edit a parameter in a long complicated equation. Second, it is easy to use an equation matrix to generate a family of equations where the equation structures are similar but only the base data and/or equation parameters differ. Of course a one-row matrix can be used to generate one unique equation, but for models with many equations with similar structures, the matrix generation approach saves enormous amounts of model building time. Third, the matrices lead users to think of terms in equations and about the grouping and organization of model equations. Similar equations are grouped together in a model. The next slide shows how an equation matrix leads to a set of model equations.

# Equation Matrix

quantity sold	shift	bolts	nuts	washers	new constr.
<b>qs</b>	*(1+ fs )	*!pwBL^#	*!pwNUT^#	*!pwW^#	*!new^#
<b>BL</b>	-	-0.89	0.02	0.01	1.02
<b>NUT</b>	-	0.04	-0.78	0.07	0.89
<b>W</b>	-	0.05	0.17	-0.83	0.98

EqsBL	2.266483*(1+EfsBL)*EpwBL^.89*EpwNUT^.02*EpwW^.01*Enew^1.02
EqsNUT	2.81267*(1+EfsNUT)*EpwBL^.04*EpwNUT^.78*EpwW^.07*Enew^.89
EqsW	0.8653203*(1+EfsW)*EpwBL^.05*EpwNUT^.17*EpwW^-.83*Enew^.98

The equation matrix at the top holds all information needed to generate the equations shown below. Each (yellow) column head contains a term of the equation. Each row contains parameters that replace the crosshatch (#) in the term for that column. Each row head (yellow) gives the Category or single code for the dependent variable of the equations, e.g. this matrix is in the sheet E so the first row contains items that will go into the terms explaining the variable EqsBL. The blue cell gives the code for the variable explained. In any equation term, ! is replaced by the sheet code when the term is written into the equation. The | is replaced by the category code (row head) and a crosshatch is replaced by the number in the cell. The precision of the number placed in the equation is determined by the format of the cell – this can be changed by the model builder. The cells above the yellow column heads are used as documentation for each term in an equation. When an equation is put in the equation ‘bank’, it is this equation generating matrix that is banked rather than the equation itself. An empty cell would mean a term is not included in an equation.

# Key Model Workbooks

**Model Definition Workbook** – This is the first workbook created in the model building process. Here the model structure and codes are defined and documented. The second sheet of the workbook contains all of the model variable names in the cells where they will be located in the model.

**Data Workbook** – The data workbook is created from the model definition workbook using the model variable names defined therein. The data workbook will be filled with all of the data used in the model building and simulation operations.

**Equation Workbook** – This workbook is where model equations are constructed and documented. The model itself is generated from this workbook after all of the equations have been entered. Buttons and menus in the workbook provide tools for constructing and editing model equations. Most of the work involved in model-building takes place here.

**Model Workbook** – A model built with VORSIM resides in an independent workbook which contains all of the software necessary for configuration, simulation, and the saving of simulation results. When a model is loaded, associated workbooks of exogenous and/or historical data needed for simulation are also loaded. Alternative models can be constructed as well and each of these will reside with its own unique name in the model folder.

If data or equations or equation parameters are changed in the data or equation workbooks, the model workbook can quickly be regenerated and reconfigured. Samples of these workbooks for the model DEMO are shown in the following slides.

# Definition Workbook

**VORSIM - DEMOdef**

File Edit View Insert Format Tools Data Window Help VORSIM

A2      DEMOnstration model

DEMO    <--Model    C:\VMOD\    Begin: 1990    Base: 1996    End: 2000

DEMOstr    name    Drive:\Directory\-----    -----Time Periods for Data and Model-----

**Sheet**

E	-	East
WR	-	Western-Region
MCM	-	Market-Clearing-Mechanism

**Category**

BL	-	BoLTs
NUT	-	NUTs
W	-	Washers

**Variable**

Variable	Format	Lag	Description.....
fs	0.00	0	shift-(unexpected change in)-sales(%)
pw	0.00	2	price-wholesale(\$/lb)
qp	0	0	quantity-produced(1000 lb)
qs	0	0	quantity-sold(1000 lb){if MC = market-clearing-variable}
new	0	0	new-construction(index,1990=100)
t	0	0	time

**Description.....**

A VORSIM model uses codes for major units called Sheets, sub-units called Categories, and Variables. These codes concatenate to make up variable names. Model design begins with the selection of these codes.

