

EIA Model Documentation

**Macroeconomic Activity Module (MAM):
Kernel Regression Documentation
of the
National Energy Modeling System**

October 15, 1998

Office of Integrated Analysis and Forecasting
Energy Information Administration
U.S. Department of Energy
Washington, DC

MACROECONOMIC ACTIVITY MODULE (MAM): KERNEL REGRESSION DOCUMENTATION

Final Report

Subtask 3

October 15, 1998

Prepared For:

Energy Information Administration
1000 Independence Avenue, S.W.
Washington, DC 20585

By:

Decision Analysis Corporation of Virginia
8009 Snowpine Way, Suite 100
McLean, VA 22102

Contract No. DE-AM01-98EI
Task 90203

DAC

Decision Analysis Corporation of Virginia
8009 Snowpine Way ■ Suite 100 ■ McLean, Virginia 22102 ■ (703)893-5087

NOTICE

This report was prepared as an account of work sponsored by the United States. Neither the United States nor the United States Department of Energy, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents its use which would infringe on privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, make, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

**This Page is
Intentionally
Left Blank**

Macroeconomic Activity Module: An Introduction to Kernel Regression

Background

The Macroeconomic Activity Module (MAM) serves two functions within the National Energy Modeling System (NEMS). First, it provides consistent sets of baseline macroeconomic variables (GDP and components, aggregate prices, interest rates, industrial output, housing starts, commercial floorspace, new car sales, etc.) which are used by the supply, demand and conversion modules in reaching an energy market equilibrium. Second, it is designed to provide a feedback mechanism that alters the baseline variables during the course of an integrated NEMS run.

For the reasons enumerated in the Component Design Report, the MAM is not a structural model. Instead, the consistent sets of baseline variables are generated by running the full slate of DRI models (macroeconomic, input-output, employment, and regional). The macroeconomic variables required by NEMS are extracted and uploaded to the RISC platform. In order to provide the feedback mechanism within NEMS, numerous simulations of the DRI models are completed on a personal computer, and the solutions saved. A simplified representation of the relationship between the important inputs (provided by the NEMS) and the required outputs (provided back to NEMS) is then constructed.

Until recently (through the AEO98 version), MAM relied upon a response surface version of the DRI US Quarterly model to incorporate energy price feedback into the macro economy. The response surface version of the model is an approximation of a larger model and is estimated on “pseudo data” generated by repeated simulations of the larger model. Energy prices, in particular the oil price, were changed in the DRI model and results of many simulations were saved in a database. Linear equations were estimated that related a particular variable (such as percent changes in GDP) to percent changes in energy prices and lagged dependent variables in order to mimic the response of DRI’s Quarterly model to changes in energy prices.

This response surface model worked relatively well when asked what the macroeconomic impacts were of changing energy prices when resource costs changed. The model was fast and relatively easy to maintain. However, in situations where energy prices were changing because of some policy tool, such as taxes or implementing an auctioned permit system, the results from the response surface version of the model became increasingly different from the results obtained from the larger DRI model. In these instances, energy end-use prices would change by far more than the underlying energy resource costs. As a result, EIA began to search for alternate ways of capturing the response of the DRI model that would be useful in both a energy policy scenario as well as a situation that involved projecting different energy prices.

Prior to the Energy Information Administration’s carbon stabilization analysisⁱ, the response surface model contained in the National submodule of the Macroeconomic Activity Module was replaced with a kernel regression model. The performance of the kernel regression model, in terms of mimicking the DRI US model, was judged superior in a variety of policy settings to that of the response surface model. As a result, the kernel regression has become, for now, the national macroeconomic model of the National Energy Modeling System. The kernel regression model has since been used to produce forecasts reported in the *1999 Annual Energy Outlook*.

Simplified Kernel Regression Example

Kernel regression is a nonparametric approach to regression analysis that makes no assumptions regarding the functional form of the unknown function m in the following relationship:

$$Y_i = m(X_i) + \varepsilon_i$$

The Y_i ’s are observed random variables, the X_i ’s are known constants and ε_i ’s are independent random errors with mean zero and constant variance. In the Macroeconomic Activity Module, the Y ’s are the one hundred and nineteen national and regional macroeconomic concepts that the module forecastsⁱⁱ. The X ’s are twelve inputs from the National Energy Modeling System. These inputs include tax collections and energy prices and quantities demandedⁱⁱⁱ. To approximate the unknown function m , the kernel regression

relies upon databases containing simulation results of the DRI US model. The simulations represent different assumptions about the price of crude oil and about policy instruments. The databases are linked so that the input variables (i.e. tax collections and energy prices and quantities) from each simulation are associated with their respective output variables (i.e. macroeconomic concepts.)

Policy questions are much more complicated in terms of the assumptions that must be made when running the larger DRI model. Consequently, constructing a surface that approximates the multiple dimensions becomes extremely complicated. Rather than attempt to uniformly define the space, we have chosen to describe the neighborhoods of likely input combinations more completely. This is both good and bad. Within the defined neighborhoods the approximation of the full DRI model solution is much more precise; but when the energy results in a particular NEMS simulation fall near the edge of a neighborhood, the kernel results are suspect.

The success of this kernel regression procedure relies upon incorporating simulations that will actually be part of the energy policies analyzed (if using the kernel regression model in a policy mode) or the path of energy prices (if using the kernel model in an energy resource cost mode). In creating the database used for the carbon constraint study, repeated NEMS simulations of alternative carbon price scenarios were needed in order to better calibrate the response of the larger DRI model to changing energy prices and quantities predicted by the energy models in NEMS. If the analyzed NEMS scenarios have different starting times or alternative paths to attain carbon targets, then the DRI model had to be simulated again using the new starting times or paths to carbon attainment.

One major concern for both the response surface version of MAM using linear estimation techniques and the kernel regression technique is how close both replicate the larger DRI model. An example is provided that compares the results using a kernel regression approach to a linear regression. This illustration, while a gross oversimplification of the application of the methodology in the MAM, does show the basics of performing kernel regression. Analogies to the MAM are made where possible. Table 1 contains data for an independent variable, X, and a dependent variable, Y.

Table 1: Data for Regression Analysis.

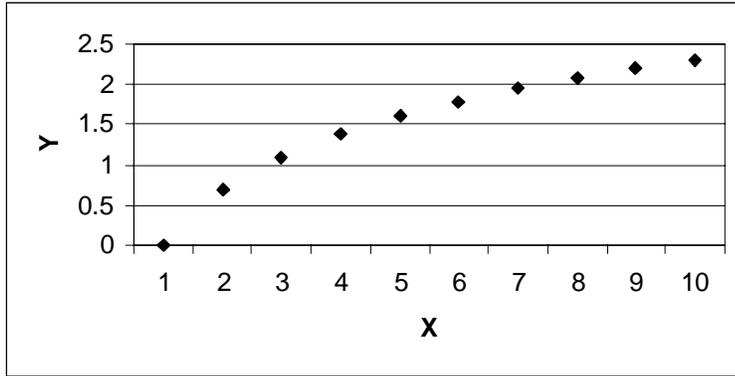
X	Y
1	0
2	0.693147
3	1.098612
4	1.386294
5	1.609438
6	1.791759
7	1.94591
8	2.079442
9	2.197225
10	2.302585

The input and output variable databases in MAM are analogous to the data in Table 1. The explanatory variables, X, in the MAM include tax collections and energy prices and quantities demanded. In the MAM, the response variables, Y, are all the national and regional macroeconomic variables forecasted by the module. The input databases contain samples of simulations that differ in both the type of policy and its degree of implementation. Results of simulations based upon these assumptions are placed in the output databases. The input and output databases are linked so that levels of tax collections and of energy prices and quantities demanded are related to their respective levels of the national and regional macroeconomic variables.

The data points in Table 1 are plotted in Figure 1. Ordinarily, the function m describing the relationship between Y and X is unknown. In this instance, the function m is known because the data for this illustration was invented. The regression equation is:

$$Y_i = m(X_i) + e_i = \ln(X_i)$$

Figure 1: Plot of Data Points X and Y.

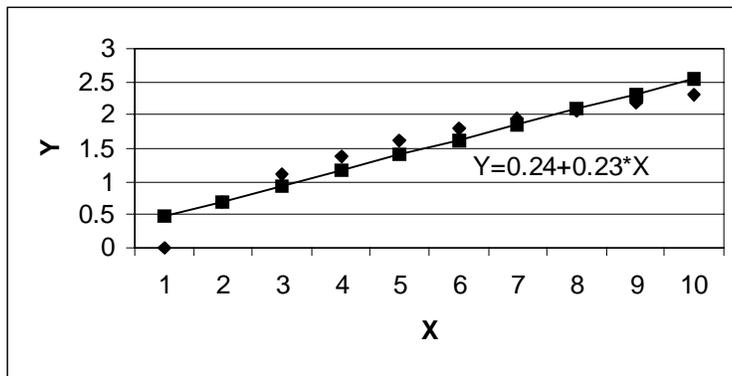


We will proceed with this illustration as though the function m is unknown and use regression analysis to approximate the function $\ln(X_i)$. The specific question this illustration answers is what is the value of Y given X equals 4.5. Since the function m is known, we know the true answer is 1.5. We proceed now to see what answer a parametric and a nonparametric approach would yield assuming m is not known.

Figure 2 contains a plot of the X and Y data points along with the regression line estimated using ordinary least squares^{iv}. In order to use ordinary least squares, we must specify a functional form and make assumptions about the behavior of the residuals. Given this approximation to the function m , the value of Y given that X equals 4.5 is:

$$Y(X) = \alpha + \beta X$$
$$Y(4.5) = 0.24 + 0.23(4.5) = 1.28$$

Figure 2: Ordinary Least Squares Regression of Y on X.



The steps to solve the kernel estimator given X equals 4.5 are now given in more detail than that for the ordinary least squares estimator. We begin by normalizing values of the explanatory variable, X , using its mean, 5.5. Table 2 contains these values.

Table 2: Data for Regression Analysis.

X	Normalized X (X_i/5.5)
1	0.181818
2	0.363636
3	0.545455
4	0.727273
5	0.909091
6	1.090909
7	1.272727
8	1.454545
9	1.636364
10	1.818182

This same transformation is done for all the inputs in the selected simulations contained in the input databases during a NEMS run. The values of tax collections and energy prices and quantities demanded supplied by NEMS are also transformed using the means of the inputs contained in the databases. In this example, the input value is 4.5 and its normalized value is 0.818182. While normalizing the input data is not necessary, the conversion to multiples of average units makes analysis of the input data easier particularly when there are a number of inputs measured in different units as in NEMS.

The kernel estimator computes for a function of X a weighted average of observations close to the i^{th} value of X. The kernel estimator used in the MAM is the Nadaraya-Watson estimator. This same nonparametric estimator is used in this illustration to approximate the function m . The Nadaraya-Watson estimator is:

$$m_h(X) = \frac{n^{-1} \sum K_h(X - X_i) Y_i}{n^{-1} \sum K_h(X - X_i)}$$

where X is the normalized value of 4.5, n is the sample size 10, Y_i is the values of the observed response variable and $K_h(u)$ is the kernel with a scale factor h . The scale factor is also known as the bandwidth. The bandwidth determines the size of the weight sequence about the i^{th} value of X . The formula for the kernel, $K_h(u)$ is:

$$K_h(u) = h^{-1} * 0.75 * (1 - (u)^2) \text{ for } |u| \leq 1, \text{ else} \\ K_h(u) = 0$$

Where $u = (X - X_i)/h$ and $h = \text{a scale factor}$

“ u ” is the scaled difference between X , the normalized value of 4.5, and that of the normalized values of the explanatory variable, X_i . The kernel determines the shape of the weights. The kernel used in MAM is known as the Epanechnikov kernel. This kernel is of parabolic shape. It is one of several kernels that could be used in an estimator. It was selected because it is considered “optimal” in terms of minimizing certain measures of error. Actually, the choice of a kernel is not as important as is the selection of a bandwidth. The value of the bandwidth, h , is arbitrarily selected as 0.5 in this illustration. In the MAM, the bandwidth is computed as a function of the sample size and the standard deviation of the explanatory variables. Table 3 contains the values of u and of the kernel for X_i .

Table 3: Computed Values of the Scaled Differences From the Normalized Value of 4.5 and of the Kernel.

X	Normalized X	U ((4.5/5.5)- X _i)/h	Kernel
1	0.181818	1.272727	0
2	0.363636	0.909091	0.260331
3	0.545455	0.545455	1.053719
4	0.727273	0.181818	1.450413
5	0.909091	-0.18182	1.450413
6	1.090909	-0.54545	1.053719
7	1.272727	-0.90909	0.260331
8	1.454545	-1.27273	0
9	1.636364	-1.63636	0
10	1.818182	-2	0

All the data required by the Nadaraya-Watson estimator has now been computed. Plugging it in and solving yields the nonparametric solution for the function m given that X is 4.5:

$$m_h(4.5/5.5) = 1.460992$$

In this illustration, the nonparametric estimator yielded a solution closer to the actual value of 1.5 as compared to that of the ordinary least squares estimator, 1.28. Nonparametric regression estimators are more robust in regression analysis problems given uncertainties about the functional form of relationships and given violations in assumptions regarding the least squares residuals.

Database Description

Before the kernel regression version of MAM was developed, macroeconomic analysis of carbon emission restriction policies was conducted outside the NEMS using the full DRI models. While it is still true that a full DRI model simulation is the basis for the final macroeconomic analysis of policy, the kernel regression model provides an approximation of the macroeconomic response during an integrated NEMS run, leading to a more complete equilibrium solution for the energy markets.

In order to run a full DRI model simulation of a policy such as an auctioned carbon permit system 26 energy variables within the DRI model are altered to replicate the NEMS solution. These energy variables include energy prices, energy quantities produced, and energy quantities consumed. The total amount of money collected by the Federal Government through the permit auction is also input to the DRI model. This is the extent of the information provided by NEMS energy models. However, additional assumptions must be made in order to perform the DRI simulation. These assumptions include fiscal policy (what the Federal Government does with the permit revenue), and monetary policy (how the Federal Reserve is assumed to react). With this information, the simulation can be performed and the macroeconomic impact of the policy reported.

To construct the database for the policy kernel regression model, consistent sets of the energy inputs and revenue collections representative of the policy of interest must be provided to run the DRI model. The inputs must be internally consistent because they are jointly determined (price and quantity combinations are determined by the NEMS equilibrium solution). Populating the database with DRI simulation results based on actual NEMS energy inputs leads to more accurate approximations of the macroeconomic impacts during an integrated NEMS run. The input sets are created either by setting a target emission level as the objective in the NEMS solution, or by directly setting the permit price.

NEMS is a complex modeling system that requires several iterations of integrated runs before a converged solution is reached. Preliminary NEMS runs of the scenarios considered in Impacts of the Kyoto Protocol on U.S. Energy Markets and Economic Activity provided the inputs used to construct the carbon policy portion of the kernel regression database.

The solution algorithm of the kernel regression model can be described as a two-stage selection process. The first stage determines the portion of the database to consider when reaching a solution. The restriction is based on several parameters that are set at the time the NEMS run is submitted. The first decision point is whether macroeconomic feedback should be turned on for the integrated NEMS run. Feedback should only be turned on if the current policy under consideration is adequately represented within the database (the current database was constructed to analyze world oil price scenarios and carbon emission restriction policies involving permits or taxes). If feedback is deemed appropriate, the second parameter determines whether the NEMS run represents a change to the world oil price, or a carbon emission restriction policy. For a carbon emission restriction policy, two additional parameters must be provided: the first year of the emissions restriction (current options 2000, 2005, 2010, and 2012), and the assumed technology penetration (frozen technology menu, base technology menu, high technology menu - - only relevant for first emission restriction year of 2005). Each combination of parameters defines the segment of the database to be considered for the integrated NEMS run.

Once the relevant segment of the database has been determined, the values of the energy prices and quantities, and the carbon revenue are used to reach the kernel solution. Although 27 variables are input to the DRI model when analyzing a carbon emission reduction policy, the kernel regression model uses a subset of twelve of the variables when determining the solution. The variables included in the subset were chosen because they sufficiently identify the appropriate neighborhood for the solution. Table 4 contains a list and description of the twelve input variables.

Table 4: Input Variable Description.

1. &TXGF – Excise Tax Collections.
2. WPI051 – Industrial Steam Coal Price.
3. WPI053 – Industrial Natural Gas Price.
4. WPI054 – Industrial Electricity Price.
5. WPI055 – Utility Natural Gas Price.
6. WPI0561 – World Oil Price.
7. WPI057 – Industrial Average Refined Petroleum Product Price.
8. DTFUELSALLB – Total Demand for Fuels.
9. DENDUSE@COAL – End-Use Demand for Coal.
10. DENDUSE@ELC – End-Use Demand for Electricity.
11. DENDUSE@PET – End-Use Demand for Petroleum.
12. DENDUSE@NG – End-Use Demand for Natural Gas.

