Re item 2. Editor’s report regarding of 2019-20[1]

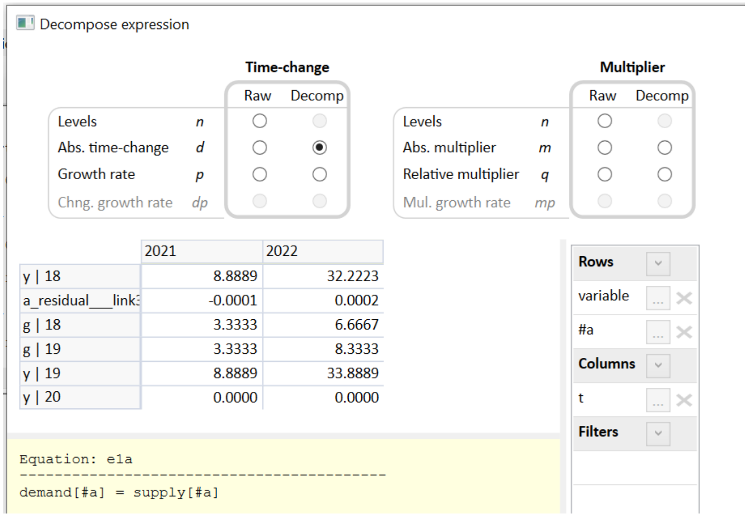
Since the last steering committee meeting, some work has been done on solver enhancements in Gekko 2.5.x (henceforward called 2.4), but the bulk of the work has been on Gekko 3.1.x (henceforward called 3.0). See the versions overview here.

Gekko 2.4 has been augmented regarding model blocks (switching parts of models on and off, and combining models), which as a side effect makes it easier to handle goals-means problems with more means than goals (interdependence between the goals).

Regarding Gekko 3.0 (current version is 3.1.9), a lot of the work has been on consolidation, fixing bugs, polishing, improving documentation, etc. The intention is to keep the 3.0 versions very stable regarding syntax and functionality, and also strictly backwards compatible relative to the original 3.0 version. See the new 3.0 (3.1.9) features here.

Gekko 3.0 developments

Much work has been done on generalized DECOMP, making it possible to decompose GAMS-like equations with dimensions, linked equations, pooling, selecting etc., in essence treating the decomposition results as a kind of pivot table. The new DECOMP looks something like this:



Decompose expression

Time-change		Raw	Decomp
Levels	*n*	○	○
Abs. time-change	*d*	○	●
Growth rate	*p*	○	○
Chng. growth rate	*dp*	○	○

Multiplier		Raw	Decomp
Levels	*n*	○	○
Abs. multiplier	*m*	○	○
Relative multiplier	*q*	○	○
Mul. growth rate	*mp*	○	○

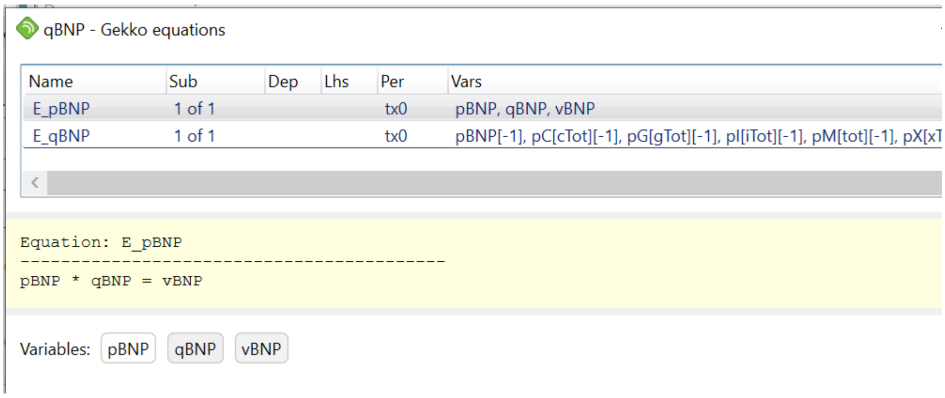
	2021	2022
y | 18	8.8889	32.2223
a_residual___link3	-0.0001	0.0002
g | 18	3.3333	6.6667
g | 19	3.3333	8.3333
y | 19	8.8889	33.8889
y | 20	0.0000	0.0000

Rows: variable, #a
Columns: t
Filters

```
Equation: e1a
-------------------------------------------
demand[#a] = supply[#a]
```

In GAMS-like models, dependent variables are not always isolated on the left-hand side of a “designating” equation, so there is also a new “FIND” window that shows all equations containing a particular variable:

[1] Links: Gekko main webpage: www.t-t.dk/gekko, organization: www.t-t.dk/gekko/organization. Gekko on GitHub: https://github.com/thomsen67/GekkoTimeseries.



Name	Sub	Dep	Lhs	Per	Vars
E_pBNP	1 of 1			tx0	pBNP, qBNP, vBNP
E_qBNP	1 of 1			tx0	pBNP[-1], pC[cTot][-1], pG[gTot][-1], pI[iTot][-1], pM[tot][-1], pX[xT

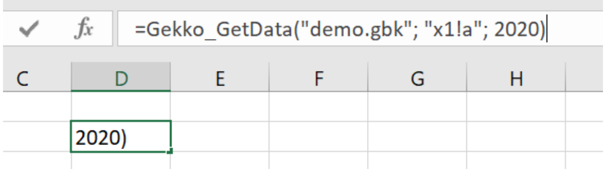
```
Equation: E_pBNP
-------------------------------------------
pBNP * qBNP = vBNP
```

The new DECOMP system is still work in progress, and the two above-mentioned windows are intended to complement each other. Two videos illustrating some of the features can be seen here (pivot) or here (decomposition) -- both videos are in Danish.

