

Minutes of the Gekko steering committee meeting, September 2, 2024, at Statistics Denmark¹

Participants:

- Asger Olsen (Statistics Denmark/ADAM)
- Michael Osterwald-Lenum (Statistics Denmark/ADAM)
- Anette Borge (Statistics Denmark/ADAM)
- Søren Gjedsted (Statistics Denmark /Quarterly National Accounts, QNA)
- Jonas Dan Petersen (Statistics Denmark /Quarterly National Accounts, QNA)
- Anders Køhlert Larsen (Ministry of Economic Affairs)
- Grane Høegh (DREAM/MAKRO)
- Dorte Grinderslev (Danish Economic Councils, DØRS)
- Nikolaj Mose Dreisig Hansen (Danmarks Nationalbank, central bank)
- Allan Sørensen (Confederation of Danish Industry, DI)
- 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. Thomas promised to produce the minutes sooner this time. There were no comments regarding last year’s minutes.

Regarding “fencing” and the root() function, making it easier and less worrisome to move around or copy a given file structure of .gcm files, Michael mentioned that the feature had been very helpful regarding the major revision of the National Accounts. Fencing will also catch any hard-coded absolute paths trying to break out of the fence.

Thomas mentioned VS Code as a very good (and free) text editor that MAKRO now also use. It has a Gekko extension for code colouring etc., just search for “Gekko” under VS Code extensions.

Asger mentioned briefly what item 9 (Gekko risk assessment paper) was about, one of the questions being “Can Gekko always handle a new Windows version?”, or “What happens if Thomas no longer can or wants to maintain Gekko?”. Among other things, this will be discussed on an upcoming ADAM board meeting at Statistics Denmark. The Microsoft .NET Framework will be slowly phased out, and .NET Core is the future, but Thomas emphasized that the phasing out of .NET Framework will probably be a slow and gradual process, not

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>.

expected to show real signs before 2030, and the risk assessment paper should be seen as due diligence in relation to the longer run.

Grane said that from their point of view, especially if migrating to .NET Core is costly resource-wise, they hoped that Thomas would consider the possibility to migrate to a data-science language like for instance R, Python or Julia, inside of which Gekko could run as a package. Thomas mentioned that regarding Python or R, a lot of the core parts of Gekko (for instance solver, decomposition etc.) would not be able to run directly inside these languages proper because Python or R are both slow/interpreted languages, so these core parts would still need to be written in another faster language like for instance C or C++, which amounts to the so-called “two-language problem” (using both a slower and a faster language at the same time). Julia solves the two-language problem, but the question is whether Julia gets enough traction relative to Python or R? Asger mentioned that pooling of resources in relation to this could be useful (Gekko is free to use, but the development must be financed). Thomas said that estimating the migration to .NET Core resource-wise is relatively easy regarding 80-90% of the code (the easy code that does not even need much translation), but the harder parts, for instance the Gekko user interface and the parts that interact with special .NET API's or third-party software, can create more of a headache.

Item 2. Editor's report regarding 2023-24.

Thomas reviewed the report, and also showed a short demo of data traces, and a prototype regarding a html equation browser for GAMS models.

Item 3. Status regarding Gekko 2.4 and 3.0

A Gekko 3.2 version is imminent, and Gekko 2.4 is no longer developed. Regarding ADAM, a few things were back-ported from Gekko 3.0 to Gekko 2.4 though, to help out with the ADAM data revision systems.

Item 4. Feedback regarding Gekko 2.4 and 3.0

Asger felt that it was “stupid” that the ADAM group still run a lot of their programs under Gekko 2.4. Thomas said that the best way to pursue this is probably to allow a mix of Gekko 2 and Gekko 3 programs to run the ADAM data revision system, switching out components one by one gradually, rather than all at once.

It was mentioned that ADAM ships with Gekko 2.5.2 as software. Grane felt that these ADAM users would also be more likely to switch if they received an ADAM version running under Gekko 3. Asger said that the ADAM data revision system would probably be migrated before the model assembly system. Thomas mentioned that in principle, all the model assembly systems could remain in Gekko 2 for the time being, whereas the shipped package could use Gekko 3 (the .frm model format has not been changed between Gekko 2 and Gekko 3, and the databanks are fully compatible).

Grane mentioned that the Ministry of Finance has some rather large systems written in a mix of languages that interact with the ADAM databank, and these systems ought to be rewritten for the MAKRO databank (and the Gekko parts ought to be written in Gekko 3 rather than Gekko 2).

Jonas mentioned that their types of programs run slowly compared to AREMOS, and he would like to find out why. In the same vein, Dorte felt that sometimes a Gekko window gets “tired” over a period, perhaps accumulating memory use (RESET/RESTART is not enough, the window needs to be reopened). Also, option <dyn> slows down programs. Thomas said that Gekko was built to be at least as fast as timeseries programs like for instance AREMOS, TSP or EViews, so any slowness relative to such programs will be investigated and prioritized.

Dorte said that data traces are a massive gain when trying to locate errors in larger databank systems, and with data traces her colleagues can actually figure out themselves where an error occurred. Nikolaj also felt that data traces are very helpful, instead of trying to track down variable names in .gcm files in a more manual guessing process.