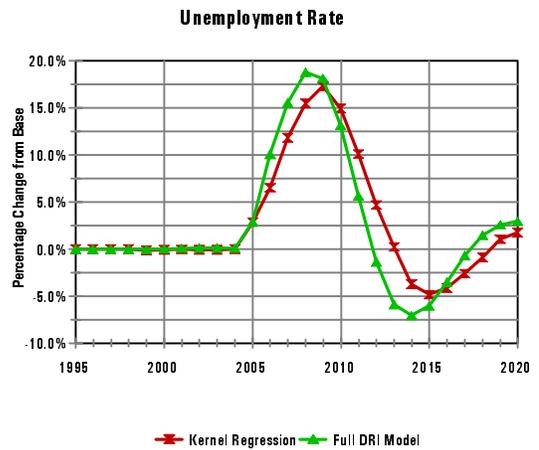
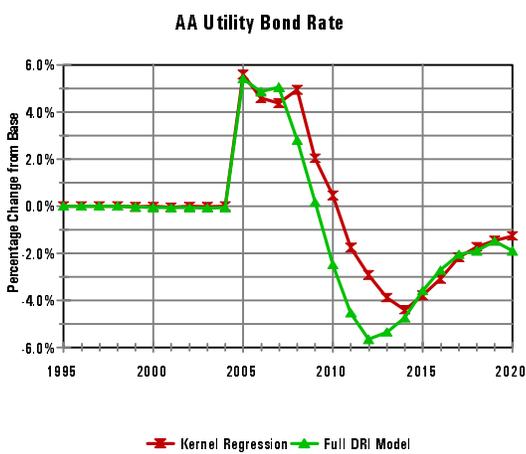
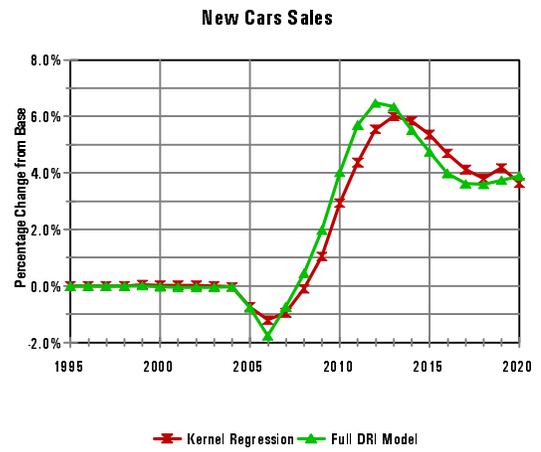
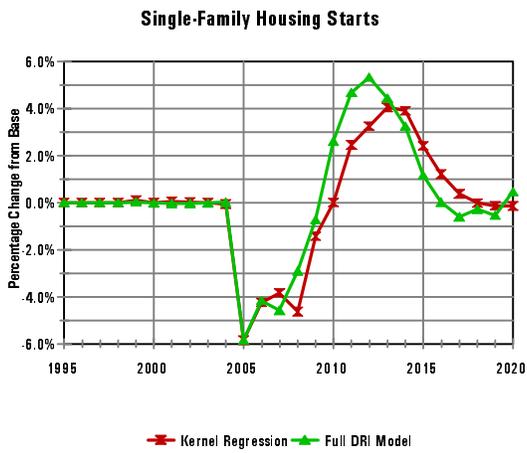
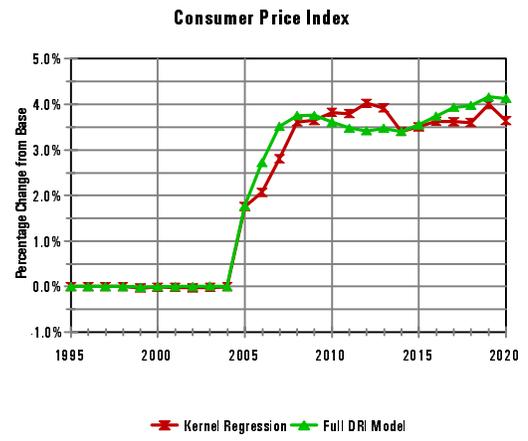
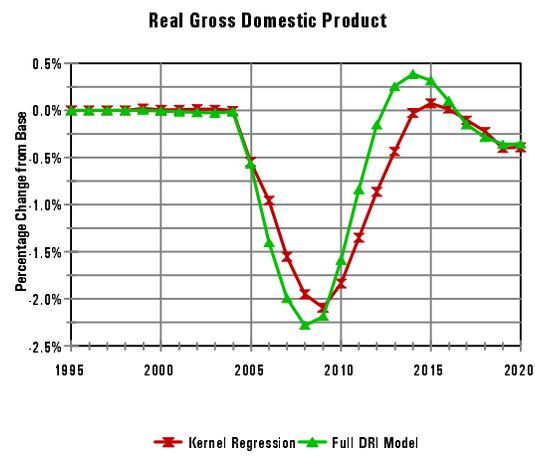
The selection process for a world oil price scenario is similar, but simpler. The only two parameter settings required before the kernel regression model can be solved are that the current NEMS run should include macro feedback, and that the run is a world oil price scenario. The same twelve variables are then used to determine the kernel regression solution.

Accuracy Tests

An out of sample test was conducted to demonstrate the accuracy with which the kernel regression model approximates a full DRI model simulation. The test looks at the kernel regression models accuracy for a carbon emission restriction policy (one of the central cases analyzed in the report [Impacts of the Kyoto Protocol on U.S. Energy Markets and Economic Activity](#)). After the integrated NEMS run was complete, the energy prices and quantities, (and permit revenues for the carbon policy) required to run the full DRI model were saved, along with the kernel regression solution. The NEMS inputs were then used to solve the full DRI model, and the DRI simulation solutions saved. The results for the two models were then compared in percentage change from base terms. Figure 3 compares the carbon policy results for six of the macroeconomic variables included in the MAM. The figures demonstrate that the kernel regression model does a good job of replicating the response of the full DRI model.

¹ Energy Information Administration, *Service Report: Analysis of Carbon Stabilization Cases*, SR-OIAF/97-01 (Washington, DC, October 1997.)

Figure 3. Comparison of Kernel Regression Results with Full DRI Simulation Results Carbon Emissions Restricted to 9% Above 1990 Level, Personal Income Tax Rebate



ⁱⁱ A complete list and description of the macroeconomic concepts are in Appendix D of the following publication: Energy Information Administration, Macroeconomic Activity Module (MAM): Kernel Regression Documentation, Contract No.: DE-AM01-98EI, Task: 90203 (Washington, DC, October 15, 1998), pg. 27.

ⁱⁱⁱ A complete list and description of the input variables are in Appendix C of the following publication: Energy Information Administration, Macroeconomic Activity Module (MAM): Kernel Regression Documentation, Contract No.: DE-AM01-98EI, Task: 90203 (Washington, DC, October 15, 1998), pg. 25.

^{iv} No attempt is made to refine this specification. The residuals are, in fact, serially correlated.

CONTENTS

	Page
1. INTRODUCTION.....	1
2. MCINTER2 INCLUDE FILE	3
3. INPUT FILES	7
MCPARMS	7
MCKBASE.....	7
MCDRII/MCDRIO	7
MCPCOFI/MCPCOFO	8
4. MAC.F SOURCE FILE.....	9
Subroutine MAC	9
Subroutine READMAC	9
Subroutine MCKCOMMSUB	10
Subroutine KERNELSUB	10
Subroutine KERDATA.....	15
Subroutine MACOUTPUT.....	15
APPENDICES.....	17
Appendix A. MCINTER2 Include File.....	19
Appendix B. MCPARMS Input File.....	23
Appendix C. Input Variable Description	25
Appendix D. Output Variable Description	27
Appendix E. Listing of Subroutine READMAC.....	31
Appendix F. Listing of Subroutine MCKCOMMSUB	41
Appendix G. Listing of Subroutine KERNELSUB	45
Appendix H. Listing of Subprogram KERDATA	53
Appendix I. Partial Listing of Subroutine MACOUTPUT	55

**This Page is
Intentionally
Left Blank**

1. INTRODUCTION

For the 1999 Annual Energy Outlook, the response surface model representation of Standard and Poor's DRI US model was replaced in the Macroeconomic Activity Module with a nonparametric estimation technique known as kernel regression. Kernel estimators are commonly used for nonparametric estimation of regression functions. Consider the following model

$$Y_i = m(x_i) + e_i$$

The Y_i 's are observed random variables, the x_i 's are known constants and the e_i 's are independent random variables with mean zero and constant variance. In the Macroeconomic Activity Module, the Y_i 's are the one hundred and nineteen national and regional macroeconomic concepts that the module forecasts. The x_i 's are twelve inputs from the National Energy Modeling System. These inputs are tax collections and energy prices and quantities.

Kernel regression estimates the unknown function m without assuming more about the function than certain smoothness conditions. Using a kernel estimator requires the selection of a kernel function and a bandwidth. The kernel function determines the shape of a sequence of weights applied to data within a neighborhood whose size is determined by the bandwidth. The Nadaraya-Watson estimator is used as the kernel estimator in the Macroeconomic Activity Module

$$m(x) = \frac{\sum K(u_i) Y_i}{\sum K(u_i)}$$

The kernel function is Epanechnikov

$$K(u) = (1 - u^2)$$

The bandwidth in the Macroeconomic Activity Module is not fixed but is a function of the number of observations and variance. If the resulting neighborhood contains no elements, a fixed factor is applied. This adjustment is repeated until at least one element appears in the neighborhood or until a predetermined threshold is exceeded. If the threshold is exceeded, the kernel regression writes an error and terminates.

To estimate the unknown function m , the kernel regression relies upon databases containing twenty-four simulations of Standard and Poor's DRI US model. The simulations represent different assumptions about the path of the real average refiners' acquisition price of crude oil. The databases are linked so that the input variables (i.e. tax collections and energy prices and quantities) from each simulation are associated with the respective output variables (i.e. macroeconomic concepts.)

The remainder of this deliverable documents changes to the Macroeconomic Activity Module for the 1999 Annual Energy Outlook required to implement the kernel regression model. The next section describes changes to the MCINTER2 include file. This is followed by changes and additions to the Macroeconomic Activity Module's input files. The fourth section describes changes to the source code. The deliverable concludes with appendices referenced in the text.

**This Page is
Intentionally
Left Blank**

2. MCINTER2 INCLUDE FILE

The MCINTER2 file declares constants and variables and dimensions arrays and matrices used in the Macroeconomic Activity Module. Appendix A lists the MCINTER2 include file used in the 1999 Annual Energy Outlook. Additions to the MCINTER2 include file needed for the kernel regression are highlighted in red and are described below.

ESTIMATR is a three dimensional matrix of real numbers containing the macroeconomic forecast. Its values are computed using data from the KERNEL, OUTPUTS and KCOUNT matrices. The first dimension is the number of macroeconomic or output variables, OVARS. The second dimension is the number of simulations per variable, NSIMS, plus two. The third dimension is the number of observations (years) per simulation, NOBS.

OUTPUTS is a three dimensional matrix of real numbers containing values of the macroeconomic or output variables. Its values are read from a human-readable input file MCPCFOFO. The first dimension is the number of macroeconomic or output variables, OVARS. The second dimension is the number of simulations per variable, NSIMS. The third dimension is the number of observations (years) per simulation, NOBS.

KERNEL is a three dimensional matrix of real numbers containing the computed kernels. Its values are computed using data from the SAMPLE, NORMAL, and BANDWTH matrices. The first dimension is the number of input variables, NVAR, plus one. The second dimension is the number of simulations per variable, NSIMS, plus one. The third dimension is the number of observations (years) per simulation, NOBS.

NORMAL is a three dimensional matrix of real numbers containing normalized values of the input variables. The inputs are tax collections and energy prices and quantities. Its values are computed using data from the INPUTS matrix. The first dimension is the number of input variables, NVAR. The second dimension is the number of simulations per variable, NSIMS, plus one. The third dimension is the number of observations (years) per simulation, NOBS.

INPUTS is a three dimensional matrix of real numbers containing values of the input variables. The inputs are tax collections and energy prices and quantities. Its values are read from a human-readable input file MCPCFOFI. The first dimension is the number of input variables, NVAR. The second dimension is the number of simulations per variable, NSIMS, plus one. The third dimension is the number of observations (years) per simulation, NOBS.

EPMAC is a two dimensional matrix of real numbers containing percent changes from the base used in DRI simulations. Its values are computed using data from the ESTIMATR and MCDRIO matrices. The first dimension is the sum of the number of non-regional, MACNMMAC, and regional, MCNMMACREG, macroeconomic variables. The second dimension is the number of observations (years) per simulation, NOBS.

MCDRIO is a two dimensional matrix of real numbers containing reference values of the macroeconomic or output variables used in DRI simulations. Its values are read from a human-readable input file with the same name. The first dimension is the number of output variables, OVARS. The second dimension is the number of observations (years) per simulation, NOBS.

MCDRII is a two dimensional matrix of real numbers containing reference values of the input variables used in DRI simulations. Inputs are tax collections and energy prices and quantities. Its values are read

from a human-readable input file with the same name. The first dimension is the number of input variables, NVARs. The second dimension is the number of observations (years) per simulation, NOBS.

WEIGHT is a two dimensional matrix of real numbers containing computed weights. Its values are computed using data from the KERNEL and KCOUNT matrices. The first dimension is the number of simulations per variable, NSIMS, plus one. The second dimension is the number of observations (years) per simulation, NOBS. The computed weights are not used directly in the kernel estimator but serve as a diagnostic.

SAMPLE is a two dimensional matrix of real numbers containing *adjusted* values of NEMS's input variables. Inputs are tax collections and energy prices and quantities. Its values are computed using data from MCKCOMM, MCKBASE and MCDRII matrices. The first dimension is the number of input variables, NVARs. The second dimension is the number of observations (years) per simulation, NOBS.

NSAMPLE is a two dimensional matrix of real numbers containing normalized *adjusted* values of the NEMS's input variables. Inputs are tax collections and energy prices and quantities. Its values are computed using data from the INPUTS matrix. The first dimension is the number of input variables NVARs. The second dimension is the number of observations (years) per simulation, NOBS.

STDDEV is a two dimensional matrix of real numbers containing the standard deviations of the normalized input variables. Inputs are tax collections and energy prices and quantities. Its values are computed using data from the NORMAL matrix. The first dimension is the number of input variables, NVARs. The second dimension is the number of observations (years) per simulation, NOBS.

BANDWTH is a two dimensional matrix of real numbers containing computed bandwidths. Its values are computed using data from the STDDEV matrix. The first dimension is the number of input variables, NVARs. The second dimension is the number of observations (years) per simulation, NOBS.

FACTOR is a two dimensional matrix of real numbers containing computed bandwidth adjustment factors. The values of FACTOR are one unless adjusted upward to increase neighborhood size. The first dimension is the number of input variables, NVARs. The second dimension is the number of observations (years) per simulation, NOBS.

MCKBASE is a two dimensional matrix of real numbers containing *raw* base values of the NEMS's input variables for the high, mid and low macroeconomic scenarios. Inputs are tax collections and energy prices and quantities. Its values are read from a human-readable input file with the same name. The first dimension is the number of input variables, NVARs. The second dimension is the number of observations (years) per simulation, NOBS, plus eight.

MCKCOMM is a two dimensional matrix of real numbers containing *raw* values of the NEMS's input variables. Inputs are tax collections and energy prices and quantities. The first dimension is the number of input variables, NVARs. The second dimension is the number of observations (years) per simulation, NOBS, plus eight.

KCOUNT is an array of integers containing counts of the number of computed kernels. It is dimensioned by the number of observations (years) per simulation, NOBS.

DATE_TIME is an array of integers containing a date stamp. BIG_BEN is a character array containing a time stamp.

INVARs and TAXVARs are identical character arrays containing input variable labels. At one point in the development of the kernel regression there were alternative sets of input variables whose use depended on the type of NEMS simulation. Inputs are tax collections and energy prices and quantities. Both arrays are dimensioned by NVARs, the number of input variables.

OUTVARS is a character array containing macroeconomic or output variable labels. It is dimensioned by OVARS, the number of output variables.

NVARS is an integer representing the number of input variables. Inputs are tax collections and energy prices and quantities. For the 1999 Annual Energy Outlook, its value is set to twelve.

OVARS is an integer representing the number of macroeconomic or output variables. For the 1999 Annual Energy Outlook, its value is set to one hundred and nineteen.

NSIMS is an integer representing the number of simulations per variable. For the 1999 Annual Energy Outlook, its value is set to twenty-four.

NOBS is an integer representing the number of observations (years) per simulation. For the 1999 Annual Energy Outlook, its value is set to twenty-three.

