Building JIDEA6: Result and its Problems

Problems occurred in model building

Mitsuhito Ono

Problems and their solutions

- **Rhoadj program** in relation to constant term adjustment
- How to set dyme.cfg for test, simulation and forecasting
- To keep the Balance of the control total
- To stabilize the model
- How to set vecfix.vfx

6. Input figures

7 . Compare program

1. **Rhoadj program** in relation to constant term adjustment

1. Problems occurred at rhoadj function using constant term adjustment program because pc did not recognize rhoval which was defined equation.h as 2, thus

   (fabs(rhoval)>1?rhos[i]:rhoval) did not work.

   This leads to diverge the model because errors are added to the value.

2. Solution:

   1. **by setting rhoval=2 in each function such as expfunc**
   2. **inserting rhoval to rhoadj program;**

      expr_rho = expreg.rhoadj(dep,expract,i,rhoval); <- (dep,expract,i);
   3. **Inserting printf("\n %f", (fabs(rhoval)>1)?rhos[i]:rhoval);**

      Then the model started recognizing the command and calculate as we expected.
   4. **Strange enough once the model works correctly, the original program starts working.**

```
/* rhoadj() -- perform rho adjustment on predicted values from equation.*/
float Equation::rhoadj(float value, float xi, short i,float rhoval)
{
        /* value is what was computed by the equation
            xi is what was already in the vector
            i is the element number being worked on.
            the return value is the rho-adjusted forecast.
            */
    float ret;

        if (useall == 'y' && t < rhostart)
                return(xi);
        if((useall == 'y' && t == rhostart) || useall == 'n' && t == godate){
                if(setrho == 'y')
                   errors[i] = (xi - value)*(fabs(rhoval)>1?rhos[i]:rhoval);
                return(xi);
                }
        else{   // Make a rho adjustment and update the error term.
                if(setrho == 'y') {
         /* When setrho == 'y', value should not change from xi */
                        errors[i] = (fabs(rhoval)>1?rhos[i]:rhoval)*errors[i];
         // try to get rid of floating point underflow..
         if(fabs(errors[i])<1.0e-8)
             errors[i]=0.0;
         ret = xi;
                        }
       else   {
            /* The return value should not be the "actual", only in the
                case where setrho is not equal to 'y'.
            */
                ret = value + errors[i];    // default
            }
                return(ret);
                }
        }
```

2. How to set dyme.cfg for test, simulation and forecasting

1. Forecasting case: start, discrepancy year=2005, useall=xxx, lastdata=2005, If we set useall as ‘yes’, ‘y’, ‘no’ , then the outcome is as follows.

JAPAN INTERINDUSTRY DYNAMIC ECONOMETRIC ANALYSIS (JIDEA)

Version 6.0

ITERATION COUNTS

Yr M Q P	GDPR	Cons	Inv	Exp	Imp	Emp	Wage/E	Infl	OUTR	GrGDPr
5 2	522.5	387.6	130.4	73.2	68.6	63560	4.034	0.10	953.1	2.26
6 1604 3	533.2	388.0	133.1	78.8	66.6	62769	3.984	-6.28	982.1	2.05
7 64 4 4	541.7	397.4	133.4	82.7	71.8	64098	4.100	4.33	996.5	1.59
8 11 4 6	543.9	399.2	134.2	84.9	74.4	64287	4.161	2.85	1002.3	0.40
9 65 9 7	548.9	398.8	140.2	87.7	77.8	64969	4.267	3.29	1014.2	0.93
10 52 9 7	550.7	401.5	137.7	91.9	80.4	65318	4.358	2.42	1019.8	0.32

Total execution time: 52 seconds

If we set useall as ‘No’ or ‘N’, then the outcome is as follow.