Item 5. Prioritization of main lines regarding 2024-25

Regarding (a1), people felt that error messages are ok.

Regarding (a2), data traces also seem ok for now.

Regarding (a3+a4), Grane felt that when the scalar models are running well together with equation browser and DECOMP, it will be a very useful package for new MAKRO users. This ought to be prioritized.

Regarding (a5), Gekcel seems to run ok.

Seasonal adjustment (a6) has the problem that even if an interface to JDemetra+ (via the so-called “cruncher”) is made, Statistics Denmark still use certain business-day or outlier-corrections etc., in a sense “hand-held” corrections. Quite a few Gekko users just use X12A, from Gekko. The value of the interface is therefore a bit uncertain.

INTERPOLATE with Denton (a7) will be refined in order to be more robust regarding differences in levels.

Regarding (a8+a9), Grane said that PLOT is very practical in daily use, but for publishing they use for instance Python.

Xlsx file templates for tables (a10) is not prioritized right now.

Regarding the modelling parts of Gekko (a11), Michael said that Tony from the ADAM group had asked about a new model definition format to replace/enhance the current .frm format. Thomas said that a blueprint² about this was written some years ago, detailing possible

² https://www.t-t.dk/gekko/docs/blueprints/Gekko_model_blueprint.pdf

syntax enhancements, model blocks, loops, pre-/post models, etc. Michael proposed that users with an interest in the model format could form a working group to discuss advantages and disadvantages, perhaps with the blueprint as an entry point.

Grane said that regarding MAKRO, the model is not as such married to GAMS, but if they should move away from GAMS, they would look at software like for instance Julia, where MAKRO could be solved via an external solver, and inside of which the best features of Gekko could perhaps live on in the form of a package. Since the Ministries of Finance and of Economic Affairs plan to migrate from ADAM to MAKRO, and since MAKRO is solved in GAMS, there is the question of the longer-run value-added of such modelling enhancements in Gekko.

Regarding guides etc. (a12), Thomas felt that these were ok, but that he would read through the help system before releasing Gekko 3.2. Dorte said that concrete examples are very helpful for learning new features.

Nikolaj would like a “refresh” button for DECOMP, which is easy to implement.

A Gekko course could be useful for DI, covering basic functionality. The data course from the ADAM group could perhaps be adjusted and reused (Thomas also mentioned the Gekko user guide from the help system). DI uses Gekko for running MONA, but could perhaps also use it for data handling. Danmarks Nationalbank could also be interested in a course, perhaps in October 2024.

Item 6. Organization and choice of editor for the next period

Thomas was elected as editor for the next period

Item 7. Date of the next meeting + any other business

Beginning of September 2025 was suggested.

Item 8. Commentaries regarding detailed checklist + prioritization of this

Regarding (b3), if the help system is good enough, including examples, new features will sooner or later be discovered. A rolling log at Gekko start up would perhaps be seen as noise, and people would not read it. In contrast, when reading about some Gekko statement, users are more open regarding “did you know that...?” cross-references and examples.

Regarding (b4), users can also just exchange library zip files or procedures/functions by mail. Putting these online raises another question, namely quality control of the files.

Regarding (b6)/(b7), if reading Apache Arrow files is relatively easy to do (now that Gekko already writes them), this could perhaps be pursued.

VS Code extension for Gekko (b8) mostly performs syntax colouring and works ok for now.

Item 9. Gekko risk assessment (long-term software reliability)

This paper was further discussed.

The reason why it would be good to migrate the main Gekko window and the PLOT window to vector-based graphics (like the Gekko DECOMP window and others, which are made as vector-based WPF windows), is that future high-resolution screens are always supported. Also, WPF just looks better and is more future-proof than the older technology (WinForms).

32-bit Gekko ought to die out, as modern cpu's are almost always 64-bit. Migrating to .NET Core would be done in 64-bit only, but until then, perhaps 32-bit releases could be supported a few more years. One thing to note though is that if Gekcel (Excel add-in) needs to run on a Windows pc with 32-bit Office, Gekko needs to be 32-bit, too. So when 32-bit Office is sufficiently phased out, that could perhaps signal the end of life of 32-bit Gekko.

Regarding Gekko as a "package" inside an environment like R/Python/Julia or similar, Thomas would add a section on this to the risk assessment paper. Besides technicalities, it is also a question of resources. If migrating to .NET Core takes a month, but migrating to for instance Julia takes a year, this resource difference must also be taken into account.

Item 10. Other potential users

Finance Denmark or CRT could maybe be candidates. The ADAM group has conducted courses for some Moroccans, teaching them ADAM/Gekko types of models.

Item 11. Status regarding documentation, help systems, etc.

The source code documentation is important in the sense that it mitigates the risk of depending upon a single developer. The previous source code documentations were from 2014 and 2021, respectively, and Michael hoped that the next one would be out before 2028 (if the gap is extrapolated). The thing about source code documentation is that it is hard to write it dynamically while writing the source code, because the documentation risks becoming obsolete due to code changes.