Daily frequency is implemented regarding some of the more central Gekko commands, and a blueprint regarding higher frequencies in Gekko 3.0 has been written (cf. here, comments are most welcome!). Date format handling has been looked into, especially in relation to Excel workbooks. There are many technical details regarding higher frequencies, so the implementation of these will probably be a gradual process.

Regarding Excel, it is now possible to read an Excel sheet as a nested list of cells (of different types), and operate on these cells in a matrix/table-like fashion. Gekko is increasingly being used for “data wrangling” purposes.

Regarding Excel, a so-called ‘Gekcel’ add-in has been developed. The idea is to make it possible to interact with Gekko databanks from inside of Excel, without opening Gekko proper. This can be done via special functions like the function shown here:



```
=Gekko_GetData("demo.gbk"; "x1!a"; 2020)
```

```
2020)
```

This functionality is provided in the form of an Excel xll-file (add-in), acting as a bridge between Excel and Gekko. In this way, it will be possible to read/write individual observations, or operate on larger groups of cells (over variables and periods). Ribbon buttons, Gekko operators and other things will be supported, too, and it will also be possible to issue Gekko commands from within Excel, if needed.

A new Python interface has been developed, similar to the R interface. Both interfaces have been simplified, and it is now also possible to interchange data via the new Apache Arrow data format. EXPORT<arrow> is implemented, and the following illustrates a dataframe in R or Python, where x is a normal series, and x1 and x2 are array-timeseries (all series are exported from Gekko):

```
name  freq  dims  dim1  dim2  per1  value
   x     a     0  None  None  2021    1.0
   x     a     0  None  None  2022    2.0
   x     a     0  None  None  2023    3.0
  x1     a     1     i  None  2021    2.0
  x1     a     1     i  None  2022    3.0
  x1     a     1     i  None  2023    4.0
  x1     a     1     j  None  2021    3.0
  x1     a     1     j  None  2022    4.0
  x1     a     1     j  None  2023    5.0
  x2     a     2     x     y  2021    4.0
  x2     a     2     x     y  2022    5.0
  x2     a     2     x     y  2023    6.0
  x2     a     2     x     z  2021    5.0
  x2     a     2     x     z  2022    6.0
  x2     a     2     x     z  2023    7.0
```

Read more on R and Python interfaces for Arrow on the last parts of the pages here and here. At the moment, EXPORT<arrow> produces an Arrow file containing all series and array-series of all frequencies, but perhaps Arrow files could be compressed and optionally stored inside .gbk databanks (a .gbk file is already a zip file).

Other new developments in Gekko 3.0 have been prompting and default argument in user-defined functions and procedures, quite a lot of OLS improvements, and some model file enhancements (among other things making it easier to use model equations for data generation purposes).

The Gekko user forum has been taken down due to spam and lack of activity, and we are probably going to move questions to Stack Overflow instead.

There has been some interest from non-Danish users, but the lack of a comprehensive examples collection is probably somewhat of a hindrance. The Quarterly National Accounts (Statistics Denmark) are running their AREMOS system in parallel with Gekko in the coming months and intend to run their revisions Gekko-only starting from February 2021.

From Gekko version 3.1.9 and on, Gekko uses .NET version 4.6.1 (which is five years old). This is not expected to cause problems for Gekko users. The bump in .NET version is necessary regarding Arrow files and dataframes.

Forwards

In general, Gekko versions 2.2, 2.4, and 2.5.x are not under active development. Since one of the formal responsibilities of the Gekko editor is to try to avoid too much “branching” and try to move the project in one direction, the development is focused on the Gekko 3.0 version.

There is a rather large project starting up now regarding documentation of the 3.0 source code etc. The project is done in collaboration with the IT department of Statistics Denmark, and hopefully, the project will also clear up some of the pending decisions regarding migration to so-called .NET Core, migration to 64-bit, and migration of the main Gekko window to so-called WPF (vector-based graphics).

Apart from this, after a lot of work in recent years regarding data handling etc. in Gekko ("data mode"), it is the feeling that the modelling parts of Gekko ("sim mode") could benefit from a modernization, too.

Gekko 3.0 supports a large number of in-built functions for dealing with for instance lists of data, supporting list syntax like for instance `#m1 = #m2.extend(#m3).unique().sort()`. If equation and model objects were supported in Gekko, a similar kind of syntax could be used to deal with model "wrangling", that is, combining models, adding and removing equations and blocks of equations from models, and also handle different types of equations. Since Gekko lists may contain any kinds of objects, the user could operate on lists of equations, lists of models, etc.

At some point it could perhaps be interesting to introduce dataframes in Gekko proper, as a new type of object.

Apart from this, there is still a need for consolidating and providing user documentation regarding Gekko 3.0, and further polishing is needed. Work on the interfaces with Excel, R, Python, GAMS, etc. is expected to continue, because it is deemed important that Gekko can operate relatively smoothly in conjunction with other data handling and modelling software packages.