JAPAN INTERINDUSTRY DYNAMIC ECONOMETRIC ANALYSIS (JIDEA)
Version 6.0

ITERATION COUNTS

Yr	M	Q	P	GDPR	Cons	Inv	Exp	Imp	Emp	Wage/E	Infl	OUTR	GrGDPr
5	4			503.3	372.5	120.4	74.5	64.1	63988	4.079	-0.06	953.1	-1.51
6	99	4	2	560.5	409.9	135.1	83.5	68.0	66239	4.161	8.61	1072.8	0.90
8	55	9	3	591.9	435.1	146.8	90.5	80.5	72110	4.329	3.65	1122.4	4.66
9	62	4	8	600.7	442.9	148.5	94.0	84.8	74076	4.427	4.31	1139.7	1.48
10	1599			610.4	441.5	160.7	98.1	89.8	75584	4.554	4.83	1162.7	1.63

2. Historical test or simulation case: start, discrepancy year=2000, useall=xxx, lastdata=2000

If we set useall as 'no' 'n' , then the outcome is as follows.

JAPAN INTERINDUSTRY DYNAMIC ECONOMETRIC ANALYSIS (JIDEA)
Version 6.0
ITERATION COUNTS

Yr	M	Q	P	GDPR	Cons	Inv	Exp	Imp	Emp	Wage/E	Infl	OUTR	GrGDPr
0	2			519.0	385.9	129.8	57.5	54.2	64460	4.137	0.15	947.9	0.93
1	57	2	2	558.2	430.1	128.8	60.0	60.7	63742	4.389	-1.50	996.7	7.53
2	1212	1		523.3	425.0	111.4	58.5	71.7	62250	4.445	1.83	949.1	-6.25

If we set useall as 'No' or 'N', then the outcome is as follows.

JAPAN INTERINDUSTRY DYNAMIC ECONOMETRIC ANALYSIS (JIDEA)
Version 6.0
ITERATION COUNTS

Yr	M	Q	P	GDPR	Cons	Inv	Exp	Imp	Emp	Wage/E	Infl	OUTR	GrGDPr
0	6			506.9	374.0	133.2	57.1	57.4	63671	4.256	0.15	947.9	-1.44
1	57	2	2	550.8	419.4	133.3	59.0	60.8	63406	4.509	-0.16	1008.1	8.67
2	36	2	2	535.1	416.1	124.4	59.2	64.6	63717	4.542	1.67	978.8	-2.86
3	47	2	2	535.4	409.5	132.1	63.1	69.5	65744	4.381	1.37	1003.3	0.06
4	45	2	2	540.1	413.4	129.7	68.5	71.5	66037	4.358	-2.48	1013.5	0.89
5	25	2	1	563.3	426.5	132.5	78.3	74.0	67854	4.358	-2.02	1054.4	4.29
6	2	7	10	434.3	292.6	128.6	84.5	71.6	60997	4.389	-112.34	840.5	-22.91

3. To Keep the Balance of the control total

To keep the balance of control total, we modified cohrfunc program in fdfunc.cpp.

Our model calculation process

start -> **data input** -> cobfunc, cogfunc, cohfunc, expfunc, adjfunc, invfunc -> iprr -> summuation ->

hwfunc, prdhfunc, wagfunc, othfunc, profunc, taxfunc, depfunc, subfunc, ->

summuation ->

PSeidel-> pexfunc -> accountant -> fdr(fd) -> print

If no

conversion check: **if yes** -> setrho='y' -> cogfunc, cobfunc, invfunc ,expfunc, adjfunc, prdhfunc ->

hwfunc, wagfunc, othfunc, profunc, taxfunc, depfunc, subfunc, pexfunc ->

cohfunc -> setrho='n'