**Description.....**

The model definition workbook (here <DEMOdef.xls>) contained the model codes and their explanations on the {Definition} sheet as well as the number of lags required for any variables and Excel format codes for variables. Codes, formats and lags are all set on this sheet.

All of the variable names and locations for the model are contained in the {Variables} sheet.

Definition / Variables

Click [Button] to start a VORSIM operation...      NUM

# Data Workbook

The screenshot shows a software window titled "VORSIM - DEMOdata". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The active cell is A1, containing the formula =E variable. The data table below shows values for various variables from 1990 to 2000. The status bar at the bottom indicates "Click [Button] to start a VORSIM operation..." and "NUM".

E variable	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000
EfsBL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
EfsNUT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
EfsW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
EpwBL	0.30	0.31	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39	0.40
EpwNUT	0.19	0.20	0.21	0.22	0.23	0.24	0.25	0.26	0.27	0.28	0.29
EpwW	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.15	0.16	0.17
EqpBL	1000	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
EqpNUT	20	21	22	23	24	25	26	27	28	29	30
EqpW	200	210	220	230	240	250	260	270	280	290	300
EqsBL	500	510	520	530	540	550	660	570	580	590	600
EqsNUT	400	410	420	430	440	450	460	470	480	490	500
EqsW	300	310	320	330	340	350	360	370	380	390	400
Enew	100	102	104	106	108	110	112	114	116	118	120



# Model Workbook

**VORSIM - DEMOmod**

File Edit View Insert Format Tools Data Window Help

B9  $=2545.688*\$B\$6^0.33*1!\$B\$6^0.13*2!\$B\$6^0.03*(1+\$D\$16)^0.02$

1996 3. Install Solver Test Solver Solve DEMO model Save Solution **QUIT** Change Exogenous Variable View Solution Variable

DEMO	E	WR	MCM
fsBL	0.00	0.00	
fsNUT	0.00	0.00	
fsW	0.00	0.00	
pwBL	0.36	0.36	0.36
pwNUT	0.25	0.25	0.25
pwW	0.13	0.13	0.13
qpBL	1600	210	
qpNUT	26	970	
qpW	260	930	
qsBL	660	1150	0
qsNUT	460	536	0
qsW	360	830	0
new	112	113	
t			7

Yellow cells contain equations. White cells contain exogenous data.

Click the [Use previous configuration] button to configure the model. Then click the [3. Install Solver] button. Upon completion, change cells will be colored green and target cells, red. The model will have been saved along with the configuration.

This box is used only with sample models. Normally you will have to click buttons 1, 2, and 3 to initialize your model the first time. However once a model has been initialized once, if it is re-created again, you have the alternative option of clicking the [Use previous configuration] button to re-configure the model.

1. Select numbers (white cells)

2. Select formulas (yellow) to drive to zero

Option: Select instrument (white) and formula (yellow) cells  $\geq 0$

Use previous configuration

Model Solver parameters Current solution status

Click [Button] to start a VORSIM operation...

NUM

Buttons in the top row control model operations while buttons at the bottom control model configuration options.

# Solving a Model

VORSIM models make use of the “circular reference” feature in Excel. A model of simultaneous equations is a circular reference problem in Excel jargon. With VORSIM, rather than treating circular references as a problem, an Excel solver is added to the model to resolve the circular references in an iterative manner. To do this, the model is configured. Variables are chosen which can change with each iteration to drive target equations to zero values. These target equations may be, for example, market clearing conditions and the change variable might be prices. The selection of target variables and equations and the change variables are part of model design; in effect the method of model closure.

After a model workbook has been created, it is configured for solution and saved. Then alternative scenarios can be run by changing exogenous variable values. This is done directly in the model for a static (one time period) model and done with a menu for exogenous variables in a dynamic (more than one time period) model. The configuration is saved and can be easily reused if the model is re-created for any reason.

When a configured model is simulated, the user can immediately examine the results and/or the solution results can be saved under a user defined name. Once saved, the user can proceed with other simulations. A simulation is saved in numerical form as a workbook using the same layout as the model; so solution values are in the same cells as the equations that generated them. The color coding of the model follows into solution workbooks. Indicators (post solution calculations from solution values) can be added to solution workbooks after they have been saved. Following slides show the DEMO model configured for solution and a solution workbook generated from a DEMO solution.

