

## Minutes of the Gekko steering committee meeting, June 17, 2021 virtually on Zoom<sup>1</sup>

Participants:

- Asger Olsen (Statistics Denmark/ADAM)
- Michael Osterwald-Lenum (Statistics Denmark/ADAM)
- Bahar Dudus Celik (Statistics Denmark /Quarterly National Accounts)
- Rasmus Bjerre (Ministry of Finance)
- Jesper Christiansen (Ministry of Finance)
- Dorte Grinderslev (Danish Economic Councils)
- Thomas Thomsen (Gekko editor)

Note: In the following, version “2.4” covers Gekko versions 2.2, 2.4, and 2.5.x, whereas version “3.0” covers Gekko versions 3.0 and 3.1.x. Most current development is on 3.1.x.

### **Item 1. Welcome + choice of minute taker + approval of last year’s minutes**

Thomas was chosen as minutes taker, and Michael as moderator. There were no comments regarding last year’s minutes.

### **Item 2. 2. Editor’s report regarding 2020-21.**

There were some questions about RESET vs. RESTART, and loading of gekko.ini files. At the moment, RESET does not load any ini files, whereas RESTART loads two ini files if they exist (first, one in the gekko.exe program folder, and next one in the working folder). If more fine-grained gekko.ini loading is needed, contact the Gekko editor.

Rasmus said that regarding the MAKRO model, from the point of view of the Ministry of Finance the MAKRO model setup has priority over existing setups. Regarding existing setups, he did not currently have any pressing Gekko development requests, and the software works fine from his point of view (this setup runs under Gekko 2.2).

Thomas mentioned that Gekko in 64-bit version (both Gekko 2.5.x and 3.1.x) seems to run ok now, and 32-bit Gekko will slowly but surely die out within a few years. Some users are happy to use Gekcel (Gekko-interface from Excel), and these users often use VBA for parts of the Gekko-Excel interface. The Quarterly National Accounts and the Central Bank of Denmark both use VBA to this effect, and Gekcel tries to keep up and distribute VBA-interfaces, too. For new Gekko users, Thomas advised to try out the new Gekko User Guide

---

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

(part of the help system from Gekko 3.1.10 and onwards). Lastly, Gekko library capabilities are under development and are practical regarding reuse of code, versioning, etc. (these work somewhat like R or Python packages). Thomas also mentioned a model blueprint, containing some proposals regarding how to design models more object-like, and a frequency blueprint with proposals regarding daily and weekly frequency implementations. Thomas finally mentioned the improved source code documentation. The source code has also been redesigned (refactored) and cleaned up, so that the Gekko project is a bit less person-dependent. Regarding error messages in Gekko, they are still pretty bad, but they are still high on the priority list.

DST/ADAM has held a course on data handling in Gekko 3.0 for the Quarterly National Accounts, and some ADAM courses for the fall of 2021 are planned. The Quarterly National Accounts have been busy regarding the daily production of data but hope to get the Gekko system (translated from AREMOS) up and running in “production mode” during the fall 2021.

### **Item 3. Status regarding Gekko 2.4 and 3.0**

Thomas mentioned briefly that 2.4 (2.5.2 to be precise) is now also available as a 64-bit version, and that this seems to run ok. Gekko 3.0 can also run as 64-bit version (from version 3.1.10 and on to be precise).

### **Item 4. 4. Feedback regarding Gekko 2.4 and 3.0**

Dorte said that the Danish Economic Councils have started using the new Gekko libraries as repositories for reusable Gekko code (functions/procedures). A library is much faster to load than a long command file with the same functions and procedures, so it helps their start up time and speeds up RESTART. In addition to being faster to load, it is easier to share a single library zip file than a collection of command files, and libraries help making sure that everybody is using the same functions/procedures. In continuation of what John Smidt said at last year’s meeting, Dorte repeated that error messages are important to improve.

Asger asked if the Danish Economic Councils are using daily timeseries now. They are, among other things they have made a rather large system that seasonally corrects daily unemployment timeseries (by means of OLS).

Rasmus said that the Ministry of Finance are running their medium-term scenarios on Gekko 2.2. They are not doing much data wrangling and are more using it for model solving, for which it works well. Jesper was plagued by a bad internet connection but wrote that they are using Gekko 2.2 too, and that it is stable, and for short-term projections they are mostly using the solver parts of Gekko. Both Rasmus and Jesper will consider the 64-bit version (2.5.2).

