FIXER – General Vector Fixing Program for InterDyme Models

Inforum
September 2009

FIXES AND THE FIXER PROGRAM

Fixes, as used here, are ways to make a model work the way we want it to, not necessarily the way that emerges from its equations. The power that fixes give over a model can certainly be, and often has been, abused. Nonetheless, they have a legitimate role. Suppose, for example, we wish to consider the impacts of some event which the equations never dreamed of, like a natural disaster or a massive overhaul of the health care system. Then a fix is the natural way to convey to the model that the equations are not to be entirely trusted.

Interdyme has three types of fixes, those for macro variables, those for vectors and matrices, and a special type for industry outputs.

Vector and Matrix Fixes

The vector fixes are more complicated than macro fixes because they can apply to individual elements of a vector, to the sum of a group of elements, or to the sum of all elements in the vector. However, the format of the vector fixes is very similar to that of the macro variable fixes, described above. Matrix fixes at the current version (Interdyme 2.2, Fixer 1.5) are still rather simple, one fix being applicable to only one cell of a matrix. The preparation of the vector and matrix fixes is the work of the Fixer program. (Fixer is also sometimes referred to as VecFixer.)

G, it should be noted, normally prepares vectors of exogenous variables; fixes apply to vectors of endogenous variables. However, the Fixer program can also be used to supply the values of exogenous variables as well. Also, when building a model, before all the equations are finished, Fixer can be used to project the values of right hand side variables of some of the equations.

When and how are fixes applied as a model runs? Unlike the macro fixes, which are automatically applied when a macro regression equation is calculated, vector and matrix fixes are applied where the model builder specifies. At the point where the fixes for the vector x should be applied, the model builder must put into the program the line

x.fix(t);

The input to Fixer is a file prepared by the user in a text editor. It should have the extension .vfx . Fixer also reads the definitions of static groups of sectors and writes them into the GROUPS.BIN file which can be used both by the simulation program and by G. To use the Fixer program, *it is essential that the model's VAM.CFG file should have a vector called "fix"* with enough rows to allow one for each fix. As Fixer reads the fixes from the input file, it stores the numerical values of the fixes into this "fix" vector in the vam file. It also creates a "fix index" file, which will have the extension .fin and tells the simulation what to do with each fix. Finally, it

produces a binary file with the definitions of groups, called GROUPS.BIN. If G has already produced a GROUPS.BIN file, Fixer reads it and may add to it.

Fixer is started by the command Model | VexFixer on the G main menu. From the information on the form which this command creates, G prepares a configuration file, called FIXER.CFG, which is read by the Fixer program. This form specifies the root names of:

the text input file
the fix index file
the vam file used for base values for the index and growth rate fixes
the name of the output check file, which will show the sectors in each group and the values of each fix in each year. It is used only for manual checking of the program.

While it is up to the user to name files, it makes good sense to give files for the same simulation the same "root" name. A simulation that involves low defense expenditures, for example, could have a vam file called LOWDEF.VAM, and a .VFX file called LOWDEF.VFX.

For vectors, fixes may apply to a single element or to a group of elements. The concept of a "group", already touched upon under Vam, is central to the working of Fixer. Basically, a group is simply a set of integers, usually representing sectors in the model. Defining groups is useful because we often want to impose a fix on a group of elements in a vector. For example, we may want to control the total exports of the chemical manufacturing sectors. We might then create a group named "chem", which would contain the sector numbers of all the sectors in question. The command for defining a group is "grp <groupname>", where the groupname can be a number or a name. The sectors defining the group are then entered on the next line. For example,

```
grp 1
   7 10 12
```

creates a group called 1 of the sectors 7, 10, and 12. The "-" sign means consecutive inclusion. Thus

```
group zwanzig
   1 - 20
```

consists of the first twenty integers. Parentheses mean exclusion. Thus

```
group duo
   :zwanzig (2 - 19)
```

makes the group "duo" consist of the integers 1 and 20.

When a group is referenced after it is defined, its name must be preceded by a colon, as shown when "zwanzig" was used in the definition of "duo" above. Names of groups are case sensitive; commands for Fixer must be lower case. Groups do not have to be kept in numerical order and can be defined anywhere in the input file before the first time you used them. If you try to redefine an existing group, the program will complain, unless the new group has less than or equal to the number of elements in the old group. References to other groups can be used in new group definitions only if the groups referenced have already been defined.

