EXPORTS AND IMPORTS IN ALTERNATIVE CONSTANT DOLLARS

FARHAT MAHMOOD

The objective of this project is to derive NIPA data in alternative constant dollar values. Exports and Imports are considered here. We want to utilize as much detail as possible in the data, and the Unpublished detail upon which the NIPA tables are based is a viable choice.

The BEA provided annual worksheets with the latest Unpublished data on Merchandise Exports and Imports. The files that I looked at were umd_94cu.wk3 and umd_95co.wk3 which contain the currant and constant dollar data.

The NIPA tables are published in billions of dollars. Merchandise exports have 11 end use categories, and merchandise imports have 12 end-us categories. The published data is given for the years 1965-present. The Unpublished data, on the other hand is more useful for this project because it is available in millions of dollars and has greater detail. Exports have about 95 detailed categories, while Imports have about 112 detailed categories. All the Unpublished current dollar data is given for the years 1978 to 1984 . The Unpublished constant dollar series have data between the years of 1985 and 1994. Some of the constant dollar series are extended back to the year 1981 while others even extend back to the year 1978. However, in the constant dollar series, all the aggregate values for 5 broad groups are given for the years 1978 - 1994.

There are however, some problems with the Unpublished data that need to be addressed before we can proceed.

- The unpublished data is slightly different than the published data.

- The unpublished data was not consistent with itself. The summations of all the individual values within a group are slightly different than the aggregate totals that are given to us. Also, the 87$ constant values do not match the current values in the year 1987.

- The constant dollar data is missing for some detailed categories for some of the early years.

The first two problems are small and we can speculate that they are caused by rounding error when changing from quarterly data to annual data. The third problem is what we will deal with. In order to correct these problem we need to do two things:

1)To fill in the missing values, we will derive deflators for the detailed categories from the movement of the published categories (aggregate values). Then, we will derive constant dollar detail by using the deflators and the current dollar detailed series.

2)To keep all the totals consistent with the aggregate values we are given, we will need to scale the value of the detailed components of the aggregate categories.

Filling in missing data - Using Group deflators

I begin with the annual data for exports and imports from the BEA (Bureau of Economic Analysis), for the spring 1995. The data in the files "umd_94cu.wk3" and "umd_95co.wk3" (one for current dollars and the other for constant dollars) covers the years 1978-1994. I have turned these files into "prn" files and then into data cards so they can be read by G. The first table (which follows the text) shows the original data as given by the BEA. The data card file is called CUCOXM.DAT (the currant and constant values have been combined into this one file) which is made into a workspace bank called CUCOXM.BNK. The table was made using the stub file CUCOXM.STB.

The export\import data is divided into five groups:

1) FOOD,FEEDS AND BEVERAGES
2) INDUSTRIAL SUPPLIES AND MATERIALS
3) CAPITAL GOODS
4) AUTOMATIVES
5)CONSUMER GOODS

The totals of these groups are available for the entire time series. However, the values that are missing are the detailed values within these groups, for the early years.

For the purpose of clarity, from here on I will refer to the group totals as aggregate values and the component of these groups as detailed values. For the detailed values the following are missing in the original Unpublished data :

	Missing
Foods,feeds,and beverages	1978 -1984
Industrial supplies and material ..	1978 -1984
Capital goods......................	1978 -1981
Automotives........................	1978 -1981
Consumer Goods.....................	1978 -1981

Our first objective is to try to fill the missing values.Our second objective is to deflate the export and import data using various base years.

The data that I have used was provided by Dave Wasshausen at the BEA, and for anyone who is further interested in this data, his telephone number is (202)-606-9752.

General Outline of Steps to be taken to fill in the values.

1) Calculate all deflators(aggregate and detail).

2) One step that we must do before recalculating deflators is to combine all the computer series into one series. The BEA had Computers divided into five categories: Computer Proc (Computer Processors), Computer PC (Personal Computers), Computer Dasd, Computer Prin (Printers), and Computer Disp (Monitors). Some of the computer values are missing for the years 1978-1981 both in the constant and currant dollar data. The missing values give us zero values for the deflators for the years 1978-1981. By adding all the computer variables up we get rid of the zeros and get a non-zero deflator.

3) Use the deflators of the aggregates to calculate the movement for deflators of the missing detail values, with the ls command. The ls command or the link series command takes the movement of one series and makes another specific series move in the same way.(The movement can be backward or forward.)