KSTYR is an integer representing the start year of simulations in the kernel regression databases. For the 1999 Annual Energy Outlook, its value is set to 1998

FTAXYR is an integer representing the start year of tax collections. The value of FTAXYR is set in the scenario descriptor file. Its scenario descriptor name is MACTAXYR. Legal arguments for FTAXYR and MACTAXYR are 0 and the years 2000, 2005, 2010 and 2012. The default value is 2005. For the 1999 Annual Energy Outlook, FTAXYR and MACTAXYR are set to zero.

TAXMODE is an integer representing the presence and type of tax policy. The value of TAXMODE is set in the scenario descriptor file. Its scenario descriptor name is MACTAX. Legal arguments for TAXMODE and MACTAX are 0 (no assumed tax policy,) 1 (deficit reduction,) 2 (personal tax,) 3 (corporate tax) and 4 (social security.) The default value is zero. For the 1999 Annual Energy Outlook, TAXMODE and MACTAX are set to zero.

MCTECH is an integer representing technology assumptions. The value of MCTECH is set in the scenario descriptor file. Its scenario descriptor name is MACTECH. Legal arguments for MCTECH and MACTECH are 1 (frozen,) 2 (base,) 3 (high) and 4 (high coal.) The default value is two. For the 1999 Annual Energy Outlook, MCTECH and MACTECH are set to two.

DIAGS is a logical switch controlling diagnostic output written to the human-readable file MACOUT5. The value of DIAGS is set in the Macroeconomic Analysis Module's parameter file, MCPARMS. Its legal arguments are ".TRUE." and ".FALSE.." For the 1999 Annual Energy Outlook, DIAGS is set to ".TRUE.."

BWRELAX is a logical switch controlled by the Macroeconomic Analysis Module and used to check if the bandwidth needs upward adjustment.

OVERRIDE is a logical switch controlling computation of the bandwidth. The value of OVERRIDE is set in the Macroeconomic Analysis Module's parameter file, MCPARMS. Its legal arguments are ".TRUE." and ".FALSE.." If its value is ".TRUE.," then the user must supply a value for the bandwidth on the line following OVERRIDE in the MCPARMS file. The default value in the MCPARMS file is 0.001. If ".FALSE.," then the Macroeconomic Analysis Module computes the bandwidth. For the 1999 Annual Energy, OVERRIDE is set to ".FALSE.."

SNSIMS is an integer representing the number of the starting simulation drawn from the kernel databases for the current NEMS run. The Macroeconomic Activity Module given the particulars of the NEMS run determines the value of SNSIMS.

ENSIMS is an integer representing the number of the ending simulation drawn from the kernel databases for the current NEMS run. The Macroeconomic Activity Module given the particulars of the NEMS run determines the value of ENSIMS.

**This Page is
Intentionally
Left Blank**

3. INPUT FILES

Implementing the kernel regression for the 1999 Annual Energy Outlook required changes to one existing Macroeconomic Activity Module input file and the addition of five others. The macroeconomic parameter file, MCPARMS, has a half dozen new parameters that usually remain unchanged from simulation to simulation. The five new input files required for the kernel regression are all databases. Four of the databases are paired, i.e. MCDRII to MCDRIO and MCPCOFI to MCPCOFO. In addition, the twenty-four simulations contained in each of the MCPCOFI and MCPCOFO files are linked and segmented.

MCPARMS

A complete listing of the Macroeconomic Activity Module's parameter file, MCPARMS, is contained in Appendix B. This is the version used in the 1999 Annual Energy Outlook. The additional parameters for the kernel regression are highlighted in red. The first parameter is the number of macroeconomic or output variables, OVARS. The value of OVARS is set to one hundred and nineteen. Appendix D contains a complete list and description of these variables. The second parameter, NOBS, is the number of observations (years) per simulation. It is set to twenty-three. This is the difference between the years 1998 and 2020, inclusive. KSTYR is the third parameter and is the start year of the simulations. The value of KSTYR is 1998. The next two parameters are logical switches. The DIAGS parameter controls the writing of diagnostics to the human-readable MACOUT5 file. It is set to '.TRUE.' The OVERRIDE parameter controls the global bandwidth. It is set to '.FALSE.:' therefore, the kernel regression computes local bandwidths. If OVERRIDE is '.TRUE.:', the value of the final parameter, BANDWTH, is used as the global bandwidth. These parameters are read by code contained in the READMAC subroutine described below.

MCKBASE

The MCKBASE input file is a database containing, for each of three macroeconomic scenarios, the *raw* base values of the input variables. The input variables are tax collections and energy prices and quantities. Appendix C contains a complete list and description of the input variables. The three scenarios are high, mid and low macroeconomic activity. The input data is considered *raw* because it has not been adjusted to the base used to generate the kernel regression simulations. The DRI reference values of the input variables used to generate the kernel regression simulations are contained in the MCDRII input file. Computed inputs from a current NEMS simulation are compared to the base values in the MCKBASE input file to determine the extent to which tax collections and energy prices and quantities have changed. When creating the NEMS reference cases, it frequently becomes necessary to revise the data in MCKBASE. At the bottom of the human-readable MACOUT4 output file is the computed *raw* base values of the input variables for that particular run. A copy, paste and a little formatting are all that is required to "refresh" the base values in MCKBASE. The data in MCKBASE starts in 1990 and ends in 2020.

MCDRII and MCDRIO

The MCDRII and MCDRIO input files are databases containing the reference values of the input and output variables used in the simulations of Standard and Poor's DRI US model. Variable descriptions for the tax collections and energy prices and quantities are contained in Appendix C. Output variable descriptions are contained in Appendix D. *Raw* inputs from a NEMS run are adjusted to the reference inputs in MCDRII prior to solving the kernel regression model. Once the kernel regression model solves,

its forecast is adjusted to the Macroeconomic Activity Module's EBMAC using the reference outputs in MCDRIO. The data in MCDRII and MCDRIO start in 1998 and end in 2020.

MCPCOFI and MCPCOFO

The MCPCOFI and MCPCOFO input files are linked and segmented databases each containing data from twenty-four simulations. The data in the MCPCOFI file are values of the input variables in each of the twenty-four simulations. The data in the MCPCOFO file are values of the macroeconomic or output variables for each of the twenty-four simulations. Each of the simulations in the two input files is linked so that the inputs for any one simulation are bound to its outputs. Within the twenty-four simulations are three equally sized subsets. The first eight simulations assume low world oil prices. The second set of eight simulations assumes mid world oil prices. The final eight simulations assume high world oil prices. Selecting one of these subsets is controlled by the scenario descriptor parameter WWOP. Data in MCPCOFI and in MCPCOFO starts in 1998 and ends in 2020.

4. MAC.F SOURCE FILE

Incorporating the kernel regression model into the MAC.F source code required moderate changes to three pre-existing subroutines, MAC, READMAC and MACOUTPUT. No change was made to the primary function of each of these subroutines. Two new subroutines and a subprogram were added to the source file. The MCKCOMMSUB subroutine computes inputs to the kernel regression model. The model is contained in the KERNELSUB subroutine. The new subprogram is KERDATA. It contains two character arrays of labels for the input and output variables.

Subroutine MAC

The MAC subroutine directs execution of the Macroeconomic Activity Module. Few changes to the MAC subroutine were required to incorporate the kernel regression. Two new calls were added for subroutines MCKCOMMSUB and KERNELSUB. Both of these subroutines are new to the Macroeconomic Activity Module and are described below. A third call to the MACROSUB subroutine was commented out. The MACROSUB subroutine contains the response surface model. The following is a partial listing of the MAC subroutine showing the code changes.

```
C Kernel regression - Call Kernel Submodule
  CALL MCKCOMMSUB
  CALL KERNELSUB

C Call the Macroeconomic Submodule.
C  CALL MACROSUB
```

Subroutine READMAC

The READMAC subroutine is called by the MAC subroutine. It reads many separate files containing parameters, coefficients and data used by the Macroeconomic Activity Module. A complete listing of the READMAC subroutine is contained in Appendix E. Highlighted in red are lines added for the kernel regression. The first file read by the READMAC subroutine is MCPARMS. A complete listing of MCPARMS is included in Appendix B. Some of the parameters required by the kernel regression are at the bottom of the MCPARMS file.

In addition to the read statements are two statements reading parameters set in the scenario descriptor file. The first of these is a tax policy parameter. In the Macroeconomic Activity Module, it is referred to as TAXMODE while in the scenario descriptor file it is known as MACTAX. Valid arguments for TAXMODE and MACTAX are 0 (no assumed tax policy,) 1 (deficit reduction,) 2 (personal tax,) 3 (corporate tax) and 4 (social security.) The second is a parameter indicating the start year of the particular tax. This is referred to as FTAXYR in the Macroeconomic Activity Module and as MACTXYR in the scenario descriptor file. Valid arguments for FTAXYR and MACTAXYR are 0 and the years 2000, 2005, 2010 and 2012. Together these parameters help determine which portions of the kernel regression's selected databases to use.

On every simulation, the READMAC subroutine reads databases containing reference values of the input and output variables used in DRI simulations. These databases are contained in the MCDRII and MCDRIO files, respectively. The subroutine also reads the MCKBASE file containing *raw* base values of the NEMS's input variables for the high, mid and low macroeconomic scenarios.

For the 1999 Annual Energy Outlook, the READMAC subroutine reads the MCPCOFI and MCPCOFO files for values of the input and output variables given assumptions about the world oil price. These last two files are selected by setting MACTXYR to '0' in the scenario descriptor file. Which of the twenty-four simulations contained in each of these files is selected depends upon the assumed path of the world oil price. Controlling this is the scenario descriptor parameter WWOP. WWOP is set to '1' for a low world oil price path. The first eight simulations in both the MCPCOFI and MCPCOFO files are generated in the DRI US model by assuming low world oil prices. Base simulations are the next eight and are selected by setting WWOP to '2.' The final eight simulations were generated assuming high world oil price paths. They are selected by setting WWOP to '3.'

Subroutine MCKCOMMSUB

The MCKCOMMSUB subroutine is called by the MAC subroutine. This is an entirely new subroutine to the Macroeconomic Activity Module. During a simulation, it computes and applies the change in input variables from the National Energy Modeling System's base to the DRI base. This occurs on every iteration of the model. Inputs are tax collections and energy prices and quantities. There is a complete list of the inputs in Appendix C. A complete listing of the MCKCOMMSUB subroutine is contained in Appendix F. The first part of the subroutine reads tax collections and energy prices and quantities from the common block and computes aggregates used in the kernel regression. The second part of the subroutine computes the change in these aggregates from base values and applies the change to the base used in simulations of the DRI model used to create the kernel regression databases. The final portion of the MCKCOMMSUB subroutine writes the current iteration's results to the MACOUT5 human-readable file.

Subroutine KERNELSUB

The KERNELSUB subroutine is called by the MAC subroutine. This is an entirely new subroutine to the Macroeconomic Activity Module, and it contains the kernel regression model. During a simulation, the KERNELSUB subroutine uses inputs from MCKCOMMSUB subroutine to compute a forecast of the 119 macroeconomic variables listed in Appendix D. The kernel regression model solves on every iteration of the National Energy Modeling System. A complete listing of the KERNELSUB subroutine is contained in Appendix G. A substantial part of the code writes diagnostics to the MACOUT5 human-readable file.

Files containing databases used by the kernel regression are selected prior to execution of the KERNELSUB subroutine. Not selected are the relevant simulations within the chosen databases. In the following excerpt from the subroutine, the scenario descriptor parameter WWOP is used to select the appropriate eight simulations used in the 1999 Annual Energy Outlook.

```
IF (FTAXYR .EQ. 0 .AND. WWOP .EQ. 1) THEN ! LOW WOP
  SNSIMS = 1
  ENSIMS = 8
ELSEIF (FTAXYR .EQ. 0 .AND. WWOP .EQ. 2) THEN ! BASE
  SNSIMS = 9
  ENSIMS = 16
ELSEIF (FTAXYR .EQ. 0 .AND. WWOP .EQ. 3) THEN ! HIGH WOP
  SNSIMS = 17
  ENSIMS = 24
```

In the next excerpt from the KERNELSUB subroutine, the values of the input variables from the selected simulations are normalized using their means.

```
C Compute a mean for every input variable at each period t. Computed means
C are written to the DIAGS.PRN file.
```

```

DO 5050 I=1,NVARS
DO 5060 K=1,NOBS
  INPUTS(I,NSIMS+1,K) = 0.0 ! Initialize
DO 5070 J=SNSIMS,ENSIMS
  INPUTS(I,NSIMS+1,K) = INPUTS(I,NSIMS+1,K)+INPUTS(I,J,K) ! Aggregate
5070   CONTINUE
  INPUTS(I,NSIMS+1,K) = INPUTS(I,NSIMS+1,K)
  *   /(ENSIMS-SNSIMS+1) ! Compute mean
5060   CONTINUE
5050   CONTINUE

```

C Normalize inputs using the computed means. Normalized input variable
C values are written to the DIAGS.PRN file.

```

DO 5080 I=1,NVARS
DO 5090 K=1,NOBS
DO 50100 J=SNSIMS,ENSIMS
  IF (INPUTS(I,NSIMS+1,K) .NE. 0.0) THEN
    NORMAL(I,J,K) = INPUTS(I,J,K)/INPUTS(I,NSIMS+1,K)
  ELSE
    NORMAL(I,J,K) = 0.0
  ENDIF
50100   CONTINUE
5090   CONTINUE
5080   CONTINUE

```

In the next excerpt from the KERNELSUB subroutine, the normalized values of the input variables in the database are used to compute local bandwidths.

C Compute an initial value of the local bandwidths. This block of code
C is executed if not overridden with a global bandwidth. It begins by
C computing the standard deviation of the normalize input variables at
C each period t. The computed standard deviations are written to the
C DIAGS.PRN file. The computed bandwidth is an increasing function of the
C standard deviation and a decreasing function of the number of simulation,
C NSIMS. The initial values of the local bandwidth are not written to an
C output file.

```

DO 50179 I=1,NVARS
  STDDEV(I,MCIYR-8) = 0.0 ! Initialize
50179   CONTINUE

  IF (.NOT. OVERRIDE) THEN
DO 50160 I = 1,NVARS
DO 50180 J = SNSIMS,ENSIMS
  STDDEV(I,MCIYR-8) = STDDEV(I,MCIYR-8)          ! Aggregate the squared deviations
  *   +(NORMAL(I,J,MCIYR-8)-NORMAL(I,NSIMS+1,MCIYR-8))**2 ! from mean
50180   CONTINUE
  STDDEV(I,MCIYR-8) = ((1.0/(DBLE(ENSIMS-SNSIMS+1))-1.0))
  *   *STDDEV(I,MCIYR-8)**0.5 ! Compute standard deviation
50160   CONTINUE
DO 50190 I = 1,NVARS
  BANDWTH(I,MCIYR-8) = 1.06*STDDEV(I,MCIYR-8)
  *   *(DBLE(ENSIMS-SNSIMS+1)**(-0.2)) ! Compute local bandwidth
50190   CONTINUE

```

```

WRITE(14,*)
WRITE(14,*) 'SNSIMS ',SNSIMS,'ENSIM ',ENSIMS,
* 'NSIMS ',(ENSIMS-SNSIMS+1)
DO 9002 I=1,12
WRITE(14,*) 'NVAR ',TAXVARS(I)
WRITE(14,*) 'STDDEV ',STDDEV(I,MCIYR-8),
* 'BANDWTH ',BANDWTH(I,MCIYR-8)
9002 CONTINUE
ENDIF

```

Prior to computing the kernels, the values of inputs coming from the National Energy Modeling System are normalized using the means computed earlier.

```

C Normalize sample inputs using the computed means. Normalized sample
C input variable values are written to the DIAGS.PRN file.
DO 50110 I=1,NVARS
IF (INPUTS(I,NSIMS+1,MCIYR-8) .NE. 0.0) THEN
NSAMPLE(I,MCIYR-8) = SAMPLE(I,MCIYR-8)
* /INPUTS(I,NSIMS+1,MCIYR-8)
ELSE
NSAMPLE(I,MCIYR-8) = 0.0
ENDIF
50110 CONTINUE

```

Once initial values of the bandwidths have been computed and all the inputs normalized, the KERNELSUB subroutine computes the kernels. In the following excerpt, the subroutine tests for the existence of a single computed kernel for each input variable across the relevant simulations from the database. If none exists, then FACTOR is used to expand the bandwidth. The subroutine again moves across the simulations and attempts to compute kernels for the particular input variable. This process is repeated until either a single simulation is found in the neighborhood or a threshold is exceeded. If the subroutine cannot compute a kernel for a particular input variable at a particular time, the subroutine writes an error and the program terminates.

```

C Compute kernels. This block computes a kernel for each simulation at
C period t for every input variable given the bandwidth. If in period
C t no kernels result, FACTOR is used to increase upward the bandwidth.
C This continues until the bandwidth exceeds 1.0. At that point, the
C program writes an error to screen and terminates. The values of KERNEL
C and the final values of FACTOR and BANDWTH are written to the DIAGS.PRN
C file.
DO 50207 I = 1,NVARS+1
DO 50208 J = 1,NSIMS+1
KERNEL(I,J,MCIYR-8) = -1.0 ! Initialize for sims outside a neighborhood
50208 CONTINUE
50207 CONTINUE

DO 50209 I = 1,NVARS
FACTOR(I,MCIYR-8) = 1.0 ! Initialize bandwidth adjustment
50209 CONTINUE

DO 50210 I=1,NVARS
BWRELAX = .TRUE. ! Initialize logical test for bandwidth adjustment