Here the model DEMO has been configured for solution with solution 'change' cells marked in green and 'target' cells driven to zero marked in red. The model can be solved by clicking the [Solve DEMO model] button. A solution is saved with the [Save Solution] button.

The screenshot shows the VORSIM - DEMOmod software interface. The main window displays a spreadsheet with the following data:

DEMO	E	WR	MCM
fsBL	0.00	0.00	
fsNUT	0.00	0.00	
fsW	0.00	0.00	
pwBL	0.36	0.36	0.36
pwNUT	0.25	0.25	0.25
pwW	0.13	0.13	0.13
qpBL	1600	210	
qpNUT	26	970	
qpW	260	930	
qsBL	660	1150	0
qsNUT	460	536	0
qsW	360	830	0
new	112	113	
t			7

Below the spreadsheet, the software provides instructions:

- Green Cells - variables changed by the Solver to solve model (drive values in red cells to zero)  
 $\$D\$6:\$D\$8$
- Red cells - equation values driven to zero  
 $\$D\$12$   
 $\$D\$13$   
 $\$D\$14$

The interface also features a toolbar with buttons: Test Solver, Solve DEMO model, Save Solution, QUIT, Change Exogenous Variable, and View Solution Variable. A large blue text overlay reads "Configured Model".

# Configured Model

After the model has been configured and the solver installed, the configuration buttons are automatically removed.

# Solution Workbook

The solution workbook is repeated for each time period in the same format as the model sheet itself. Cell colors indicate the source of the numbers (e.g. the 1600 in the yellow cell means that this number was generated by an equation). In addition to the model solution values, indicators in rows and columns have been added to the model values (light green row and column headings). Each solution file is saved with a unique name (here fsbl).

VORSIM Solution Manager - DEMOfsbl.sol

File Edit View Insert Format Tools Data Window Help VORSIM

B9 1600

	1996	exogenous shift in sal			1990	2000
fsbl	E	WR	MCM	AR		
DEMO						
fsBL	0.00	0.00				
fsNUT	0.00	0.00				
fsW	0.00	0.00				
pwBL	0.36	0.36	0.36			
pwNUT	0.25	0.25	0.25			
pwW	0.13	0.13	0.13			
qpBL	1600	210			1810	
qpNUT	26	970			996	
qpW	260	930			1190	
qsBL	660	1150	0			
qsNUT	460	536	0			
qsW	360	830	0			
new	112	113				
t			7			
qpAP	1886	2110			3996	
pwAP	0.33	0.21			0.26	
shBL	88	12				
shNUT	3	97				
shW	22	78				

1996 / 1997 / 1998 / 1999 / 2000

Click [Button] to start a VORSIM operation... NUM

# Simulation Results

Selected solution values can be shown in charts immediately from the model workbook after a simulation is completed and before it is saved in a solution workbook. Once solutions have been saved, VORSIM provides many ways of viewing results and preparing reports.

Once solutions (simulation results) have been saved, they can be directly examined. The user can load and examine solution workbooks that have been saved. Alternatively, solution charting routines available on the Solution Manager control screen can be used to view results.

The quickest way to compare saved solution results is to use the Pivot Table option on the Solution Manager control screen. A menu driven operation allows the user to gather selected variables from selected solutions into a pivot table and chart framework. This fast automatic routine allows the immediate comparison of selected solution values in tables and charts. In addition, selected subsets of solution values can be viewed.