4) Use the detailed deflator values (only for the missing years) to calculate the missing detail constant values.

5) Sum all the detailed values and compare the sums with the known aggregate values. The method which was used to construct the constant dollar detail does not guarantee that the sum of the detail is equal to the known aggregate value. The ratio of the aggregate to the sum of the detail will be used as a scaling factor to insure consistency. The scaling factors that we got were between 0.973 - 1.272 percent. A table at the end of the text will show the scaling factors. After everything has been scaled the summation of the scaled detail values will equal the aggregate values.

6) Now replace the unscaled detail values by the scaled ones.

7) Lastly, recompute the detailed deflators.

Exports and Imports Deflated with Various Base Years

Once we have the data filled in and scaled, it is ready for the second stage which is deflating the data using different base years, deriving aggregates as the sum of the detail, and analyzing the results.

Steps to deflate and aggregate:

1) Rebase the deflators - Divide the deflator for each series by the deflator of the base year.

For example: if you want the data to be in 1978 dollars then deflators for all years must be divided by the deflator for the year 1978.

dfl = dfl/dfl[78]

2) Recalculate the constant values, in terms of the new base year, by dividing the current values by the new deflator for the detailed series.

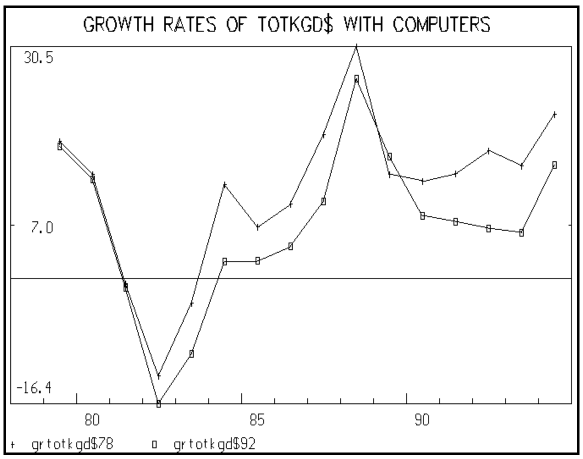
3) Form constant value aggregates, for the groups as well as the overall total, as the sum of the detailed values.

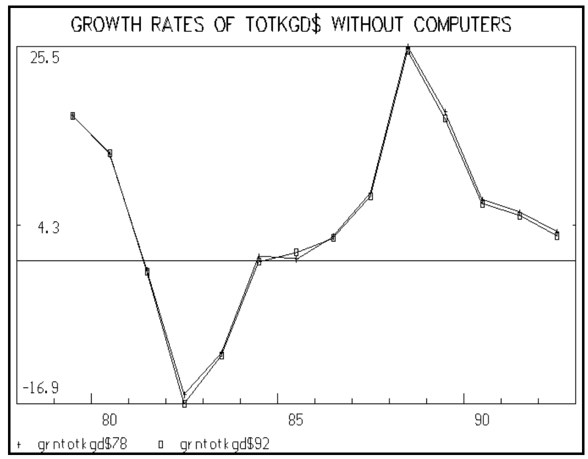
Results

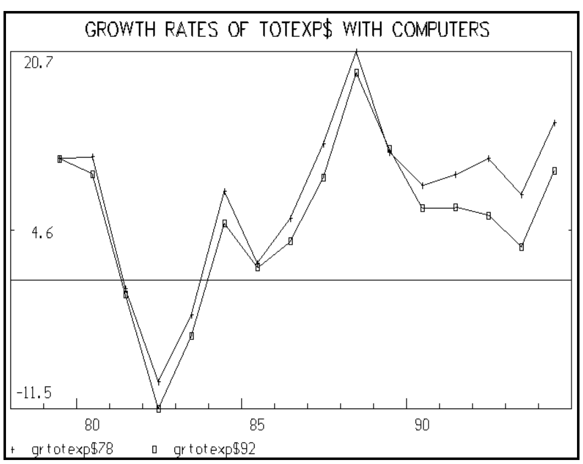
With the above method we have deflated exports and imports in 1978, 1982, 1887, and 1992 dollars.The tables that follow show growth rates of the values we have deflated.