Interdyme provides a number of ways for a fix to work. In all of them, a time series is specified by the fix. The forms of the fix differ in how they obtain and in how they apply this time series. The basic format of the input file for a vector fix is:

<command> <vectorname> <GroupOrSector>

followed on the next line by the year and value of the fix. The basic format of the input file for a matrix fix is:

<command> <matrixname> <row> <col>

Definitions of the 6 legal commands and examples follow.

ovr

overrides the result of the equations with the value of the time series given. Again, intermediate values are linearly interpolated. In the example below, the fix program would calculate and override fix series that starts in 1992, ends in 2000, and moves in a straight line between the two points. For example,

```
ovr ex 10
  92  154.1
  2000  182.3;
```

would override the value of the forecast of element 10 of the "ex" vector (probably exports) with the values shown for the years shown. Note that year can be either 2-digits or 4-digits (they are all converted to 4-digits in the program). As an example of a matrix fix,

```
ovr am 1 9
       1990 .23
       1995 .26
       2000 .28;
```

would override the value of the A-matrix in the Vam file for element (1,9), from 1990 to 2000. As before, missing values are linearly interpolated.

mul

multiplies the equation forecast by a factor specified by the data series on the following line. For example,

```
mul im 44
  1992  1.0
  1995  1.05
  2000  1.10;
```

multiplies the forecast results for imports of sector 44 by the factors shown. Values of the multiplicative fix on imports between the years shown are linearly interpolated.

cta

does a constant term adjustment. That is, it adds or subtracts the value of the time series to the result of the equations. The time series is provided by the fix definition. For example,

```
cta def :Alice
   1992 .0001
   1995  200
   2000  180;
```

is a constant term adjustment for defense expenditures of all sectors in the Alice group. Intermediate values are linearly interpolated.

ind, dind

is a variety of the override fix that specifies the time series as an index. There must be data in the vam file for the item being fixed up until at least the first year of the index series specified. The value for the item in that year is then moved by the index of the time series given by the fix lines. For example,

```
ind pceio :zwanzig
  1982  1.0  1.03  1.08  1.12  1.15
```

```
    1997  1.21  1.29  1.31  1.34;
```

will calculate the sum of the elements of the pceio vector included in the group "zwanzig" in 1982, will move that sum forward by the index of the series given, and will impose that control total on the those elements when the model is run.

The “dind” version of the fix can start in any year, and indexes the series to the value of the expression in the starting year of the fix.

gro, dgro

is a type of override fix that specifies the time series by growth rates. For the growth rate fix to be legal, there must be data in the vam file up until at least the year before the first year of the growth rate fix. Missing values of the growth rates are linearly interpolated.

```
gro out 10
  1983  3.1
  2000  3.4;
```

The “dgro” version of the growth rate fix can start in any year, and always calculates the series in the present period based on the value in the previous period.

stp, dstp

is a step-growth fix. It is like “gro” except that a growth rate continues until a new one is provided. A value for the final period is necessary.

```
stp out 1
  83  4.1
  95  4.5
  2000 5.0;
```

The “dstp” version is the dynamic version, which can start in any year. It is just like “dgro”, except for the method of interpolation of the fix values.

eqn

The equation fix for vectors works in the same way as the version for macrovariables, with the exception that the name of the vector must be separated from the sector number by a space. For example:

```
# Make the pce deflator for category 3 grow like the aggregate PCE deflator,
#   based on the ratio in 1997, from 1998 to 2010.
eqn cprices 3 = cprices3{1997}/apc{1997} * cprices3
    1998 1
    2010 1
# Make corporate profits in sector 1 remain a constant share of total
corporate
#   profits, equal to the share in 1997:
eqn cpr 1 = cpr1{1997}/vcpr{1997} * vcpr
   1998 1
   2010 1
```

fol

The follow fix specifies that an element or group of a certain vector should follow the expression on the right, plus or minus a certain growth rate, which can be specified in the body

of the fix. It is often used to make imports of a certain commodity grow like domestic demand. For example, the following follow fix makes crude petroleum imports grow like domestic demand, plus 0.2 percent per year:

```
fol im 4 = dd4
    1998 0.2
    2030 0.2
```

shr

The share fix takes the value of the body of the fix (fixval), and multiplies the right hand expression by it, before assigning the value to the left hand side variable or group. Like the follow fix, a typical use for this fix might be in controlling the relation between imports and domestic demand. The example below specifies the share of domestic demand for imports of Radio, television and video equipment:

```
shr im 42 = dd42
    1998 .9
    2000 .92
    2030 1.0
```

When the input file as described above is ready and the FIXER.CFG file calls for its use, type "fixer" at the DOS prompt to invoke the Fixer program.

Please visit the Software pages of the Inforum web site for more details and to download the Fixer software: www.inforum.umd.edu/software/software.html.