Gekko steering committee meeting, October 29, 2020 (virtually on Zoom)[1]

Proposed agenda

Part 1

1. Welcome + choice of minute taker + approval of last year's minutes.
2. Editor's report regarding 2019-20.
3. Status regarding Gekko 2.4 and 3.0
4. Feedback regarding Gekko 2.4 and 3.0
5. Prioritization of main lines regarding 2020-21

Part 2

6. Commentaries regarding detailed checklist + prioritization of this
7. Other potential users
8. Status regarding documentation, help systems, etc.
9. Organization and choice of editor for the next period
10. Date of the next meeting + any other business

[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.

Re item 5. Main lines

1. Continue work on daily and weekly frequencies, implement the blueprint on conceptual questions regarding this.
2. Make new DECOMP and pivot of array-series available for users.
3. Make new Gekcel project (Excel add-in) available for users.
4. Further work on the Apache Arrow data interface (mostly a question of how to design the dataframes that exchange data back and forth between Gekko and R, Python and others).
5. MODEL and SIM improvements, so that .frm files also support the array-series syntax of Gekko 3.0, for instance implicit looping over sets, sums over sets, $-conditionals, etc. Static simulation possibilities (in a sense removing lags and solving the model for one period to obtain long-run values). Doing more means than goals (interdependence between goals) via addition of a special model block.
6. Related: equation and model objects. This would make it much easier to combine equations and models in different kinds of ways, and new in-built functions could be used to handle and transform models. Syntax could be something like for instance: `#model2 = #model1.removeblock('housing').addeqs(#housing_new_eqs);` This makes it possible to operate on model equations in a list-like fashion with special in-built functions to deal with all the particulars of model definitions, equation types, etc. Experiences from the modelling enhancements of Gekko 2.5.x could be used.
7. Examples collections etc. for Gekko 3.0. More guided tours, example command files, best practices. The document "The sun is always shining in Gekko" adapted for Gekko 3.0. This is essential for new users.
8. Better error messages in Gekko 3.0
9. Improved translator from 2.2/2.4 to Gekko 3.0. Quite a lot of basic syntax differences can already be done with the existing TRANSLATE command, but it is far from perfect. See more on syntax differences and translation here.
10. More advanced PLOT windows.
11. Seasonal adjustment could perhaps be done via the R package RJDemetra. This package handles all the tedious interface stuff in relation to JDemetra+, and Apache Arrow files could be used for data interchange. Using R would make the project much more manageable resource-wise.

Re item 6. More detailed check list

1. Extra allowed symbols in variable names in Gekko? For instance with a prefix like `¤x` for the structural level of `x`, or as a suffix like `x~s` or `x|s` (could also be used for seasonally adjusted levels, etc). Such symbols generally interfere poorly with data interchange to/from other software packages, and there is also the question of the availability of the symbols on different keyboards (for instance, ¤ does not exist on French or English-language keyboards). A possibility could be to treat a name like `¤x` as a simple syntactical short-hand for `x_s`, `x__s` or similar. Note that there is a conceptual difference between a variable name and an index name. Indices are typically much more flexible than variable names, try for instance this in Gekko 3.0: `x = series(1); x['¤~%æøå'] = 100; prt x;`
2. Metadata, more possibilities?
3. Logic regarding ini files, gekko.exe parameters, procedure libraries, etc. What should be run with RESET/RESTART? Also: namespaces and packages. User-developed procedures/functions could possibly be hosted on Gekko's website.
4. Showing Gekko options in a clickable tree structure with explanations.
5. Gekko working environment: more remote control of Gekko (e.g. Sublime Text or Spyder) and/or improvement of the Gekko input window, inspired by for instance RStudio? Autocomplete of variable names in the Gekko window? Other autocomplete or highlighting?
6. Better description of Gekko table syntax, especially the part where the user can custom design a table cell by cell.
7. More advanced general Gekko solver.
8. Improving the interface to StatBank Denmark? For instance interactive selection from tables and their fields/columns? Write to PC-Axis format? Interface to other online databases like for instance Jobindsats?
9. In the longer run, implementing dataframes in Gekko proper could be useful. This could be done via the Microsoft C# dataframe project (Microsoft.Data.Analysis), which also uses the Apache Arrow format as backend.