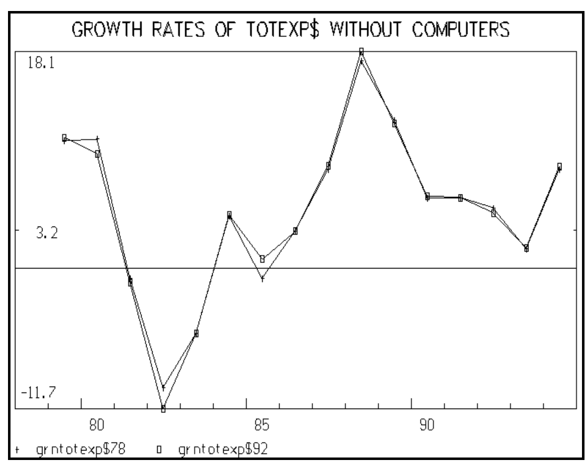
When we compare any aggregate value of total exports or total imports in 78$ and in 92$ we see a large difference in the growth rates. How much of the difference is caused by computer prices? Computers are being used more and more with time. However, the prices of computers have gone down drastically. To test the importance of the computer deflator I looked at the growth rates of capital and of total exports and imports with and without computers.

The results shown in the following graph show that the disparity between 1978$ data and 1992$ data is very small when we take out computers.









MERCHANDISE EXPORTS DEFLATED IN ALTERNATIVE BASE YEARS

	1978	1982	1986	1990	1994	78-82	82-86	86-90	90-94	78-94
TOTAL EXPORTS 78$	145.4	161.9	181.8	298.8	447.6	2.68	2.91	12.42	10.10	7.03
TOTAL EXPORTS 82$	196.1	215.2	242.0	396.2	587.4	2.33	2.94	12.32	9.84	6.86
TOTAL EXPORTS 87$	203.0	217.7	234.3	368.9	496.9	1.75	1.83	11.35	7.45	5.59
TOTAL EXPORTS 92$	232.0	246.2	257.6	398.4	509.1	1.49	1.13	10.90	6.12	4.91

MERCHANDISE IMPORTS DEFLATED IN ALTERNATIVE BASE YEARS

	1978	1982	1986	1990	1994	78-82	82-86	86-90	90-94	78-94
TOTAL IMPORTS 78$	177.4	176.1	295.6	379.0	650.6	-0.19	12.96	6.21	13.51	8.12
TOTAL IMPORTS 82$	267.0	250.6	404.6	505.9	817.6	-1.58	11.97	5.59	12.00	7.00
TOTAL IMPORTS 87$	245.2	243.0	398.0	461.4	658.5	-0.22	12.33	3.70	8.89	6.17
TOTAL IMPORTS 92$	273.4	274.2	447.1	505.9	679.0	0.08	12.22	3.09	7.36	5.69

MERCHANDISE EXPORTS WITHOUT COMPUTERS DEFLATED IN ALTERNATIVE BASE YEARS

	1978	1982	1986	1990	1994	78-82	82-86	86-90	90-94	78-94
TOTAL EXPORTS 78$	141.1	154.3	155.7	234.7	286.6	2.24	0.22	10.26	5.00	4.43
TOTAL EXPORTS 82$	190.8	205.9	209.6	316.8	388.0	1.90	0.45	10.32	5.07	4.44
TOTAL EXPORTS 87$	200.8	213.8	220.6	335.2	412.4	1.57	0.78	10.46	5.18	4.50
TOTAL EXPORTS 92$	230.8	244.1	250.3	380.5	464.0	1.41	0.63	10.47	4.96	4.37

MERCHANDISE IMPORTS WITHOUT COMPUTERS DEFLATED IN ALTERNATIVE BASE YEARS

	1978	1982	1986	1990	1994	78-82	82-86	86-90	90-94	78-94
TOTAL IMPORTS 78$	176.4	173.5	275.8	312.8	400.1	-0.42	11.59	3.14	6.15	5.12
TOTAL IMPORTS 82$	265.9	247.9	383.6	435.8	552.5	-1.75	10.92	3.19	5.93	4.57
TOTAL IMPORTS 87$	244.8	241.9	389.1	431.6	545.8	-0.30	11.88	2.59	5.87	5.01
TOTAL IMPORTS 92$	273.1	273.6	442.4	490.3	619.9	0.05	12.01	2.57	5.86	5.12

The Final Output

The final output is found in the data bank EXIM.BNK, and the table showing the final manipulation is created with EXIM.BNK and CUCOXM.STB and is called EXIM.OUT. These files are found in the h:\gbanks\nipa\nipaua\exim directory. Table EXIM.OUT is provided at the end of this document.