```

```

DO 50230 WHILE (BWRELAX)
DO 50240 J=SNSIMS,ENSIMS
  IF (BANDWTH(I,MCIYR-8) .EQ. 0.0) THEN
    BANDWTH(I,MCIYR-8) = 0.001
  ENDIF
  IF (ABS((NSAMPLE(I,MCIYR-8)-NORMAL(I,J,MCIYR-8))
*   /BANDWTH(I,MCIYR-8)) .LE. 1.0) THEN ! Test if sim is inside the neighborhood
  IF (KERNEL(NVARS+1,J,MCIYR-8) .EQ. -1.0) THEN
    KERNEL(NVARS+1,J,MCIYR-8) = 1.0 ! Initialize product of bandwidths
  ENDIF
  KERNEL(I,J,MCIYR-8) =
*   (.75/BANDWTH(I,MCIYR-8))
*   *(1.0-(((NSAMPLE(I,MCIYR-8)-NORMAL(I,J,MCIYR-8))
*   /BANDWTH(I,MCIYR-8))**2)) ! Compute kernel
  KERNEL(NVARS+1,J,MCIYR-8) = KERNEL(NVARS+1,J,MCIYR-8)
*   *KERNEL(I,J,MCIYR-8) ! Compute kernel product
  BWRELAX = .FALSE. ! Bandwidth adjustment not necessary
ENDIF

  WRITE(14,*) 'NVAR ',TAXVARS(I),'SIM ',J
  WRITE(14,*) 'KERNEL ',KERNEL(I,J,MCIYR-8)
  WRITE(14,*) 'NSAMPLE ',NSAMPLE(I,MCIYR-8),
*   'NORMAL ',NORMAL(I,J,MCIYR-8),
*   'BANDWTH ',BANDWTH(I,MCIYR-8)

50240  CONTINUE
  IF (BWRELAX) THEN
    FACTOR(I,MCIYR-8) = FACTOR(I,MCIYR-8)+0.10 ! Increase bandwidth factor
    BANDWTH(I,MCIYR-8) = FACTOR(I,MCIYR-8)
*   *BANDWTH(I,MCIYR-8) ! Adjust bandwidth
  ENDIF
  IF (BANDWTH(I,MCIYR-8) .GT. 5.0) THEN
    WRITE(6,*) 'ERROR! Bandwidth adjustment has failed ',
*   'for variable ',INVAR(I),' in year ', MCIYR
    CLOSE (14)
    STOP ''
  ENDIF

  WRITE(14,*) 'FACTOR INCREASE FOR NVAR ',TAXVARS(I)
  WRITE(14,*) FACTOR(I,MCIYR-8),'BANDWTH ',BANDWTH(I,MCIYR-8)
  WRITE(14,*)

50230  CONTINUE
50210  CONTINUE

```

Once the kernels are computed, the KERNELSUB subroutine begins computing a solution to the kernel regression model. The Nadaraya-Watson estimator is used as the kernel estimator in the Macroeconomic Activity Module.

```

C Compute Rosenblatt-Parzen kernel densities using product kernels. The
C marginal densities are written to the DIAGS.PRN file.
  KCOUNT(MCIYR-8) = 0 ! Initialize
  KERNEL(NVARS+1,NSIMS+1,MCIYR-8) = 0.0 ! Initialize

```

```

DO 50260 J=SNSIMS,ENSIMS
  IF (KERNEL(NVARS+1,J,MCIYR-8) .GE. 0.0) THEN
    KERNEL(NVARS+1,NSIMS+1,MCIYR-8)
  *   = KERNEL(NVARS+1,NSIMS+1,MCIYR-8)
  *   +KERNEL(NVARS+1,J,MCIYR-8) ! Aggregate product kernels
    KCOUNT(MCIYR-8) = KCOUNT(MCIYR-8)+1
  ENDIF
50260  CONTINUE
  IF (KCOUNT(MCIYR-8) .GT. 0) THEN
    KERNEL(NVARS+1,NSIMS+1,MCIYR-8) =
  *   KERNEL(NVARS+1,NSIMS+1,MCIYR-8)
  *   /DBLE(KCOUNT(MCIYR-8)) ! Compute density
  ENDIF
...
C Compute Nadaraya-Watson Estimate. Estimates are written
C to the FORECAST.PRN file.
  DO 50298 I = 1,OVARS
    DO 50299 J = 1,NSIMS+2
      ESTIMATR(I,J,MCIYR-8) = 0.0 ! Initialize
50299  CONTINUE
50298  CONTINUE

  DO 50300 J=SNSIMS,ENSIMS
    IF (KERNEL(NVARS+1,J,MCIYR-8) .GE. 0.0) THEN
      DO 50310 I=1,OVARS
        ESTIMATR(I,J,MCIYR-8) = KERNEL(NVARS+1,J,MCIYR-8)
      *   *OUTPUTS(I,J,MCIYR-8)
        ESTIMATR(I,NSIMS+1,MCIYR-8)
      *   = ESTIMATR(I,NSIMS+1,MCIYR-8)
      *   +ESTIMATR(I,J,MCIYR-8)
50310  CONTINUE
      ENDIF
50300  CONTINUE
      DO 50320 I=1,OVARS
        ESTIMATR(I,NSIMS+2,MCIYR-8) = (ESTIMATR(I,NSIMS+1,MCIYR-8)
      *   /KERNEL(NVARS+1,NSIMS+1,MCIYR-8))/DBLE(KCOUNT(MCIYR-8))
50320  CONTINUE

```

The resulting kernel regression solution is based upon simulations generated by the DRI US model. In the following excerpt, the subroutine computes the change in the kernel regression solution from the DRI base, MCDRIO, and applies that to the Macroeconomic Activity Module's base, EBMAC. The rebased model solution contains the values of the output variables written to the common block.

```

DO 50330 I=1,MCNMMAC+MCNMMACREG
  IF (MACFDBK .NE. 0) THEN
    IF (I .LE. OVARS) THEN
      IF (MCDRIO(I,MCIYR-8) .NE. 0.0) THEN
        EPMAC(I,MCIYR) = (ESTIMATR(I,NSIMS+2,MCIYR-8)
      *   /MCDRIO(I,MCIYR-8))-1.0
      ELSE
        EPMAC(I,MCIYR) = 0.0

```

```

        ENDIF
        ENDIF
        ELSE
            EPMAC(I,MCIYR) = 0.0
        ENDIF
50330 CONTINUE
        ENDIF

        DO 50331 I=1,MCNMMAC+MCNMMACREG
            IF (I .LE. MCNMMAC) THEN
                ESMAC(I,MCIYR) = (1.0+EPMAC(I,MCIYR))*EBMAC(I,MCIYR)
            ELSE
                ESMACREG(11,I-MCNMMAC,MCIYR) = (1.0+EPMAC(I,MCIYR))
                * *EBMAC(I,MCIYR)
            ENDIF
60331 CONTINUE

```

Subprogram KERDATA

The KERDATA subprogram is an entirely new subprogram to the Macroeconomic Activity Module. A complete listing of the MCKCOMMSUB subprogram is contained in Appendix H. The KERDATA subprogram contains two character arrays, TAXVARS and OUTVARS, of input and output variable labels, respectively.

Subroutine MACOUTPUT

The MACOUTPUT subroutine is called by the MAC subroutine. All of the code for the kernel regression is added to the bottom of the subroutine. A partial listing of the MACOUTPUT subroutine is contained in Appendix I. The kernel regression writes out to two human-readable files, MACOUT4 and MACOUT5. MACOUT5 is a file of diagnostics and is written to while the Macroeconomic Activity Module is executing up to the point when MACOUTPUT is called. At that point, MACOUT5 is closed and MACOUT4 is opened for writes. All of the code in the MACOUTPUT subroutine is for writes to the MACOUT4 file. In addition to the kernel regression model forecast, MACOUT4 contains the values of inputs from the National Energy Modeling System. At the very bottom of the MACOUT4 file is a block of input variable data used to refresh the high, mid and low macroeconomic data in the MCKBASE file.

**This Page is
Intentionally
Left Blank**

APPENDICES

**This Page is
Intentionally
Left Blank**

Appendix A

MCINTER2 Include File

Appendix A

MCINTER2 Include File

```
C $Header: /mac-int/rt2/mam_aeo99/RCS/mcinter2,v 1.20 1998/09/01 17:53:29 rt2 Exp $
C
C          COMMON MCINTER2
C          Internal Variables for the Macroeconomic Submodel
C
C INTERNAL CONSTANT DEFINITION:
C
REAL BPRICE(6)      ! 1 BASE YEAR FUEL PRICES AND GNP DEFLATOR
REAL WPIBY(5)       ! 2 BASE YEAR FUEL PRICE COMPONENT WPI VALUES
REAL WPIWT(5)       ! 3 WEIGHTING FACTORS FOR FUEL PRICE COMPONENTS
REAL PCBPRICE(4)    ! 4 BASE YEAR CONSUMER ENERGY PRICES
REAL PABE(MNUMYR)   ! POLLUTION ABATEMENT EQ. (TEMPORARY)
REAL BPABE(MNUMYR)  ! POLLUTION ABATEMENT EQ. BASELINE
REAL TXGF(MNUMYR)   ! EXCISE TAXES (TEMPORARY)
REAL BTXGF(MNUMYR)  ! EXCISE TAXES BASELINE
REAL BMIPET(MNUMYR) ! MINING & PETROL INVEST BASELINE
REAL BPU(MNUMYR)    ! PUBLIC UTILITIES INVEST BASELINE
REAL MCCAPEEL(MNUMYR) ! INVESTMENT IN NONRESIDENTIAL STRUCTURE -
                    ! PUBLIC UTILITIES, ORIG. SOURCE: EMM/EFP/UEFP.F
INTEGER MCNMSERV    ! NO. OF SERVICE INDUSTRIES
INTEGER MCNMREGDRS  ! NO. OF REGIONAL SUBMODULE DRIVER VARS.
C
C READ INPUT CONSTANT DEFINITION:
C
REAL REGSHRS(9,92,MNUMYR) ! 5 REGIONAL SHARES
REAL BDRIVER(20,MNUMYR)   ! 6 BASELINE DRIVER VARIABLES
REAL EBMAC(149,MNUMYR)    ! 7 FINAL BASE MACRO FORECAST FRM DRI MODEL
REAL ECMACW(149,5)        ! 5 MACRO VARIABLE PREDICTOR COEFFICIENTS
REAL EBIND(53,MNUMYR)     ! 10 FINAL BASE INDUSTRIAL OUTPUT FORECAST
REAL EBINDOLD(45,MNUMYR)  ! 11 INITIAL BASE INDUST. OUTPUT FORECAST
REAL ECIND(45,59)         ! 14 INDUSTRIAL OUTPUT BASELINE SLOPE
REAL ECNDCH(45,59)        ! 15 INDUSTRIAL OUTPUT CHANGE IN SLOPE
REAL GRACOFFS(149,11)     ! 16 GROWTH RATE ADJ. COEFFICIENTS
REAL BDRIVREG(6,9,MNUMYR) ! 17 REGIONAL DRIVER VARIABLES
REAL ECREG(9,828,7)       ! 18 REGIONAL SUBMOD COEFFS
REAL EBREG(92,9,MNUMYR)   ! 19 BASE REGIONAL FORECAST FRM DRI MODEL
C
C INTERNAL VARIABLE DEFINITION:
C
REAL ESMACREG(11,92,MNUMYR) ! 17 SOLUTION MACRO VARS, REGIONALIZED
REAL EPMACW(149,MNUMYR+2)  ! 18 MACRO VARIABLE PCNT CHANGE FROM BASE
REAL EPREG(92,9,MNUMYR)    ! 18 REG. VARIABLE PCNT CHNG FROM BASE
REAL ESEMP(11,50,MNUMYR)   ! EMPLOYMENT SOLUTION
REAL EDEMP(48,MNUMYR)      ! EMPLOYMENT CHANGE
REAL ECEMP(48,3)           ! EMPLOYMENT COEFFICIENTS
```

REAL EBEMP(50,MNUMYR) ! EMPLOYMENT BASELINE

REAL ESMAC(92,MNUMYR) ! 19 SOLUTION MACRO VARIABLES, NOT REGIONALIZED
 REAL EBREGSUM(92,MNUMYR) ! 19 SUM OF REG VALUES FOR EACH VAR
 REAL EDIND(45,MNUMYR) ! 20 CHANGE IN INDUSTRIAL OUTPUT FORECAST
 REAL ESIND(11,42,MNUMYR) ! 21 INDUSTRIAL OUTPUT FORECAST
 REAL ESSERV(11,MNUMYR) ! 21 SERVICE OUTPUT FORECAST
 REAL GRADELTA(3,MNUMYR) ! 22 PCT. CHANGE FROM BASE, SUPPLY VARS
 REAL GRAPCTCH(149,MNUMYR) ! 23 PCT CHG FROM BSE, NEW MACRO BASELINE
 REAL DDRIVER(11,MNUMYR+5) ! 24 PCT CHG FROM BASE, DRIVER VARIABLES
 REAL DDRIVREG(6,9,MNUMYR) ! 24 PCT CHG FROM BASE, REG DRVR VARS
 REAL PCWPI05(5,MNUMYR) ! 24 PCT CHANGE IN WPI05 COMPONENTS
 REAL MCINDGROW(5,MNUMYR) ! 24 GROWTH RATES APPLIED TO 5 IND OUTS
 REAL RGADJ(6,MNUMYR) ! 25 REGIONAL DRIVER ADJUSTMENT VARIABLES
 REAL MCPCWGDP(MNUMYR) ! 26 BASELINE (ORIGINAL) GDP87 DEFLATOR
 INTEGER GRAFLAG ! 27 GROWTH RATE ADJUSTMENT FLAG
 INTEGER GRSWITCH ! 27 GROWTH RATE MODEL CHOICE
 INTEGER MCIYR ! 28 GROWTH RATE ADJUSTMENT FLAG
 REAL PCPASS(7) ! PASS THROUGH COEFFS FILTERING WPI EFFECT
 INTEGER INFDSW ! INDIVIDUAL FEEDBACK SWITCHES
 INTEGER RGFDSW(6) ! INDIVIDUAL REGIONAL FEEDBACK SWITHCES
 REAL NEWMAC(149,MNUMYR) ! MACRO BASELINE FOR GRIFFIN PROCEDURE
 REAL GRIFPCNT ! PCNT CHANGE CALCULATED IN GRIFFIN PROCEDURE
 REAL USERRATE ! DESIRED GROWTH RATE INPUT BY THE USER
 INTEGER NUMPCS ! NUMBER OF PC VARIABLES
 INTEGER NUMINVS ! NUMBER OF INVESTMENT VARIABLES
 INTEGER MCNXLAW ! NUMBER OF EXOG LAGS -- WPI05 MODEL
 INTEGER MCNELAGW ! NUMBER OF ENDO LAGS -- WPI05 MODEL
 INTEGER NUMEMPL ! NUMBER OF INDUSTRIAL EMPLOYMENT CATEGORIES
 REAL ALTCMACW(149,6) ! ALTERNATIVE WPI05 COEFFICIENTS
 REAL ALTPMACW(149,MNUMYR) ! ALTERNATIVE WPI05 PCNT CHANGE FROM BASE
 INTEGER SWITCHEC(149) ! SWITCHES FOR WPI05 COEFFICIENTS
 INTEGER NYRINDX ! NUMBER OF YEARS, 1990 to 2020, MNUMYR
 REAL INVAR(MNUMYR) ! INVESTMENT FOR NON-ELECTRIC UTILITIES
 REAL EBINV(216,MNUMYR) ! INVESTMENT IN CAPITAL EQUIPMENT
 INTEGER MCNMINV ! NUMBER OF CAPITAL EQUIPMENT CONCEPTS