Asger said that Gekko mostly works well for the ADAM group, and that 64-bit is convenient. Michael added that it is convenient that Gekko 3.0 now remembers past commands after Gekko is closed and reopened (also in case Gekko crashes).

On behalf of the Central Bank of Denmark, Thomas said that they are still running MONA via Gekko, and that they seemingly slowly moving their AREMOS data handling systems to Gekko 3.x, too. They are using Gekcel quite a lot to interface with Gekko databanks.

## **Re item 5. Prioritization of main lines regarding 2021-22**

Item 1 (finish daily and weekly frequencies) will be finished soon.

Item 2 (decomposition). The new DECOMP now works more like a pivot table, which has to do with array-series, the MAKRO model, etc. (with the current DECOMP as a special case). This is also almost done, it is mostly a question of Thomas wrapping it up and writing a documentation for it.

Item 3 (error messages). Thomas felt that improving these is very important.

Item 4 (model blueprint). Maybe this is best suited to discuss in a subcommittee. The question of the best syntax to define a model with sub-models etc. is important to have an idea of the design, before actually building it. For instance, it would be nice if the design would correspond to other structures like databanks, lists, maps, etc.

Item 5 (solver). Some work has already been done on that, for instance more means than goals. Again, a subcommittee would be advisable.

Item 6 (interpolate). Thomas felt that the R package tempdisagg seems like state of the art, and that a convenient interface could be made.

Item 7 (seasonal correction). Thomas felt that in order to do this properly, JDemetra+ should be used, perhaps via the R package RJDemetra.

Item 8. Metadata. Thomas had already spoken to Asger, Michael and Dorte on this, and the idea is to provide a kind of track and trace regarding timeseries in databanks (how does the data originate?). Asger would like it to remember previous calculations in “tree rings”, for instance the last 10 calculations/expressions or so, and it should handle copying, too.

Item 9 (more guides, exercises, etc.). Asger felt that consolidating is still important, but asked if consolidating is more or less finished by now, including guides, examples, etc.? The source code has just been renovated and made up to date, but there is also the documentation, examples collections and so on. Thomas said that the “Gekko User Manual” part of the help system could use some renovation, but it helps that the “Gekko User Guide” (also part of the help system) is brand new and written recently. For example, in the user manual, the SERIES description is long and detailed, and could benefit from being rewritten, perhaps with a shorter main page and some sub-pages with details (LIST is another example of many details regarding one command). To answer the question Thomas said that

consolidating was still work in progress, and error message could also be seen as part of that.

Regarding error messages, Thomas said that feedback from completely new users is perhaps the most useful for him. Thomas suggested fixing the worst kinds of error messages (which he has an idea of: an example is the use of {}-curlies), and then releasing a version with in-built error reporting. That is, a mechanism for the easy reporting (via mail) of incomprehensible error messages, where Gekko will pack up the context of the error message and help mailing it. Michael suggested creating an errors-subcommittee in the beginning of 2022 to help avoid preventable situations where Gekko crashes or shows a lot of mystical red error messages.

To sum up, consolidation, documentation and error messages is still important. Dorte would like to prioritize the question of modelling syntax and components (how to assemble models and sub-models).

## **6. Organization and choice of editor for the next period**

Thomas was elected as editor for the next period.

## **7. Date of the next meeting + any other business**

Next meeting is planned for medio June 2022.

## **8. Commentaries regarding detailed checklist + prioritization of this**

Libraries and how Gekko is set up on a shared network was discussed. In the long run, it is the intention to release libraries developed as part of the Gekko project either on the Gekko homepage or on GitHub. Some components that are not overly complicated or overly speed-sensitive could be developed as Gekko functions/procedures rather than in-built functions and commands.

With more people working from home, any server access over VPN is painful, so it would be nice to be able to somehow synchronize a server version of Gekko and of libraries with local copies.