The scalers are also shown in a table called SCALER.OUT, which was created with EXIM.BNK and SCALER.STB. This table is also in the end of the document.

The various deflated banks are new178.bnk, new182.bnk, new187.bnk, new192.bnk. the corresponding output files are : new178.out, new182.out, new187.out, new192.out. Tables showing growth rates are made with the gcoxm.stb stub file.

Note

After completing the project we made a table called xmdefl.out which contained all the deflators. Here we noticed that the deflator values for the year 1987 (the base year) were considerably different from one. Therefore, we went back and did another scaling in which we made the deflators for the year 1987 equal to one. Due to that rescaling the actual figures in the banks are a little different then the values I have presented in the above tables. However, the trends are the same. Deflators are in the file XMDEFL.OUT and were made with XMDEFL.STB and EXIM.STB.

APPENDIX: USING G TO DERIVE MISSING CONSTANT DATA SERIES

The calculations described in the previous page were done with G. Here are the steps:

1) Calculating all deflators:

```
fadd dfl.add arg10.run

add file :          Dfl.add     --->   f  d%1 = %1/%1$
argument file :     Arg10.run   --->   list of all values[detailed and aggregate]
```

2) Fill in the missing deflator values:

```
fadd Dls.add arg14.run

add file :      Dls.add    --->  ls %1 %2  %3  b
argument file : Arg14.run  --->  has the detail values,
                                 the corresponding aggregate values and  the year 85 or 81.
```

3) Just for the years where the constant values are missing, calculate them by dividing the constant value by the deflator.

```
add file :      co.add          --->    fdate 78 %2
                                        f %1$ = %1/d%1
                                        fdate  off
argument file : arg15.run       --->    has the detail values and the
                                        end dates for the missing values
                                        [84or80]

fadd co.add arg15.run
```

4) Sum all the detail values for each aggregate category. I have created argument files for each aggregate group :

```
add file:       sumall.add      --->    f sum = 0
                                        fadd  sum.add %1.run
                                        f sum%1$ = sum
                [sum.add        --->    f  sum =  sum + %1$]

argument file:  arg18.run       --->    has all the aggregate
                                        value names
fadd sumall.add arg18.run
```

5) Calculate scalers for each aggregate category :

```
add file:        sclall.add       --->    f scl%1 =%1$/sum%1$
argument file:   arg18.run        --->    has all the aggregate value names
```

6) Use the scalers to scale the detail values :

```
add file:        sclval.add       --->    f scl = scl%1
                                          fadd scl.add  %1$.run
                 [scl.add         --->    f s%1$ = %1$ * scl]
argument file:   arg18.run        --->    has all the aggregate value names
```

7) Replace each original detail value with the scaled value.

```
add file:        repall.add       --->    fadd replace.add  %1$.run
                 [replace.add     --->    f %1$ = s%1$]
argument file:   Arg18.run        --->    has all the aggregate value names
```

8) Recompute deflators.

```
add file:        Dfl.add --->    f d%1 = %1/%1$

argument file:   Arg10.run
```

Now to deflate the various series in different base years.

E.g. in 1978 dollars.

```
1)      Copy new1.* new178.*

2)      >pdg
        :wsba new178

3)      fadd rundfl.add arg10.run

        Rundfl.add      ---->   fdate 78 94
                                f d%1 = d%1/d%1{78}
                                f %1$ = %1/d%1
                                fdate off

4)      fadd sumall.add arg19.run

        Sumall.add      ---->   f sum = 0
                                fadd sum.add %1.run
                                f %1$ = sum [slightly changed]

        Arg19.run       ---->   has all aggregate values, including totexp$.
```

Rescaling to adjust for the deflator problem

1) Recalculate the constant values so that dfl{87} is equal to one.

```
fadd r.add arg12.run
```

```
r.add           ---->   f r%1 = %1{87}/%1${87}
                        fdate 78 94
                        f %1$ = r%1*(%1$)
                        fdate off

arg12.run       ---->   all detailed values
```

2) Rescale the values so that the sums add up.

Re-do steps 4-8 above.