C

C Kernel regression

C

DOUBLE PRECISION ESTIMATR(119,101,23) !ovars,nsims+2,nobs
 DOUBLE PRECISION OUTPUTS(119,99,23) !ovars,nsims,nobs
 DOUBLE PRECISION KERNEL(13,100,23) !nvars+1,nsims+1,nobs
 DOUBLE PRECISION NORMAL(12,100,23) !nvars,nsims+1,nobs
 DOUBLE PRECISION INPUTS(12,100,23) !nvars,nsims+1,nobs
 DOUBLE PRECISION EPMAC(149,MNUMYR) !mcnmmac+mcnmmacreg,mnumyr
 DOUBLE PRECISION MCDRIO(119,23) !ovars,nobs
 DOUBLE PRECISION MCDRII(12,23) !nvars,nobs
 DOUBLE PRECISION WEIGHT(100,23) !nsims+1,nobs
 DOUBLE PRECISION SAMPLE(12,23) !nvars,nobs
 DOUBLE PRECISION NSAMPLE(12,23) !nvars,nobs
 DOUBLE PRECISION STDDEV(12,23) !nvars,nobs
 DOUBLE PRECISION BANDWTH(12,23) !nvars,nobs
 DOUBLE PRECISION FACTOR(12,23) !nvars,nobs
 DOUBLE PRECISION MCKBASE(12,31) !nvars,nobs+8

```

DOUBLE PRECISION MCKCOMM(12,31) !nvars,nobs+8
INTEGER KCOUNT(23) !nobs
INTEGER DATE_TIME(8) ! date stamp
CHARACTER*10 BIG_BEN(3) ! time stamp
CHARACTER*20 INVARS(12) !nvars
CHARACTER*20 TAXVARS(12) !nvars
CHARACTER*20 OUTVARS(119) !ovars
INTEGER NVARS ! Number of input variables
INTEGER OVARS ! Number of output variables
INTEGER NSIMS ! Number of simulations per variable
INTEGER NOBS ! Number of observations (years) per simulation
INTEGER KSTYR ! Start year of simulations
INTEGER FTAXYR ! Start year of &TXGF
INTEGER TAXMODE ! Presence and type of tax
INTEGER MCTECH ! Tech Switch: 1=frozen,2=base,3=high,4=hcoal
LOGICAL DIAGS ! Logical switch for diagnostic output
LOGICAL BWRELAX ! Logical test for empty neighborhood
LOGICAL OVERRIDE ! Logical switch for overriding computation
! of local bandwidth
INTEGER SNSIMS ! Starting sim from kernel database
INTEGER ENSIMS ! Ending sim from kernel database

```

C

```

COMMON/MACINTER/ESTIMATR,OUTPUTS,KERNEL,NORMAL,INPUTS,EPMAC,
*   MCDRIO,MCDRII,WEIGHT,SAMPLE,STDDEV,
*   BANDWTH,FACTOR,MCKBASE,MCKCOMM,
*   WPIBY,BPRICE,WPIWT,REGSHRS,EBMAC,
*   EBIND,ECIND,ECNDCH,
*   ESMACREG,ESMAC,EDIND,ESIND,GRACOFFS,
*   GRADELTA,GRAPCTCH,BDRIVER,DDRIVER,PCBPRICE,
*   GRAFLAG,MCPCWGDP,MCIYR,
*   EBINDOLD,PCPASS,INFDSW,
*   PABE,BPABE,TXGF,BTXGF,PCWPI05,
*   MCINDGROW,BMIPET,BPU,MCCAPEEL,MCNMSERV,
*   ESSERV,NEWMAC,GRIFPCNT,USERRATE,GRSWITCH,
*   DDRIVREG,BDRIVREG,RGFDSW,EPREG,ECREG,
*   EBREG,EBREGSUM,MCNMREGDRS,RGADJ,
*   EPMACW,ECMACW,MCNXLAGW,MCNELAGW,
*   NUMINVS,NUMPCS,ESEMP,EDEMP,ECEMP,EBEMP,NUMEMPL,
*   ALTCMACW,ALTPMACW,SWITCHEC,NYRINDX,INVAR,EBINV,
*   KCOUNT,DATE_TIME,
*   ENSIMS,NVARS,OVARS,NSIMS,NOBS,KSTYR,FTAXYR,
*   TAXMODE,SNSIMS,DIAGS,BWRELAX,OVERRIDE,BIG_BEN,
*   INVARS,TAXVARS,OUTVARS

```

Appendix B

MCPARMS Input File

Appendix B

MCPARMS Input File

```
1.29060 ! BPRICE(1) => BASE COAL PRICE (1992) USED IN WPI05
2.54430 ! BPRICE(2) => BASE NAT. GAS PRICE (1992) USED IN WPI05
12.31080 ! BPRICE(3) => BASE ELEC. PRICE USED (1992) IN WPI05
15.05090 ! BPRICE(4) => BASE WOP (1992) USED IN WPI05 CALC.
3.65650 ! BPRICE(5) => BASE PEROLEUM PRICE (1992)USED IN WPI05
1.20337 ! BPRICE(6) => PCWGDP PRICE DEFLATOR (1992), 1987=1.0
0.9500 ! WPIBY(1) => BASE (92) WPI051 PRICE (COAL)
0.7590 ! WPIBY(2) => BASE (92) WPI053 PRICE (N. GAS)
1.2630 ! WPIBY(3) => BASE (92) WPI054 PRICE (ELEC)
0.5800 ! WPIBY(4) => BASE (92) WPI0561 PRICE (WOP)
0.6460 ! WPIBY(5) => BASE (92) WPI057 PRICE (REFINED PETROL)
0.0452 ! WPIWT(1) => COAL SHARE OF WPI05
0.2191 ! WPIWT(2) => NAT GAS SHARE OF WPI05
0.2476 ! WPIWT(3) => ELEC SHARE OF WPI05
0.1527 ! WPIWT(4) => CRUDE PETROL SHARE OF WPI05
0.3354 ! WPIWT(5) => REFINED PETROL SHARE OF WPI05
2 ! MCNELAG => LAGGED ENDOGENOUS
2 ! MCNXLGW => LAGGED EXOGENOUS - WPI
59 ! MCNMFVARS => # OF FINAL DEMAND VARIABLES
35 ! MCNMIND => # OF INDUSTRIAL OUTPUT VARIABLES
10 ! MCNMSERV => # OF SERVICE OUTPUT VARIABLES
1996 ! MCLHISYR => LAST HISTORICAL YEAR IN FORECAST
92 ! MCNMMAC => # OF NON-REGIONAL MACRO VARIABLES
57 ! MCNMMACREG => # OF REGIONAL MACRO VARIABLES
216 ! MCNMINV => # OF CAPITAL EQUIPMENT VARIABLES
29 ! MCNUMMNF => # OF MANUFACTURING INDUSTRIAL OUTPUTS
11 ! MCNUMREGS => # OF CENSUS DIST.S.INCL. CALIF & NATL
20 ! MCNMDRVRS => # OF MACRO DRIVER VARIABLES
14 ! MCNMFLTYPE => # OF COMM FLSPACE TYPES INCL. TOTAL
46 ! NUMEMPL => # OF INDUSTRIAL EMPLOYMENT CATEGORIES
C Base year manufacturing product, MCBIMPRD => BASE YEAR (87) MFG PROD.
168027.2 367500.6 519445.7 158497.7 382163.3 166210.4 180646.7 59548.5 331006.5 1.0
2333047.0
1 ! WPI05 FEEDBACK SWITCH (1=FEEDBACK ON, 0=FEEDBACK OFF)
C Kernel regression
119 ! Number of output variables, OVARS
23 ! Number of observations(years) per simulation, NOBS
1998 ! Start year of simulations, KSTYR
.TRUE. ! Logical switch for diagnostic output written to DIAGS.PRN, DIAGS
.FALSE. ! Logical switch for overriding computation of local bandwidth, OVERRIDE
0.001 ! Value of global bandwidth, BANDWTH
```

Appendix C

Input Variable Description

Appendix C

Input Variable Description

1. &TXGF – Excise Tax Collections.
2. WPI051 – Industrial Steam Coal Price.
3. WPI053 – Industrial Natural Gas Price.
4. WPI054 – Industrial Electricity Price.
5. WPI055 – Utility Natural Gas Price.
6. WPI0561 – World Oil Price.
7. WPI057 – Industrial Average Refined Petroleum Product Price.
8. DTFUELSALLB – Total Demand for Fuels.
9. DENDUSE@COAL – End-Use Demand for Coal.
10. DENDUSE@ELC – End-Use Demand for Electricity.
11. DENDUSE@PET – End-Use Demand for Petroleum.
12. DENDUSE@NG – End-Use Demand for Natural Gas.

Appendix D

Output Variable Description

Appendix D

Output Variable Description

1. CDMV&P92C – Consumption of Motor Vehicles & Parts; bil of chained 92\$.
2. CDFURN@CMP92C – Consumption of Furniture & Household Eq., excl. Computers; bil of chained 92\$.
3. CDCMP - Consumption of Furniture & Household Eq., Computers; bil of \$.
4. CDO92C - Consumption of Other Durables; bil of chained 92\$.
5. CNCS92C - Consumption of Clothing & Shoes; bil of chained 92\$.
6. CNFOOD92C - Consumption of Food; bil of chained 92\$.
7. CNFUEL92C - Consumption of Fuel Oil & Coal; bil of chained 92\$.
8. CNGAS92C - Consumption of Gas & Oil; bil of chained 92\$.
9. CNOO92C - Consumption of Other Nondurables; bil of chained 92\$.
10. CSHOUS92C - Consumption of Housing; bil of chained 92\$.
11. CSHHOPE92C - Consumption of Electricity; bil of chained 92\$.
12. CSHHOPG92C - Consumption of Natural Gas; bil of chained 92\$.
13. CSHHOPO92C - Consumption of Other Household Operations; bil of chained 92\$.
14. CSTRANS92C - Consumption of Transportation Services; bil of chained 92\$.
15. CSO92C – Consumption of Other Services; bil of chained 92\$.
16. CSMED92C - Consumption of Medical Care; bil of chained 92\$.
17. IPDENRAUTO92C - Investment in Non-Res. Durable Eq.: Autos; bil of chained 92\$.
18. IPDENRMCOF - Investment in Non-Res. Durable Eq.: Office; bil of \$.
19. IPDENROTHR92C - Investment in Non-Res. Durable Eq.: Other; bil of chained 92\$.
20. ICNRB&O92C - Investment in Pvt. Struct.: Bldgs. & Other; bil of chained 92\$.
21. ICNRMI&PET92C - Investment in Pvt. Struct.: Mining & Explor; bil of chained 92\$.
22. ICNRPU92C - Investment in Pvt. Struct.: Public Utilities; bil of chained 92\$.
23. IPDER92C - Investment in Producer's Durable Equipment; bil of chained 92\$.
24. ICR92C - Investment in Total Construction; bil of chained 92\$.
25. GFMLCFC92C – National Defense Consumption Expenditures: Fixed Capital; bil of chained 92\$.
26. GFMLCO92C – National Defense Consumption Expenditures: Other; bil of chained 92\$.
27. GFMLGI92C – National Defense Gross Investment; bil of chained 92\$.
28. GFML92C – National Defense Consumption Expenditures and Gross Investment; bil of chained 92\$.
29. GFMLWSS@FAC92C - National Defense Consumption Expenditures: Compensation; bil of chained 92\$.
30. GFOFC92C – Nondefense Consumption Expenditures: Fixed Capital; bil of chained 92\$.
31. GFOCO92C – Nondefense Consumption Expenditures: Other; bil of chained 92\$.
32. GFOGI92C – Federal Nondefense Gross Investment; bil of chained 92\$.
33. GFO92C - Government Spending: Total Non-Defense; bil of chained 92\$.
34. GSL92C - Government Spending: State & Local Purchases; bil of chained 92\$.
35. GFOWSS@FAC92C - Nondefense Consumption Expenditures: Compensation; bil of 92\$.
36. GFONINV92CCH – Nondefense Consumption Expenditures: Nondurables, Inventory Change; bil of 92\$.
37. GSLCFC92C – S&L Consumption Expenditures: Fixed Capital; bil of chained 92\$.
38. GSLCO92C – S&L Consumption Expenditures: Other; bil of chained 92\$.
39. GSLGI92C – State and Local Gross Investment; bil of chained 92\$.
40. GSLWSS@FAC92C – State and Local Consumption Expenditures: Compensation; bil of chained 92\$.
41. EX92CNIA0 - Exports of Food, Feed, and Beverages; bil of chained 92\$.

42. EX92CNIA1 - Exports of Industrial Supplies & Materials; bil of chained 92\$.
43. EX92CNIA2O – Exports of Other Capital Goods Except Autos; bil of chained 92\$.

44. EXNIA2BM – Exports of Computers and Peripherals; bil of \$.
45. EX92CNIA2AC - Exports of Aircraft; bil of chained 92\$.
46. EX92CNIA3 - Exports of Autos; bil of chained 92\$.
47. EX92CNIA4 - Exports of Consumer Goods; bil of chained 92\$.
48. EXD&N92C – Export of Goods; bil of chained 92\$.
49. EXS92C - Exports of Services; bil of chained 92\$.
50. TYF92C - Exports of Factor Income; bil of chained 92\$.
51. M92CNIA0 - Imports of Food, Feed, & Beverages; bil of chained 92\$.
52. M92CNIA1@PET - Imports of Ind. Supp. & Maters., excl. Petro; bil of chained 92\$.
53. M92CNIA100 - Imports of Petroleum & Products; bil of chained 92\$.
54. M92CNIA2@BM - Imports of Capital Goods excl. Autos, Computers & Periph.; bil of chained 92\$.
55. MNIA2BM – Imports of Computers and Peripherals; bil of \$.
56. M92CNIA3 - M87NIA3 Imports of Autos; bil of chained 92\$.
57. M92CNIA4 - Imports of Consumer Goods; bil of chained 92\$.
58. MS92C - Imports of Services; bil of chained 92\$.
59. PAYYF92C - Imports of Factor Income; bil of chained 92\$.
60. INV92CCH - Inventory Change: Total; bil of chained 92\$.
61. GDP92C - Gross Domestic Product; bil of chained 92\$.
62. GDP - Gross Domestic Product; bil of \$.
63. C92C - Personal Consumption Expenditure, Total; bil of chained 92\$.
64. C - Personal Consumption Expenditure, Total; bil of \$.
65. I92C – Gross Private Domestic Investment; bil of chained 92\$.
66. I – Gross Private Domestic Investment; bil of \$
67. IFIXNR92C – Fixed Investment, Nonresidential; bil of chained 92\$.
68. IFIX92C - Total Fixed Investment; bil of chained 92\$.
69. G92C - Government Consumption Expenditures & Gross Investment; bil of chained 92\$.
70. EX92C - Export of Goods & Services; bil of chained 92\$
71. M92C - Imports of Goods & Services; bil of chained 92\$.
72. GNP92C - Gross National Product; bil of chained 92\$.
73. PCWGDP - Implicit Price Defl., GDP; chained, 1992=1.0.
74. GDP92CFE - Full Employment GDP; bil of chained 92\$.
75. PCWEX – Chain-type Price Index, Exports of Goods and Services; 1992=1.0.
76. PCWM – Chain-type Price Index, Imports of Goods and Services; 1992=1.0.
77. RMGBS3NS - Avg. Market Rate on US Govt. 3-Month Bills; %.
78. RMMTGCCNS - Conventional Mortgage Commitment Rate; %.
79. RMPUAANS - Yield on AA Utility Bonds; %.
80. REALRMGBLUS - Real Avg. Yield on 10 year US Govt. Bonds; %
81. ECIWSP - Employment Cost Index, Wages & Salaries; June 1989=1.0.
82. JULCNF - Unit Labor Cost Index, Nonfarm Business Sector; 1982=1.0.
83. SQTRCARSIMP - Unit Sales of Automobiles, Imported; mil of units.
84. SQTRCARS - Unit Sales of Automobiles, Total; mil of units.
85. SQDTRUCKSL - Truck Deliveries, Light Duty; mil of units.
86. SQDTRUCKSH&M - Truck Deliveries, Heavy and Medium Duty; mil of units.
87. RUC - Unemployment Rate, All Civilian Workers; %.
88. WPI - Producer Price Index, All Commodities; 1982=1.0.
89. WPI14 - Producer Price Index, Transportation Equipment; 1982=1.0.
90. WPI11 - Producer Price Index, Machinery and Equipment; 1982=1.0.
91. LC - Civilian Labor Force, Millions of Persons.
92. RMFEDFUNDNS – Effective Rate on Federal Funds.
93. CPI - Consumer Price Index; 1982-84=1.0.
94. YD92C - Disposable Personal Income; bil of chained 92\$.
95. WSD Wage and Salary Disbursements; bil of \$.
96. YP92C - Personal Income; bil of chained 92\$.
97. SHUMBL - Mobile Home Shipments, mil of units.

