

Re item 2. Editor's report regarding of 2022-23¹

Since the last steering committee meeting, error messages have been improved (hopefully) quite a bit for Gekko $\geq 3.1.14$. The messages are now more compact, and – in addition to counting parentheses etc. – the Gekko parser has been rewired to produce less confusing error messages (hopefully). Also, Gekko 3.0 identifies if Gekko 2.4 syntax is inputted, and if so tries to offer a translation.

The new DECOMP module for GAMS models like MAKRO was finished in January 2023 (in Gekko 3.1.14), including support for .frm models like ADAM, and including a graphical user interface to navigate and link up equations. See here for a guided tour: <https://www.t-t.dk/gekko/test/Decomp1.pdf>.²

The data tracing module is up and running in the upcoming Gekko 3.1.16, but is at the moment deactivated as default (activate with “option databank trace = yes”). The details of data tracing proved harder than expected, among other things because traces of series that do not exist anymore may or may not be needed, and because traces “shadow” each other according to update periods. The current data tracing module is experimental, but as soon as it is stable enough, it will be activated as default. (The module is not yet able to handle array-series, but will be soon).

Databanks and models in Gekko $\geq 3.1.14$ are now read with an internal cache as backbone. For .gdx files this means that Gekko will first check if the file is already known from a previous read, and if so, Gekko reads the databank from so-called protobuffer files (in parallel). The speedup can be a factor 3-5x, depending upon the pc used. (The inner workings of this cache can be tuned in the future: at the moment it uses a fixed number of 4 threads at a time.). The 127 MB databank makro0.gdx from the DECOMP guided tour reads in about 2.5 s on a standard laptop (Intel i5-1235U) without cache activated, whereas it reads in about 0.7 s with cache turned on (which is default).³

Autocomplete of variable names is activated with [Tab] or [Ctrl+Space] in the user interface. It works for normal series or array-series. For instance, “prt x” followed by [Tab] will show series starting with ‘x’, and “prt x[” followed by [Tab] will show elements of the array-series x. Labels are shown, too.

Some basic program tracing can be shown with “option global program trace = simple”, but beware that this option can only be put in a gekko.ini file next to the gekko.exe file. Alternatively, use the Gekko menu Data, Trace Program. When activated, after something is

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>.

² The DECOMP guide is in Danish language. In the longer run, the guide will be integrated into the Gekko help system (in English).

³ As a test regarding RAM, the 127 MB databank makro0.gdx from the DECOMP guided tour has been opened as 100 instances at the same time (same session), which proceeds without issues on a 24 GB RAM normal laptop. On a machine with 8 GB RAM, Gekko starts to stall after opening about 50 of these databanks (Gekko code: `for val %i = 1 to 100; open <gdx> makro0 as b{%i}; end;`). Reading the 48 MB makro.zip scalar model from the DECOMP guided tour takes about 27 s without cache, and around 7 s with cache. (Gekko code: `model <gms> makro.zip;`).

RUN in Gekko, the program will show which command files were called, which databanks were read, etc. In a sense, this tells the user which files were “touched”.

SPLICE has been improved a lot, as a result of a project initiated by the Central Bank of Denmark. Users may now splice more than two series at the same time (“multi-splice”), optionally using overlap periods with > 1 observation. The new splice offers absolute and three kinds of relative adjustments.

For GAMS, the user can now copy a GAMS equation/expression and paste it as Gekko code. In Gekko, use right-click, Paste special, Paste GAMS --> Gekko. This will transform, for instance, a GAMS equation/expression like $y[j, t] = \text{sum}(i, x[i, j, t])$ into the Gekko-equivalent $y[\#j] = \text{sum}(\#i, x[\#i, \#j])$. This pasting functionality uses the existing inbuilt GAMS to Gekko translator (among other things also used for showing equations in DECOMP).

Since the last meeting, the Quarterly National Accounts have started to use Gekko 3.0 instead of AREMOS in “production” mode, and the Central Bank has more or less migrated from AREMOS to Gekko 3.0, too. In addition, the GAMS module that provides data for MAKRO has been implemented in Gekko 3.0, too. Because of (in the light of) these migrations, quite a lot of smaller bugfixes and enhancements have been implemented in Gekko.

A two-page “Cheat sheet” has been developed, highlighting the most important syntax differences between Gekko 2.4 and 3.0. See here: https://www.t-t.dk/gekko/Gekko_cheat_sheet.pdf.

The free VS Code text editor (<https://code.visualstudio.com>) now implements Gekko syntax highlighting via a downloadable extension (in VS Code go to extensions, and search for “Gekko” in the Marketplace). The extension is work in progress, and perhaps in the longer run, step-by-step debugging and other more advanced features could be provided for Gekko command files. Note that there is also an existing Gekko extension for Sublime Text (https://github.com/MartinBonde/gekko_sublime).