Regarding 2.4 vs. 3.0, translators, etc. Thomas said that he planned to take the best parser components of the AREMOS to Gekko 3.0 translator (which was quite advanced) and use them for the Gekko 2.4 to 3.0 translator, too. He envisioned a more complete translator that also handles {}-curlies, list expressions and other things better (from 2.4. to 3.0). Perhaps the translator could go from around 80% precision and approach 90% precision. And then the rest would be manageable to fix by hand. Thomas felt that a lot of the “problems” regarding this translation (and perhaps error messages too) has to do with the use of {}-curlies like {%x} or {#x}, and the distinction between for instance “PRT #m;” printing

the list items themselves (for instance a number of strings), or “PRT {#m};”, printing the variables that the list elements refer to by name. Contributing to this confusion is the fact that the syntactical use of {}-curlies in Gekko 2.4 is deeply inconsistent compared to 3.0. This was one of the main reasons that the syntax of 3.0 was adjusted.

Asger mentioned that he found the use or non-use of quotes and slashes for filenames in Gekko 2.4 cumbersome. Similarly, when using for instance the INDEX command, it is hard to remember if you must indicate databank name, or that if other banks are covered by for instance “INDEX \*;”. Getting a “0 variables found” is perhaps correct in a strict sense, but not too helpful if we know there are variables somewhere! Thomas would look into this.

Asger said that even after migrating for instance the data revision system, it would still be nice with a proper translator to be able to translate and view older versions of the data revision files. Regarding databank formats, Thomas said that all Gekko 3.x.x versions read previous databank formats, and that Gekko versions 2.5.x read all 3.x.x databanks, so the older Gekko’s are actually forward-compatible.

Asger and Michael would be willing to try out an improved translator in the fall of 2021 for the data revision system.

Asger mentioned that the statbank.dk (DOWNLOAD) dimension names and element are very verbose, and asked if it would somehow be possible to abbreviate these names. Something automatic or semi-automatic would be nice. In the longer run, it would also be nice to be able to download series in a binary non-px format, for instance the new arrow format. Regarding arrow and how to best organize Gekko array-series as dataframes or arrow files, it would perhaps also be beneficial to take a look at the packages statsDK (R) or PyDST (Python) which offer download from statbank.dk. Regarding the dataframe design, there is the question of whether these should be broad or narrow. For instance, the ADAM databank could have a column for each variable name (broad), or there could be one column called “name” which is just repeated over and over for each observation.

Asger spoke about a potential problem about PC-Axis files (px) lacking short dimension codes, which could be used to avoid too long series and array-series dimension names. Apparently, there are some suggestions (from Lars Knudsen) regarding decorating px files with the shorter abbreviated codes, too. The problem regarding verbosity in px files seems to concern only the dimension names, not the names of the dimension elements. Are the Swedes (SCB) doing anything regarding this?

Asger proposed the possibility of using wildcards in array-series, like x[a, b\*, c?], for the INDEX command etc.

Asger thought that autocomplete of variable names would be helpful.

Thomas said that a JobIndsats interface would be convenient, but there are some questions regarding long dimension names here, too.

Options in a tree structure would be much easier to do now, because the internals of the OPTION command have been reworked and are therefore easier to represent in other ways.

Dorte did not think it was that pressing, among other things because it would not work from a text editor anyway.

It would be convenient to be able to design the table schemas in Excel. Asger called for an easier way to produce tables. DØRS does this by means of having (in the same workbook) a data worksheet (filled from the SHEET command) and a design worksheet where the data is presented.

Dataframes as a variable type inside Gekko would be convenient in the longer run, supporting a subset of pandas or dplyr. For now, this can wait.

.NET Core is the future, and Gekko needs to migrate to this sooner or later. Thomas hopes for good translators at that time.

Michael asked about a Mac edition. The engine itself could run on a Mac relatively easily, at least after migrating to .NET Core. But the problem would be the user interface, and this applies to Linux, too. Versions for phones and tablets are the same story: .NET Core could run the Gekko engine on iOS and Android, but the user interface would be the question. Some people would propose to code the Gekko user interface in HTML5+JavaScript and have users run Gekko from a browser-interface, kind of like Google Sheets etc. This would in principle also be possible, but Thomas is against this right now. Browser incompatibilities would consume too much developer time, and the end result would probably still feel kind of cheap.

Dorte would think about use cases where a web version or Mac/Linux/Chromebook/iOS/Android version would make sense. Actually, Gekcel is an example of cutting out the Gekko engine and rewiring it with some other interface (in this case Excel).