- 98. HUSTS1 - Single Family Housing Starts; mil of units.
- 99. HUSTS2& - Multi Family Housing Starts; mil of units.
- 100.KQMH - Stock of Mobile Homes; mil of units.
- 101.KQHUSTS1 - Stock of Single Family Housing; mil of units.
- 102.KQHUSTS2& - Stock of Multi Family Housing; mil of units.
- 103.N - Population Include. Armed Forces Overseas; mil of persons.
- 104.N16& - Population Aged 16 and Over; mil of persons.
- 105.MNFWGRT - Manufacturing Wage Rate; \$ per hour.
- 106.NMFFWGRT - Non-Manufacturing Wage Rate; \$ per hour.
- 107.KAMUSE - Commercial Floor Space, Amusement; bil square feet.
- 108.KAUTO - Commercial Floor Space, Auto; bil square feet.
- 109.KDORM - Commercial Floor Space, Dorm; bil square feet.
- 110.KEDUC - Commercial Floor Space, Education; bil square feet.
- 111.KHEALTH - Commercial Floor Space, Health; bil square feet.
- 112.KHOTEL - Commercial Floor Space, Hotel; bil square feet.
- 113.KMFG - Commercial Floor Space, Manufacturing; bil square feet.
- 114.KMISCNR - Commercial Floor Space, Misc. Non-Residential; bil square feet.
- 115.KOFFICE - Commercial Floor Space, Office; bil square feet.
- 116.KPUB - Commercial Floor Space, Public; bil square feet.
- 117.KREL - Commercial Floor Space, Religion; bil square feet.
- 118.KSTORES - Commercial Floor Space, Stores; bil square feet.
- 119.KWARE - Commercial Floor Space, Warehouse; bil square feet.

Appendix E

**Listing of
Subroutine
READMAC**

Appendix E

Listing of Subroutine READMAC

```
C*****
SUBROUTINE READMAC
C*****
C The READMAC subroutine reads many separate files for parameters,
C coefficients, and for baseline and adjustment values used by the
C Macroeconomic Activity Module (MAM). These files only need to be
C read once in a run, so this subroutine is executed for the first
C iteration of the first year. Files read include:
C
C mcparms - parameter values of Macroeconomic Activity Module.
C mcinadj - adjustment values for macroeconomic driver variables.
C mcibass - baseline values for macroeconomic driver variables.
C mcecind - coefficients for industrial output.
C mcecmac - coefficients for non-regional and regional macroeconomic
C variables.
C mcecomp - coefficients for industrial employment.
C mcbass - baseline values for non-regional and regional macroeconomic
C variables, industrial output, and employment.
C mcrgbas - baseline values for regional macroeconomic variables and
C industrial outputs but not services.
C mcinv - baseline values for investment including rental cost of
C capital equipment, investment in capital equipment, capital
C equipment stock and price index for investment in capital
C equipment.
C
C Four files (mcibass, mcbass, mcrgbas and mcinv) have separate values
C for each of three possible scenerios: high, moderate, and low
C economic growth.
C*****

      IMPLICIT NONE

C This subroutine is executed for the first iteration of the first year.

C Include parameter files.
      INCLUDE(PARAMETR)
      INCLUDE(MACPARM)
      INCLUDE(MCINTER2)
      INCLUDE(NCNTRL)

      external getindex,rtovalue
      integer getindex,rtovalue

      INTEGER   FILE_MGR
      EXTERNAL  FILE_MGR
```

CHARACTER*18 NAME ! File name for FILE_MGR

LOGICAL NEW ! Logical variable for FILE_MGR
INTEGER INUNIT ! Unit number for reads as returned from FILE_MGR
INTEGER I,J,K ! Indexes for arrays

C Open macroeconomic parameters file mcparms for read.

C File mcparms is always read first.

NAME='MCPARMS'

NEW=.FALSE.

INUNIT=FILE_MGR('O',NAME,NEW)

C Read six base prices used for aggregate WPI05 index.

DO 200 I = 1, 6

200 READ(INUNIT,2000) BPRICE(I)

C Read five WPI base prices used for aggregate WPI05 index.

DO 220 I = 1, 5

220 READ(INUNIT,2000) WPIBY(I)

C Read five WPI shares used for aggregate WPI05 index.

DO 230 I = 1, 5

230 READ(INUNIT,2000) WPIWT(I)

C Read number of lags on endogenous variables.

READ(INUNIT,2010) MCNELAG

C Read number of lags on endogenous and exogenous variables

C used to estimate WPI model.

READ(INUNIT,2030) MCNXLAGW

C Read number of final demand variables.

READ(INUNIT,2010) MCNMFDVARS

C Read number of industrial output variables.

READ(INUNIT,2010) MCNMIND

C Read number of service output variables.

READ(INUNIT,2010) MCNMSERV

C Read last historical year in forecast.

READ(INUNIT,2040) MCLHISYR

C Read number of non-regional macro variables.

READ(INUNIT,2010) MCNMMAC

C Read number of regional macro variables.

READ(INUNIT,2010) MCNMMACREG

C Read number of investment macro variables.

READ(INUNIT,2020) MCNMINV

C Read number of manufacturing industrial outputs.

READ(INUNIT,2010) MCNUMMNF

C Read number of Census districts plus California

C and national.
 READ(INUNIT,2010) MCNUMREGS

C Read number of macro driver variables.
 READ(INUNIT,2010) MCNMDRVRS

C Read number of commercial floorspace types including total.
 READ(INUNIT,2010) MCNMFLTYPE

C Read number of industrial employment categories.
 READ(INUNIT,2010) NUMEMPL

C Read base year manufacturing product.
 READ(INUNIT,2050) (MCBIMPRD(I), I= 1, 11)

C Read feedback switch for WPI05.
 READ(INUNIT,2060) INFDSW

C Kernel regression - Read presence and type of tax
 TAXMODE = RTOVALUE('MACTAX ',0)

C Kernel regression - World oil price and policy sims have same input variables
 NVAR = 12
 INVAR = TAXVAR
 IF (TAXMODE .EQ. 0) THEN
 NSIMS = 24
 ELSE
 NSIMS = 54
 ENDIF

C Kernel regression - Read first year of tax
 FTAXYR = RTOVALUE('MACTXYR ',2005)

C Kernel regression - Read number of output variables
 READ(INUNIT,2070) OVARS

C Kernel regression - Read number of observations (years) per simulation
 READ(INUNIT,2070) NOBS

C Kernel regression - Read start year of simulations
 READ(INUNIT,2070) KSTYR

C Kernel regression - Read logical switch for diagnostic output
 READ(INUNIT,2071) DIAGS

C Kernel regression - Read logical switch for local bandwidth override
 READ(INUNIT,2071) OVERRIDE

C Kernel regression - Read global bandwidth
 READ(INUNIT,2072) BANDWTH(1,1)
 BANDWTH = BANDWTH(1,1)

C Read formats for mcparms file.
 2000 FORMAT(F10.0)

```

2010 FORMAT(I2)
2020 FORMAT(I3)
2030 FORMAT(5(I2,1X))
2040 FORMAT(I4)
2050 FORMAT(/,11(F9.1,1X))
2060 FORMAT(I1,/)
2070 FORMAT(I4)
2071 FORMAT(L7)
2072 FORMAT(F5.3)

```

```

C Close mcparms file for read.
  INUNIT=FILE_MGR('C',NAME,NEW)

C Open macroeconomic adjustment file mcinadj for read.
  NAME='MCINADJ'
  INUNIT=FILE_MGR('O',NAME,NEW)

C Read extra lines at top of file.
  READ(INUNIT,2080)

C Read first eight obs, 1993 - 2000, for wholesale prices:
C PCLIN - price of industrial coal.
C PNGIN - price of industrial natural gas.
C PELIN - price of industrial electricity.
C IT_WOP - world oil price.
C PTPIN - price of industrial total petroleum.
C WPI05 - aggregate energy price index.
  READ(INUNIT,2090) ((MCADJ(I,J), J=4,11), I= 1,6)

C Read next eight obs, 2001 - 2008, for same.
  READ(INUNIT,2090) ((MCADJ(I,J), J=12,19), I= 1,6)

C Read next eight obs, 2009 - 2016, for same.
  READ(INUNIT,2090) ((MCADJ(I,J), J=20,27), I= 1,6)

C Read last four obs, 2017 - 2020, for same.
  READ(INUNIT,2100) ((MCADJ(I,J), J=28,31), I= 1,6)

C Read formats for mcinadj file.
2080 FORMAT(///)
2090 FORMAT(/,6(8X,8(1X,F6.3)/))
2100 FORMAT(/,6(8X,4(1X,F6.3)/))

C Close mcinadj file for read.
  INUNIT=FILE_MGR('C',NAME,NEW)

C Test for assumed level of macroeconomic activity.
  IF (MMAC .EQ. 1) THEN
    NAME='MCIBASS(LOW)'
  ELSEIF (MMAC .EQ. 3) THEN
    NAME='MCIBASS(HIGH)'
  ELSE
    NAME='MCIBASS(MID)'
  ENDIF

```

C Open macroeconomic baseline driver variable file mcibass for read.
 INUNIT=FILE_MGR('O',NAME,NEW)

C Read first line.
 READ(INUNIT,*)

C Read baseline driver time series.
 READ(INUNIT,2240) ((BDRIVER(I,J), J=1,NYRINDX), I=1,MCNMDRVRS)
 2240 FORMAT(<NYRINDX>(F15.8,1X))

C Close file mcibass for read.
 INUNIT=FILE_MGR('C',NAME,NEW)

C Open industrial output coefficient file mcecind for read.
 NAME='MCECIND'
 INUNIT=FILE_MGR('O',NAME,NEW)

C Read industrial output coefficients.
 READ(INUNIT,2250) ((ECIND(I,J), J=1,MCNMFDVARS), I=1,MCNMIND
 * +MCNMSERV)
 READ(INUNIT,2250) ((ECNDCH(I,J), J=1,MCNMFDVARS), I=1,MCNMIND
 * +MCNMSERV)
 2250 FORMAT(59(F10.0,1X))

C Close file mcecind for read.
 INUNIT=FILE_MGR('C',NAME,NEW)

C Open macroeconomic coefficient file for read.
 C NAME='MCECMAC'
 C INUNIT=FILE_MGR('O',NAME,NEW)

C Read wholesale price coefficients for RSM specification:
 C y regressed on WPI05{0 to 2} and y{1 to 2}.
 C Add three to number of non-regional and regional variables
 C for price deflators.
 C READ(INUNIT,2280) ((ECMACW(I,J),J=1,MCNXLAGW+MCNELAG+1),
 C * I=1,MCNMMAC+MCNMMACREG)
 C2280 FORMAT(5(F8.0,1X))

C Read blank line.
 C READ(INUNIT,*)

C Read wholesale price coefficients for alternative RSM specification:
 C y regressed on WPI05{0 to 5}.
 C Add three to number of non-regional and regional variables
 C for price deflators. An additional one is added to adjust the
 C number of lagged exogenous and endogenous variables for the
 C alternative specification.
 C READ(INUNIT,2282) ((ALTCMACW(I,J),J=1,MCNXLAGW+MCNELAG+1+1),
 C * I=1,MCNMMAC+MCNMMACREG)
 C2282 FORMAT(6(F8.0,1X))

C Close file mcecmac for read.

```

C   INUNIT=FILE_MGR('C',NAME,NEW)

C   Open industrial employment coefficient file mcecomp for read.
    NAME='MCECEMP'
    INUNIT=FILE_MGR('O',NAME,NEW)

C   Read industrial employment coefficients.
C
C   Add 2 to number of employment categories for breakdown
C   of services and agriculture.
    DO 260 I=1,NUMEMPL+2
      READ(INUNIT,2310)(ECEMP(I,J),J=1,3)
    260 CONTINUE
    2310 FORMAT(3(1X,F13.0))

C   Close file mcecomp for read.
    INUNIT=FILE_MGR('C',NAME,NEW)

C   Test for assumed level of macroeconomic activity.
    IF (MMAC .EQ. 1) THEN
      NAME='MCBASS(LOW)'
    ELSEIF (MMAC .EQ. 3) THEN
      NAME='MCBASS(HIGH)'
    ELSE
      NAME='MCBASS(MID)'
    ENDIF

C   Open macroeconomic baseline variable file mcbass for read.
    INUNIT=FILE_MGR('O',NAME,NEW)

C   Read macroeconomic variable baseline values.
C
C   Add three to number of non-regional and regional variables
C   for price deflators.
    READ(INUNIT,2315) ((EBMAC(I,J),J=1,NYRINDX),
*                   I=1,MCNMMAC+MCNMMACREG)

C   Read blank line.
    READ(INUNIT,2330)

C   Read industrial output baseline values.
    READ(INUNIT,2315) ((EBIND(I,J),J=1,NYRINDX),
*                   I=1,MCNMIND+MCNMSERV)

C   Read blank line.
    READ(INUNIT,2330)

C   Read employment baseline values.
    READ(INUNIT,2315) ((EBEMP(I,J),J=1,NYRINDX),I=1,NUMEMPL)

C   Read formats for mcbass file.
    2315 FORMAT(<NYRINDX>(1X,E17.10))
    2330 FORMAT()

```

```

C Close mcbass file for read.
  INUNIT=FILE_MGR('C',NAME,NEW)

C Test for assumed level of macroeconomic activity.
  IF (MMAC .EQ. 1) THEN
    NAME='MCRGBAS(LOW)'
  ELSEIF (MMAC .EQ. 3) THEN
    NAME='MCRGBAS(HIGH)'
  ELSE
    NAME='MCRGBAS(MID)'
  ENDIF

C Open regional baseline values file mcrgbas for read.
  INUNIT=FILE_MGR('O',NAME,NEW)

C Read regional baseline values.
  DO 270 K = 1,MCNMMACREG+MCNMIND
    READ(INUNIT,2340) ((EBREG(K,I,J),J=1,NYRINDX),I=1,MCNUMREGS-2)
  270 CONTINUE
  2340 FORMAT(<NYRINDX>(1X,E17.10))

C Close mcrgbas file for read.
  INUNIT=FILE_MGR('C',NAME,NEW)

C Open switch file for alternative wholesale price coefficients.
C   NAME='MCSWTCH'
C   INUNIT=FILE_MGR('O',NAME,NEW)

C Read wholesale price coefficient switch values.
C   READ(INUNIT,2352) (SWITCHEC(I),I=1,MCNMMAC+MCNMMACREG)
C2352 FORMAT(////,148(I2,/))

C Close mcswtch file for read.
C   INUNIT=FILE_MGR('C',NAME,NEW)

C Test for assumed level of macroeconomic activity.
  IF (MMAC .EQ. 1) THEN
    NAME='MCINVST(LOW)'
  ELSEIF (MMAC .EQ. 3) THEN
    NAME='MCINVST(HIGH)'
  ELSE
    NAME='MCINVST(MID)'
  ENDIF

C Open macroeconomic investment baseline variable file mcinvst for read.
  INUNIT=FILE_MGR('O',NAME,NEW)

C Read macroeconomic investment variable baseline values.
  READ(INUNIT,2353) ((EBINV(I,J),J=1,NYRINDX),I=1,MCNMINV)

C Read formats for mcbass file.
  2353 FORMAT(<NYRINDX>(F12.0,1X))

C Close mcbass file for read.

```

```
INUNIT=FILE_MGR('C',NAME,NEW)
```

C Kernel regression - FILE OPEN AND READ

C Open and read file(s) containing input variables.

C Simulations are grouped by variable. Sims are across rows, and time

C is across the columns. The input variable values along a row are in

C scientific notation and separated by a space. Input variable values

C are not written to an output file.

```
NAME='MCDRIO'
```

```
INUNIT=FILE_MGR('O',NAME,NEW)
```

```
READ(INUNIT,*)
```

```
READ(INUNIT,5009) ((MCDRIO(I,K),K=1,NOBS),I=1,OVARS)
```

```
INUNIT=FILE_MGR('C',NAME,NEW)
```

```
5009 FORMAT(<NOBS>(1X,E17.10))
```

C Kernel regression - Open and read file(s) containing input variables.

C Simulations are grouped by variable. Sims are across rows, and time

C is across the columns. The input variable values along a row are in

C scientific notation and separated by a space. Input variable values

C are not written to an output file.

```
IF (TAXMODE .EQ. 0) THEN
```

```
NAME='MCPCOFI'
```

```
FTAXYR = 0
```

```
ELSEIF (TAXMODE .EQ. 1) THEN
```

```
NAME='MCDRI'
```

```
ELSEIF (TAXMODE .EQ. 2) THEN
```

```
NAME='MCPTI'
```

```
ELSEIF (TAXMODE .EQ. 3) THEN
```

```
NAME='MCCTI'
```

```
ELSEIF (TAXMODE .EQ. 4) THEN
```

```
NAME='MCSSI'
```

```
ELSE
```

```
WRITE(6,*) 'ERROR! ',TAXMODE,' is an invalid argument for '
```

```
*'TAXMODE'
```

```
STOP ''
```

```
ENDIF
```

```
INUNIT=FILE_MGR('O',NAME,NEW)
```

```
DO 5010 J=1,NSIMS
```

```
READ(INUNIT,*)
```

```
READ(INUNIT,5020) ((INPUTS(I,J,K),K=1,NOBS),I=1,NVARS)
```

```
5010 CONTINUE
```

```
INUNIT=FILE_MGR('C',NAME,NEW)
```

```
5020 FORMAT(<NOBS>(1X,E17.10))
```

C Kernel regression - Open and read file(s) containing output variables.