then next year

```
/* cohfunc() -- the Household Consumption function */

     ********** omission ********

     if (which == 'a'){
                     cohrpopcalc = cohreg[i][1] + cohreg[i][2]*dum85y[t]
                           + cohreg[i][3]*dum90y[t] + cohreg[i][4]*dum95y[t]
                           + cohreg[i][5]*ctaxdumy[t] + cohreg[i][6]*timet;
                     }
               else if (which == 'b')

     ********** omission ********

      cohrpop_rho = cohreg.rhoadj(cohrpopcalc,cohrpopact,i);
      cohrpop[k] = cohrpop_rho;
      cohr[k] = cohrpop[k] * pop[t];
      }
     totcohr[t] = cohr.sum();
     consmv[t] = cohr[46]; // consumption of motor vehicles for savrat.reg
     if( setrho = 'n') {
     // yields cohcontrol[t]
       savratf();   // Savings rate equation:
       depend = savrat[t];
       savrat.modify(depend);
       if(t>cohr.LastDat) {
              // Scale all consumption to control total.
              for(i=1;i<=NJIDEA;i++)
                     v[i] = cohr[i];
              n = NJIDEA;
              err = rdscale(v,n,group,cohcontrol[t]);
              for(i=1;i<=NJIDEA;i++)
                     cohr[i] =  v[i];
              }
       cohr.fix(t);
       totcohr[t] = cohr.sum();

       float ctrl = cohcontrol[t];
       float cohrsum = cohr.sum(); // compare for debugging.

       err=err; // prevent warning
       }
       return(n);
       }
```

4. To stabilize the model

As Japan depends almost all of her oil demand on imports, the Japanese I-O data does not have large domestic output in the sector 5; Petro & Gas exploration. In case of 2005, the real import of the sector was 6950.7 billion yen against th at of output 111.2.

When calculating price deflator of domestic demand of sector 5; pdd5, the figure fluctuates a lot and leads the model unstable as the consumption of this sector is very small.
(pdd = ddtot/ddtotr = {out + imp - (exp+adj)}/{outr + impr - (expr+adjr)}

So we imposed an assumption: pdd[5] = pim[5], which contributed much to the model stability.

5. How to fix vecfix.vfx

In case of model running, we need to fix some figures of variables. However, it is obscure how the vecfix program copes with our instructions.

In case that we put ovr, cta and ind individually, we can find the expecting figures are correctly inputted by fixer.chk.
However, when using these commands with group fix, we can not find what figures are used in the program.

For example,

Case 1
cta chor 64
2006 1000,
2007 500;

cta cohr: All
2006 -9000.
2007 -8000.;

Result

Yr M Q P	GDPR	Cons
5 2	522.5	387.6
6 33 4 2	536.1	395.0
7 24 4 2	543.4	401.4
8 9 4 5	553.6	408.6
9 15 9 7	564.1	416.7
10 17 9 6	570.4	421.5
11 35 9 6	578.0	426.2
12 12 4 6	580.2	431.6

Case 2
cta cohr: All
2006 -9000.
2007 -8000.;

cta chor 64
2006 1000,
2007 500;

Result

Yr M Q P	GDPR	Cons
5 2	522.5	387.6
6 33 4 2	535.8	395.1
7 24 4 2	543.2	401.5
8 9 4 5	553.4	408.8
9 15 9 7	564.1	416.9
10 17 9 6	570.5	421.8
11 2004 6	595.3	424.0
12 13 9 6	590.5	440.3

6 . Input figures

- When I try to register a figure 1.111111111110 in g/vam bank, the input figure recorded as 1.1111111641 in a bank.
- I think this is because default value of G7 is 7 digit below decimal point.
- I want make the effectiveness of digits much more like 10 digit below decimal point.
- As the number of our model sectors is 100, when aggregated the sector's figures from 1 to 100, there happens a discrepancy in the fourth or fifth digit below decimal point. That corresponds to 100 million yen order and I want to avoid any error in this digit.

- The trouble became outstanding when we calculate @csum(cohr,1-100) .
- The result was 296.684906. When we added from cohr1 to cohr100 by excel, result was 296.684899.
- The difference was 7 million yen.

- When we applied this to output, @csum(outr,1-100)=1041211.5625 and excel result was 1041211.66187525. (unit: billion yen)
- The difference became 99 million yen.

7. Compare program

- When using compare program, some discrepancies occurs.

  In compare program, ∑cohr =296.685**4148778**,

  however, in G, @ cum（cohr, 1-100）=296.685**394287109**.

  Why does this happen? How can we rectify the differences?