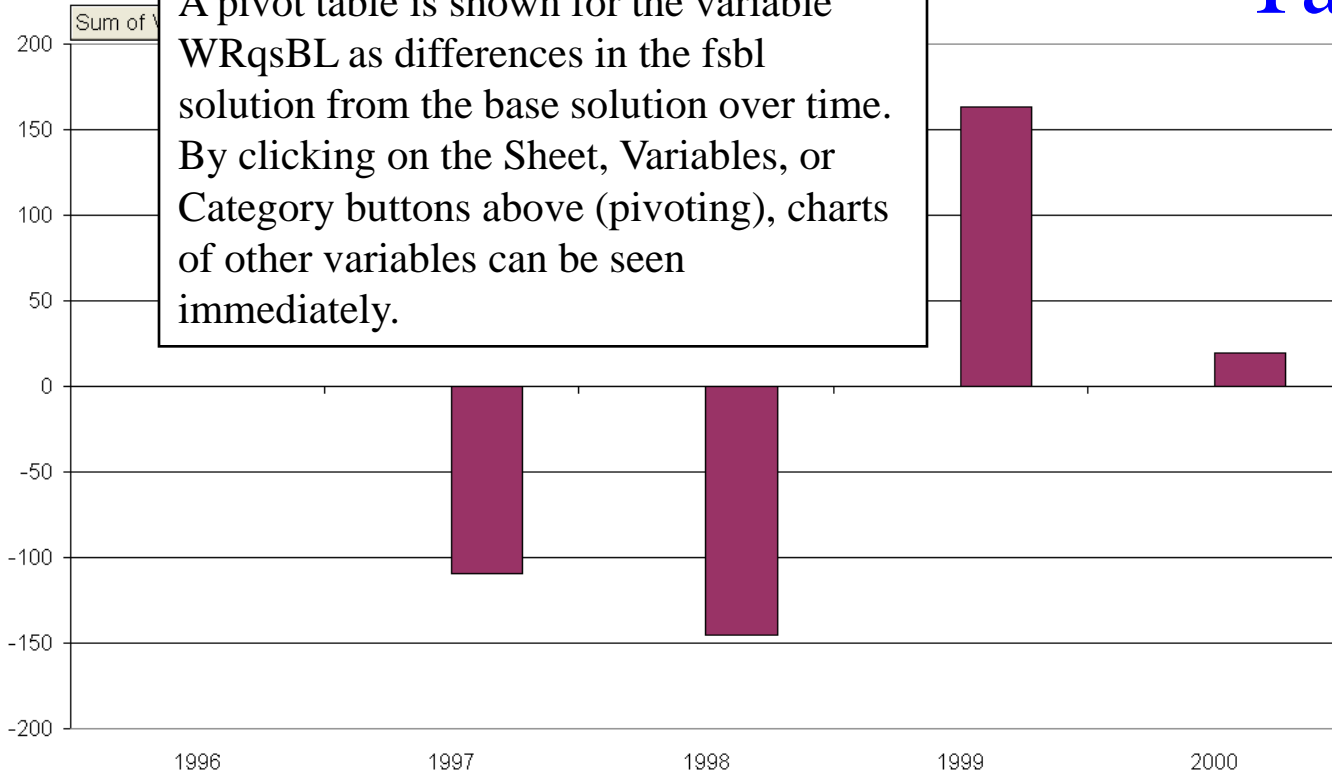
Finally, the Report Writer control screen contains programs which allow the users to create their own reports in Excel with several methods for moving solution values into these customized reports. While normal Excel formatting and other facilities are used for report creation, VORSIM provides for easy movement of solution values into reports. A 'movelist' option can be created where the user defines a list that uses an automatic procedure for moving solution values into their chosen places in a report. The user can create a report with direct links between chosen report cells and the source values in solution files. Finally, sets of solution data can be moved into reports using the VORSIM pull-down menu in the Report Writer. Once solution data is in a report, it can be automatically updated if solutions change.

Sheet WR Variable qs Category BL

# Pivot Table

DEMO > Difference From > base

A pivot table is shown for the variable WRqsBL as differences in the fsbl solution from the base solution over time. By clicking on the Sheet, Variables, or Category buttons above (pivoting), charts of other variables can be seen immediately.



Solution  
 base  
 fsbl

Time

# Summary

VORSIM is a powerful modeling tool for the spreadsheet environment. It helps a user organize and operate large models in the Excel spreadsheet. Models and associated workbooks built with the system are all open to all Excel operations – what VORSIM does not do can usually be done by Excel. VORSIM makes good use of Excel capabilities by automating most of the tedious spreadsheet work needed in the process of building and operating a model but retaining all of the Excel spreadsheet capabilities for everything else.

A User's Guide and tutorials are found on the VORSIM website [www.vorsim.com](http://www.vorsim.com). By using the VORSIM model building system, you can concentrate your valuable time on designing your model and examining the solution results rather than dealing with the mechanics of creating and running a large model in the spreadsheet environment.



**[www.vorsim.com](http://www.vorsim.com)**