C Output variables are grouped by simulation. Within a simulation,

C output variables are across rows, and time

C is across the columns. The output variable values along a row are in

C scientific notation and separated by a space. Output variable values

C are not written to an output file.

```
IF (TAXMODE .EQ. 0) THEN
```

```
NAME='MCPCOFO'
```

```
ELSEIF (TAXMODE .EQ. 1) THEN
```

```

    NAME='MCDRO'
ELSEIF (TAXMODE .EQ. 2) THEN
    NAME='MCPTO'
ELSEIF (TAXMODE .EQ. 3) THEN
    NAME='MCCTO'
ELSEIF (TAXMODE .EQ. 4) THEN
    NAME='MCSSO'
ENDIF

INUNIT=FILE_MGR('O',NAME,NEW)
DO 5022 J=1,NSIMS
    READ(INUNIT,*)
    READ(INUNIT,5030) ((OUTPUTS(I,J,K),K=1,NOBS),I=1,OVARS)
5022 CONTINUE
    INUNIT=FILE_MGR('C',NAME,NEW)
5030 FORMAT(<NOBS>(1X,E17.10))

```

C Kernel regression - Open and read file containing NEMS base values for input variables.
C Input variables are across the rows and time is across the columns.
C The input variable values are in scientific notation and separated
C by a space.
C Test for assumed level of macroeconomic activity.

```

    IF (MMAC .EQ. 1) THEN
        NAME='MCKBASE(LOW)'
    ELSEIF (MMAC .EQ. 3) THEN
        NAME='MCKBASE(HIGH)'
    ELSE
        NAME='MCKBASE(MID)'
    ENDIF

INUNIT=FILE_MGR('O',NAME,NEW)
READ(INUNIT,*)
READ(INUNIT,5041) ((MCKBASE(I,K),K=1,NYRINDX),I=1,12)
INUNIT=FILE_MGR('C',NAME,NEW)
5041 FORMAT(<NYRINDX>(1X,E17.10))

```

C Kernel regression - Open and read file(s) containing input variables.
C Simulations are grouped by variable. Sims are across rows, and time
C is across the columns. The input variable values along a row are in
C scientific notation and separated by a space. Input variable values
C are not written to an output file.

```

    NAME='MCDRII'
INUNIT=FILE_MGR('O',NAME,NEW)
READ(INUNIT,*)
READ(INUNIT,5042) ((MCDRII(I,K),K=1,NOBS),I=1,12)
INUNIT=FILE_MGR('C',NAME,NEW)
5042 FORMAT(<NOBS>(1X,E17.10))

```

C End of subroutine READMAC. Return to calling subroutine MAC.
RETURN
END

Appendix F

**Listing of
Subroutine
MCKCOMMSUB**

Appendix F

Listing of Subroutine MCKCOMMSUB

```
C*****
SUBROUTINE MCKCOMMSUB
C*****
C The MCKCOMMSUB subroutine applies the change in input variables
C from the NEMS's base to that for the DRI base. This is a
C kernel regression subroutine.
C*****

      IMPLICIT NONE
      INTEGER I

C Include parameter files.
      INCLUDE (PARAMETR)
      INCLUDE (NCNTRL)
      INCLUDE (MACOUT)
      INCLUDE (MACREP)
      INCLUDE (MACPARM)
      INCLUDE (MCINTER2)
      INCLUDE (APQ)
      INCLUDE (INTOUT)
      INCLUDE (CAPEXP)
      INCLUDE (EMISSION)
      INCLUDE (RESDREP)
      INCLUDE (NGTDMREP)

C Compute input variables for current NEMS's run.
      MCKCOMM(1,MCIYR) = EMREV(1,MCIYR) + EMREV(2,MCIYR)
      * + EMREV(3,MCIYR) + EMREV(4,MCIYR) + EMREV(5,MCIYR)
      MCKCOMM(2,MCIYR) = PCLEL(11,MCIYR)*MC_PCWGDP(MCIYR)
      MCKCOMM(3,MCIYR) = OGWPRNG(13,MCIYR)*MC_PCWGDP(MCIYR)
      MCKCOMM(4,MCIYR) = (PELRS(11,MCIYR) + PELCM(11,MCIYR)
      * + PELIN(11,MCIYR) + PELTR(11,MCIYR))*MC_PCWGDP(MCIYR)/4.0
      MCKCOMM(5,MCIYR) = (PNGRS(11,MCIYR) + PNGCM(11,MCIYR)
      * + PNGIN(11,MCIYR) + PNGTR(11,MCIYR) + PNGEL(11,MCIYR))
      * *MC_PCWGDP(MCIYR)/5.0
      MCKCOMM(6,MCIYR) = IT_WOP(MCIYR,1)*MC_PCWGDP(MCIYR)
      MCKCOMM(7,MCIYR) = (PTPRS(11,MCIYR) + PDSCM(11,MCIYR)
      * + PRSCM(11,MCIYR) + PDSIN(11,MCIYR) + PRSIN(11,MCIYR)
      * + PDSTR(11,MCIYR) + PJFTR(11,MCIYR) + PMGTR(11,MCIYR)
      * + PRSTR(11,MCIYR))*MC_PCWGDP(MCIYR)/9.0
      MCKCOMM(8,MCIYR) = QTPAS(11,MCIYR)
      * + QNGAS(11,MCIYR) + QGPTR(11,MCIYR) + QLPIN(11,MCIYR)
      * + QCLAS(11,MCIYR) + QMCIN(11,MCIYR) + QCIIN(11,MCIYR)
      * + QUREL(11,MCIYR)
      * + QTRAS(11,MCIYR) - QSTRS(11,MCIYR) - QGERS(11,MCIYR)
```

* - QSTCM(11,MCIYR)

```

* + QEIEL(11,MCIYR) + QMETR(11,MCIYR) + QHYTR(11,MCIYR)
* - QTREL(11,MCIYR) - QPCEL(11,MCIYR)
MCKCOMM(9,MCIYR) = QMCIN(11,MCIYR)
* + QCLAS(11,MCIYR) - QCLEL(11,MCIYR)
* + QCIIN(11,MCIYR)
MCKCOMM(10,MCIYR) = QELAS(11,MCIYR)
MCKCOMM(11,MCIYR) = QDSAS(11,MCIYR) - QDSEL(11,MCIYR)
* + QKSAS(11,MCIYR) + QJFTR(11,MCIYR) + QLGAS(11,MCIYR)
* + QMGAS(11,MCIYR) + QPFIN(11,MCIYR)
* + QRSAS(11,MCIYR) - QRSEL(11,MCIYR)
* + QOTAS(11,MCIYR) + QSGIN(11,MCIYR) + QPCIN(11,MCIYR)
* + QASIN(11,MCIYR)
MCKCOMM(12,MCIYR) = QNGAS(11,MCIYR) + QGPTR(11,MCIYR)
* + QLPIN(11,MCIYR) - QNGEL(11,MCIYR)

```

C Compute the input variables by applying the change from the NEMS's
C base to the DRI base.

```

IF (MCIYR .GT. 8) THEN
WRITE(14,*) 'MCIYR ',MCIYR,'CURITR ',CURITR
WRITE(14,*) '*****'
WRITE(14,*) '*****'

SAMPLE(1,MCIYR-8) = MCKCOMM(1,MCIYR)
SAMPLE(2,MCIYR-8) = ((MCKCOMM(2,MCIYR)/MCKBASE(2,MCIYR)))
* *MCDRII(2,MCIYR-8)
SAMPLE(3,MCIYR-8) = ((MCKCOMM(3,MCIYR)/MCKBASE(3,MCIYR)))
* *MCDRII(3,MCIYR-8)
SAMPLE(4,MCIYR-8) = ((MCKCOMM(4,MCIYR)/MCKBASE(4,MCIYR)))
* *MCDRII(4,MCIYR-8)
SAMPLE(5,MCIYR-8) = ((MCKCOMM(5,MCIYR)/MCKBASE(5,MCIYR)))
* *MCDRII(5,MCIYR-8)
SAMPLE(6,MCIYR-8) = ((MCKCOMM(6,MCIYR)/MCKBASE(6,MCIYR)))
* *MCDRII(6,MCIYR-8)
SAMPLE(7,MCIYR-8) = ((MCKCOMM(7,MCIYR)/MCKBASE(7,MCIYR)))
* *MCDRII(7,MCIYR-8)
SAMPLE(8,MCIYR-8) = ((MCKCOMM(8,MCIYR)/MCKBASE(8,MCIYR)))
* *MCDRII(8,MCIYR-8)
SAMPLE(9,MCIYR-8) = ((MCKCOMM(9,MCIYR)/MCKBASE(9,MCIYR)))
* *MCDRII(9,MCIYR-8)
SAMPLE(10,MCIYR-8) = ((MCKCOMM(10,MCIYR)
* /MCKBASE(10,MCIYR))*MCDRII(10,MCIYR-8)
SAMPLE(11,MCIYR-8) = ((MCKCOMM(11,MCIYR)
* /MCKBASE(11,MCIYR))*MCDRII(11,MCIYR-8)
SAMPLE(12,MCIYR-8) = ((MCKCOMM(12,MCIYR)
* /MCKBASE(12,MCIYR))*MCDRII(12,MCIYR-8)

```

C Write to file the input variables.

```

WRITE(14,*)
WRITE(14,*) 'MC_PCWGDP ',MC_PCWGDP(MCIYR)
DO 9001 I=1,12
WRITE(14,*) 'NVAR ',TAXVARS(I)
WRITE(14,*) 'MCKCOMM ',MCKCOMM(I,MCIYR),
* 'SAMPLE ',SAMPLE(I,MCIYR-8)
WRITE(14,*) 'MCKBASE ',MCKBASE(I,MCIYR),

```

```
* 'MCDRII ',MCDRII(I,MCIYR-8)
9001 CONTINUE
```

```
ENDIF
```

```
C End of subroutine MCKCOMMSUB. Return to calling subroutine MAC.
RETURN
END
```

Appendix G

**Listing of
Subroutine
KERNELSUB**

Appendix G

Listing of Subroutine KERNELSUB

```
C*****
SUBROUTINE KERNELSUB
C*****
C The KERNELSUB subroutine contains the kernel regression.
C*****

      IMPLICIT NONE

C Include parameter files.
      INCLUDE (PARAMETR)
      INCLUDE (NCNTRL)
      INCLUDE (MACPARM)
      INCLUDE (MCINTER2)
      INTEGER I,J,K

      external getindex,rtovalue
      integer getindex,rtovalue

      IF ((BASEYR+CURIYR-1) .EQ. 1998 .AND. CURITR .EQ. 1) THEN

C Define relevant simulations
C Read technology switch value: 1=frozen,2=base,3=high,4=hcoal
      MCTECH = RTOVALUE('MACTECH ',2)
      IF (FTAXYR .EQ. 0 .AND. WWOP .EQ. 1) THEN ! LOW WOP
         SNSIMS = 1
         ENSIMS = 8
      ELSEIF (FTAXYR .EQ. 0 .AND. WWOP .EQ. 2) THEN ! BASE
         SNSIMS = 9
         ENSIMS = 16
      ELSEIF (FTAXYR .EQ. 0 .AND. WWOP .EQ. 3) THEN ! HIGH WOP
         SNSIMS = 17
         ENSIMS = 24
      ELSEIF (FTAXYR .EQ. 2000) THEN !2000 SIMS
         SNSIMS = 1
         ENSIMS = 11
      ELSEIF (FTAXYR .EQ. 2005 .AND. MCTECH .EQ. 4) THEN !HCOAL SIMS
         SNSIMS = 55
         ENSIMS = 61
      ELSEIF (FTAXYR .EQ. 2005 .AND. MCTECH .EQ. 2) THEN !BASE SIMS
         SNSIMS = 12
         ENSIMS = 33
      ELSEIF (FTAXYR .EQ. 2005 .AND. MCTECH .EQ. 1) THEN !FRZEN SIMS
         SNSIMS = 47
         ENSIMS = 54
      ELSEIF (FTAXYR .EQ. 2010) THEN !2010 SIMS
```

SNSIMS = 34

```

    ENSIMS = 39
ELSEIF (FTAXYR .EQ. 2012) THEN !DELAY SIMS
    SNSIMS = 40
    ENSIMS = 46
ELSE
    WRITE(6,*) 'ERROR! ',FTAXYR,' is an invalid start year ',
*'for &TXGF.'
    CLOSE (14)
    STOP ''
ENDIF

```

C Compute a mean for every input variable at each period t. Computed means
C are written to the DIAGS.PRN file.

```

    DO 5050 I=1,NVARS
    DO 5060 K=1,NOBS
        INPUTS(I,NSIMS+1,K) = 0.0 ! Initialize
        DO 5070 J=SNSIMS,ENSIMS
            INPUTS(I,NSIMS+1,K) = INPUTS(I,NSIMS+1,K)+INPUTS(I,J,K) ! Aggregate
5070    CONTINUE
        INPUTS(I,NSIMS+1,K) = INPUTS(I,NSIMS+1,K)
        * / (ENSIMS-SNSIMS+1) ! Compute mean
5060    CONTINUE
5050    CONTINUE

```

C Normalize inputs using the computed means. Normalized input variable
C values are written to the DIAGS.PRN file.

```

    DO 5080 I=1,NVARS
    DO 5090 K=1,NOBS
        DO 50100 J=SNSIMS,ENSIMS
            IF (INPUTS(I,NSIMS+1,K) .NE. 0.0) THEN
                NORMAL(I,J,K) = INPUTS(I,J,K)/INPUTS(I,NSIMS+1,K)
            ELSE
                NORMAL(I,J,K) = 0.0
            ENDIF
50100    CONTINUE
5090    CONTINUE
5080    CONTINUE

```

C Compute mean of normalized value. Result is 1.0 if inputs are
C normalized properly. Results of this check are written to the
C DIAGS.PRN file.

```

    DO 50130 I = 1,NVARS
    DO 50140 K = 1,NOBS
        NORMAL(I,NSIMS+1,K) = 0.0 ! Initialize
        DO 50150 J = SNSIMS,ENSIMS
            NORMAL(I,NSIMS+1,K) = NORMAL(I,NSIMS+1,K)
            * +NORMAL(I,J,K) ! Aggregate
50150    CONTINUE
        NORMAL(I,NSIMS+1,K) = NORMAL(I,NSIMS+1,K)
        * /DBLE(ENSIMS-SNSIMS+1) ! Compute
50140    CONTINUE
50130    CONTINUE

```

KERNEL = -1.0 ! Initialize for sims outside a neighborhood

```

    FACTOR = 1.0 ! Initialize bandwidth adjustment
    KCOUNT = 0 ! Initialize
    WEIGHT = 0.0 ! Initialize
    ESTIMATR = 0.0 ! Initialize
    EPMAC = 0.0 ! Initialize

C End IF conditional testing for current year equals 1998 and
C current iteration equals one.
    ENDIF

    IF ((BASEYR+CURIYR-1) .GE. 1998) THEN

C Compute an initial value of the local bandwidths. This block of code
C is executed if not overridden with a global bandwidth. It begins by
C computing the standard deviation of the normalize input variables at
C each period t. The computed standard deviations are written to the
C DIAGS.PRN file. The computed bandwidth is an increasing function of the
C standard deviation and a decreasing function of the number of simulation,
C NSIMS. The initial values of the local bandwidth are not written to an
C output file.
        DO 50179 I=1,NVARS
            STDDEV(I,MCIYR-8) = 0.0 ! Initialize
50179    CONTINUE

        IF (.NOT. OVERRIDE) THEN
            DO 50160 I = 1,NVARS
                DO 50180 J = SNSIMS,ENSIMS
                    STDDEV(I,MCIYR-8) = STDDEV(I,MCIYR-8)          ! Aggregate the squared deviations
                    *      +(NORMAL(I,J,MCIYR-8)-NORMAL(I,NSIMS+1,MCIYR-8))**2 ! from mean
50180    CONTINUE
                    STDDEV(I,MCIYR-8) = ((1.0/(DBLE(ENSIMS-SNSIMS+1)-1.0))
                    *      *STDDEV(I,MCIYR-8))**0.5 ! Compute standard deviation
50160    CONTINUE
                DO 50190 I = 1,NVARS
                    BANDWTH(I,MCIYR-8) = 1.06*STDDEV(I,MCIYR-8)
                    *      *(DBLE(ENSIMS-SNSIMS+1)**(-0.2)) ! Compute local bandwidth
50190    CONTINUE

                WRITE(14,*)
                WRITE(14,*) 'SNSIMS ',SNSIMS,'ENSIM ',ENSIMS,
                    *      'NSIMS ',(ENSIMS-SNSIMS+1)
                DO 9002 I=1,12
                    WRITE(14,*) 'NVAR ',TAXVARS(I)
                    WRITE(14,*) 'STDDEV ',STDDEV(I,MCIYR-8),
                    *      'BANDWTH ',BANDWTH(I,MCIYR-8)
9002    CONTINUE
                ENDIF

C Normalize sample inputs using the computed means. Normalized sample
C input variable values are written to the DIAGS.PRN file.
        DO 50110 I=1,NVARS
            IF (INPUTS(I,NSIMS+1,MCIYR-8) .NE. 0.0) THEN
                NSAMPLE(I,MCIYR-8) = SAMPLE(I,MCIYR-8)
                *      /INPUTS(I,NSIMS+1,MCIYR-8)

```

```

ELSE
  NSAMPLE(I,MCIYR-8) = 0.0
ENDIF
50110 CONTINUE

```

```

WRITE(14,*)
DO 9003 I=1,12
  WRITE(14,*) 'NVAR ',TAXVARS(I)
  WRITE(14,*) 'NSAMPLE ',NSAMPLE(I,MCIYR-8),
* 'SAMPLE ',SAMPLE(I,MCIYR-8),
* 'INPUTS ',INPUTS(I,NSIMS+1,MCIYR-8)
9003 CONTINUE
WRITE(14,*)

```

C Compute kernels. This block computes a kernel for each simulation at C period t for every input variable given the bandwidth. If in period C t no kernels result, FACTOR is used to increase upward the bandwidth. C This continues until the bandwidth exceeds 1.0. At that point, the C program writes an error to screen and terminates. The values of KERNEL C and the final values of FACTOR and BANDWTH are written to the DIAGS.PRN C file.

```

DO 50207 I = 1,NVARS+1
  DO 50208 J = 1,NSIMS+1
    KERNEL(I,J,MCIYR-8) = -1.0 ! Initialize for sims outside a neighborhood
50208 CONTINUE
50207 CONTINUE

```

```

DO 50209 I = 1,NVARS
  FACTOR(I,MCIYR-8) = 1.0 ! Initialize bandwidth adjustment
50209 CONTINUE

```

```

DO 50210 I=1,NVARS
  BWRELAX = .TRUE. ! Initialize logical test for bandwidth adjustment
  DO 50230 WHILE (BWRELAX)
    DO 50240 J=SNSIMS,ENSIMS
      IF (BANDWTH(I,MCIYR-8) .EQ. 0.0) THEN
        BANDWTH(I,MCIYR-8) = 0.001
      ENDIF
      IF (ABS((NSAMPLE(I,MCIYR-8)-NORMAL(I,J,MCIYR-8))
* /BANDWTH(I,MCIYR-8)) .LE. 1.0) THEN ! Test if sim is inside the neighborhood
      IF (KERNEL(NVARS+1,J,MCIYR-8) .EQ. -1.0) THEN
        KERNEL(NVARS+1,J,MCIYR-8) = 1.0 ! Initialize product of bandwidths
      ENDIF
      KERNEL(I,J,MCIYR-8) =
* (.75/BANDWTH(I,MCIYR-8))
* *(1.0-(((NSAMPLE(I,MCIYR-8)-NORMAL(I,J,MCIYR-8))
* /BANDWTH(I,MCIYR-8))**2)) ! Compute kernel
      KERNEL(NVARS+1,J,MCIYR-8) = KERNEL(NVARS+1,J,MCIYR-8)
* *KERNEL(I,J,MCIYR-8) ! Compute kernel product
      BWRELAX = .FALSE. ! Bandwidth adjustment not necessary
    ENDIF
  ENDIF

```

```

WRITE(14,*) 'NVAR ',TAXVARS(I),'SIM ',J
WRITE(14,*) 'KERNEL ',KERNEL(I,J,MCIYR-8)

```

```

        WRITE(14,*) 'NSAMPLE ',NSAMPLE(I,MCIYR-8),
*       'NORMAL ', NORMAL(I,J,MCIYR-8),
*       'BANDWTH ',BANDWTH(I,MCIYR-8)

50240   CONTINUE
        IF (BWRELAX) THEN
            FACTOR(I,MCIYR-8) = FACTOR(I,MCIYR-8)+0.10 ! Increase bandwidth factor
            BANDWTH(I,MCIYR-8) = FACTOR(I,MCIYR-8)
*       *BANDWTH(I,MCIYR-8) ! Adjust bandwidth
        ENDIF
        IF (BANDWTH(I,MCIYR-8) .GT. 5.0) THEN
            WRITE(6,*) 'ERROR! Bandwidth adjustment has failed ',
*for variable 'INVAR(S),' in year ', MCIYR
            CLOSE (14)
            STOP ' '
        ENDIF

        WRITE(14,*) 'FACTOR INCREASE FOR NVAR ',TAXVARS(I)
        WRITE(14,*) FACTOR(I,MCIYR-8),'BANDWTH ',BANDWTH(I,MCIYR-8)
        WRITE(14,*)

50230   CONTINUE
50210   CONTINUE

```

C Compute Rosenblatt-Parzen kernel densities using product kernels. The C marginal densities are written to the DIAGS.PRN file.

```

        KCOUNT(MCIYR-8) = 0 ! Initialize
        KERNEL(NVARS+1,NSIMS+1,MCIYR-8) = 0.0 ! Initialize
        DO 50260 J=SNSIMS,ENSIMS
            IF (KERNEL(NVARS+1,J,MCIYR-8) .GE. 0.0) THEN
                KERNEL(NVARS+1,NSIMS+1,MCIYR-8)
*           = KERNEL(NVARS+1,NSIMS+1,MCIYR-8)
*           +KERNEL(NVARS+1,J,MCIYR-8) ! Aggregate product kernels
                KCOUNT(MCIYR-8) = KCOUNT(MCIYR-8)+1
            ENDIF
50260   CONTINUE
        IF (KCOUNT(MCIYR-8) .GT. 0) THEN
            KERNEL(NVARS+1,NSIMS+1,MCIYR-8) =
*           KERNEL(NVARS+1,NSIMS+1,MCIYR-8)
*           /DBLE(KCOUNT(MCIYR-8)) ! Compute density
        ENDIF

```

C Compute kernel weights. The computed weights and their sum is C written to the DIAGS.PRN file.

```

        DO 20279 I = 1,NSIMS+1
            WEIGHT(I,MCIYR-8) = 0.0 ! Initialize
20279   CONTINUE

        DO 50280 J=SNSIMS,ENSIMS
            IF (KERNEL(NVARS+1,J,MCIYR-8) .GE. 0.0) THEN
                WEIGHT(J,MCIYR-8) = KERNEL(NVARS+1,J,MCIYR-8)
*           /KERNEL(NVARS+1,NSIMS+1,MCIYR-8) ! Compute
            ENDIF
            WEIGHT(NSIMS+1,MCIYR-8) = WEIGHT(NSIMS+1,MCIYR-8)

```

```

* +(WEIGHT(J,MCIYR-8)/DBLE(KCOUNT(MCIYR-8))) ! Aggregate
50280 CONTINUE

```

C Compute Nadaraya-Watson Estimate. Estimates are written
C to the FORECAST.PRN file.

```

DO 50298 I = 1,OVARS
DO 50299 J = 1,NSIMS+2
ESTIMATR(I,J,MCIYR-8) = 0.0 ! Initialize
50299 CONTINUE
50298 CONTINUE

DO 50300 J=SNSIMS,ENSIMS
IF (KERNEL(NVARS+1,J,MCIYR-8) .GE. 0.0) THEN
DO 50310 I=1,OVARS
ESTIMATR(I,J,MCIYR-8) = KERNEL(NVARS+1,J,MCIYR-8)
* *OUTPUTS(I,J,MCIYR-8)
ESTIMATR(I,NSIMS+1,MCIYR-8)
* = ESTIMATR(I,NSIMS+1,MCIYR-8)
* +ESTIMATR(I,J,MCIYR-8)
50310 CONTINUE
ENDIF
50300 CONTINUE
DO 50320 I=1,OVARS
ESTIMATR(I,NSIMS+2,MCIYR-8) = (ESTIMATR(I,NSIMS+1,MCIYR-8)
* /KERNEL(NVARS+1,NSIMS+1,MCIYR-8))/DBLE(KCOUNT(MCIYR-8))
50320 CONTINUE

```

```

DO 50330 I=1,MCNMMAC+MCNMMACREG
IF (MACFDBK .NE. 0) THEN
IF (I .LE. OVARS) THEN
IF (MCDRIO(I,MCIYR-8) .NE. 0.0) THEN
EPMAC(I,MCIYR) = (ESTIMATR(I,NSIMS+2,MCIYR-8)
* /MCDRIO(I,MCIYR-8))-1.0
ELSE
EPMAC(I,MCIYR) = 0.0
ENDIF
ENDIF
ELSE
EPMAC(I,MCIYR) = 0.0
ENDIF
50330 CONTINUE
ENDIF

```

```

DO 50331 I=1,MCNMMAC+MCNMMACREG
IF (I .LE. MCNMMAC) THEN
ESMAC(I,MCIYR) = (1.0+EPMAC(I,MCIYR))*EBMAC(I,MCIYR)
ELSE
ESMACREG(11,I-MCNMMAC,MCIYR) = (1.0+EPMAC(I,MCIYR))
* *EBMAC(I,MCIYR)
ENDIF
50331 CONTINUE

```

C End of subroutine KERNELSUB. Return to calling subroutine MAC.
RETURN

END

**This Page is
Intentionally
Left Blank**

Appendix H

**Listing of
Subprogram
KERDATA**

Appendix H

Listing of Subprogram KERDATA

C*****

BLOCK DATA KERDATA

C*****

IMPLICIT NONE

INCLUDE (PARAMETR)

INCLUDE (NCNTRL)

INCLUDE (MACPARM)

INCLUDE(MCINTER2)

INTEGER I

C TAXVARS is the array of input variable labels. It is dimensioned by the
C number of input variables, NVAR.

DATA (TAXVARS(I),I=1,12) /'&TXGF','WPI051','WPI053','WPI054',
*'WPI055','WPI0561','WPI057','DTFUELSALLB','DENDUSE@COAL',
*'DENDUSE@ELC','DENDUSE@PET','DENDUSE@NG'/

C OUTVARS is the array of output variable labels. It is dimensioned by
C the number of output variables, OVAR.

DATA (OUTVARS(I),I=1,119) /'CDMV&P92C','CDFURN@CMP92C','CDCMP',
*'CDO92C','CNCS92C','CNFOOD92C','CNFUEL92C','CNGAS92C','CNOO92C',
*'CSHOUS92C','CSHHOPE92C','CSHHOPG92C','CSHHOPO92C','CSTRANS92C',
*'CSO92C','CSMED92C','IPDENRAUTO92C','IPDENRMCOF','IPDENROTHR92C',
*'ICNRB&O92C','ICNRMI&PET92C','ICNRPU92C','IPDER92C','ICR92C',
*'GFMLCFC92C','GFMLCO92C','GFMLGI92C','GFML92C','GFMLWSS@FAC92C',
*'GFOCFC92C','GFOCO92C','GFOGI92C','GFO92C','GSL92C',
*'GFWSS@FAC92C','GFONINV92CCH','GSLCFC92C','GSLCO92C','GSLGI92C',
*'GSLWSS@FAC92C','EX92CNIA0','EX92CNIA1','EX92CNIA2O','EXNIA2BM',
*'EX92CNIA2AC','EX92CNIA3','EX92CNIA4','EXD&N92C','EXS92C',
*'TYF92C','M92CNIA0','M92CNIA1@PET','M92CNIA100','M92CNIA2@BM',
*'MNIA2BM','M92CNIA3','M92CNIA4','MS92C','PAYYF92C','INV92CCH',
*'GDP92C','GDP','C92C','C','I92C','I','IFIXNR92C','IFIX92C',
*'G92C','EX92C','M92C','GNP92C','PCWGDP','GDP92CFE','PCWEX',
*'PCWM','RMGBS3NS','RMMTGCCNS','RMPUAANS','REALRMGBLUS','ECIWSP',
*'JULCNF','SQTRCARSIMP','SQTRCARS','SQDTRUCKSL','SQDTRUCKSH&M',
*'RUC','WPI','WPI14','WPI11','LC','RMFEDFUNDNS','CPI','YD92C',
*'WSD','YP92C','SHUMBL','HUSTS1','HUSTS2&','KQMH','KQHUSTS1',
*'KQHUSTS2&','N','N16&','MFGWGRT','NMFGWGRT','KAMUSE','KAUTO',
*'KDORM','KEDUC','KHEALTH','KHOTEL','KMFG','KMISCNR','KOFFICE',
*'KPUB','KREL','KSTORES','KWARE'/

END BLOCK DATA KERDATA

Appendix I

**Partial Listing of
Subroutine
MACOUTPUT**

Appendix I

Partial Listing of Subroutine MACOUTPUT

```
C*****  
SUBROUTINE MACOUTPUT  
C*****
```

..

C Kernel regression - printout results

C Begin writing forecast of output variables and sample input variables

C to FORECAST.PRN file. This file resides in the user's PWD. Any

C existing file with the same known is overwritten without warning.

```
OPEN(14,FILE='MACOUT4',IOSTAT=MCRETCOD)  
IF (MCRETCOD .NE. 0) WRITE(*,2360) MCRETCOD,'MACOUT3'  
REWIND 14
```

C Write time and date stamp to FORECAST.PRN file.

```
WRITE(14,('"'OUTPUT FROM KERNEL REGRESSION PROGRAM, KREG"''))  
CALL DATE_AND_TIME(BIG_BEN(1),BIG_BEN(2),BIG_BEN(3),DATE_TIME)  
WRITE(14,('"'Run date is: ",2(I2.2,"."),I4,  
* " and time is: ",I2.2,2(":",I2.2)," ",I3.3,""/)')  
* DATE_TIME(2),DATE_TIME(3),DATE_TIME(1),  
* DATE_TIME(5),DATE_TIME(6),DATE_TIME(7),DATE_TIME(8)
```

C Write output variable forecast to file FORECAST.PRN

```
WRITE(14,('"'OUTPUT VARIABLE FORECAST"''))  
WRITE(14,50330) "'YEAR'",(K,K=KSTYR,KSTYR+NOBS-1)  
50330 FORMAT(A6,<NOBS>(1X,"",I4,""))  
DO 50340 I=1,OVARs  
WRITE(14,50350) OUTVARS(I),(ESTIMATR(I,NSIMS+2,K),K=1,NOBS)  
50340 CONTINUE  
50350 FORMAT("",A20,"",<NOBS>(1X,E16.10))
```

C Write input variables to file FORECAST.PRN

```
WRITE(14,('/,""'SAMPLE VARIABLES"''))  
WRITE(14,50360) "'YEAR'",(K,K=KSTYR,KSTYR+NOBS-1)  
DO 50370 I=1,NVARs  
WRITE(14,50380) INVARs(I),(SAMPLE(I,K),K=1,NOBS)  
50370 CONTINUE  
50360 FORMAT(A6,<NOBS>(1X,"",I4,""))  
50380 FORMAT("",A20,"",<NOBS>(1X,E16.10))
```

C Write output variable forecast to file FORECAST.PRN

```
WRITE(14,('/,""'MCDRIO"''))  
WRITE(14,50410) "'YEAR'",(K,K=KSTYR,KSTYR+NOBS-1)  
DO 50410 I=1,OVARs
```

```
WRITE(14,50350) OUTVARS(I),(MCDRIO(I,K),K=1,NOBS)
```

50410 CONTINUE

C Write file header information identifying NEMS run.

```
WRITE(14,*)  
WRITE(14,2370) SCEN, DATE, COMMENT
```

C Write dates to file.

```
WRITE(14,2380) "'YEAR'",(I,I=1990,1990+NYRINDX-1)
```

```
DO 50390 I=1,MCNMMAC+MCNMMACREG
```

```
WRITE(14,50400) 'EPMAC',MV1(I),MV2(I),
```

```
* (EPMAC(I,J),J=1,NYRINDX)
```

50390 CONTINUE

```
50400 FORMAT ('"',A8,'"','"',A12,A12,'"','1X,<NYRINDX>(1X,E16.10))
```

C Write MCKCOMM

```
WRITE(14,*)  
WRITE(14,2380) "'YEAR'",(I,I=1990,1990+NYRINDX-1)
```

```
DO 50391 I=1,12
```

```
WRITE(14,50392) TAXVARS(I),(MCKCOMM(I,J),J=1,NYRINDX)
```

50391 CONTINUE

```
50392 FORMAT('"',A20,'"','<NYRINDX>(1X,E16.10))
```

C Finished writing to file FORECAST.PRN.

```
CLOSE(UNIT=14)
```

C End of subroutine MACOUTPUT. Return to calling subroutine MAC.

```
RETURN  
END
```