Minutes of the Gekko steering committee meeting, September 15, 2025, at Statistics Denmark[1]

Participants:

- Ingeborg Vind (Statistics Denmark/ADAM)
- Jes Asger Olsen (Statistics Denmark/ADAM)
- Michael Osterwald-Lenum (Statistics Denmark/ADAM)
- Dawit Sisay Temere (Statistics Denmark/ADAM)
- Søren Gjedsted (Statistics Denmark /Quarterly National Accounts, QNA)
- Jonas Dan Petersen (Statistics Denmark /Quarterly National Accounts, QNA)
- Grane Høegh (DREAM/MAKRO)
- Mads Wulff Harslund (Ministry of Economic Affairs)
- Dorte Grinderslev (Danish Economic Councils, DØRS)
- Sofie Holme Andersen (The Economic Council of the Labour Movement, ECLM)
- 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 "3.2" covers Gekko versions 3.0, 3.1.x, 3.2 and 3.3.x. Current development is on 3.3.x.

Item 1. Welcome + choice of minute taker + approval of last year's minutes

Thomas was chosen as minutes taker, and Michael as moderator. The minutes from last year were approved without comment.

Item 2. Editor's report regarding 2024-25.

Thomas mentioned that regarding the revised help system implemented in Gekko 3.2 and 3.3.x, this help system can also be found online on the Gekko homepage, together with the corresponding `Gekko.chm` file. On a given Gekko < 3.2 installation, the user can just manually overwrite the existing `Gekko.chm` file with the new file, if this is easier than having a new Gekko installed.

Given that the new PLOT window is now vector graphics based (WPF), only one non-vector graphics window remains: the main Gekko window.

[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.

A slowness issue regarding timeseries operations in some QNA programs had been fixed. QNA are migrating to Gekko 3.2 soon, after which the fix can be evaluated.

Regarding eps (small value representing a number that is normally 0), the concept is inspired by GAMS, but it seems that something similar has previously been used in Gekko (and perhaps AREMOS) data revision programs, for capital data. In Gekko, however, eps values have certain arithmetical rules, though, so the concept is built into Gekko more deeply (eps + eps = eps, 1 + eps = 1, exp(eps) = 1, eps/eps = missing, etc.). One idea regarding eps values is that if a timeseries has an eps value in some period, it can be assumed that it has eps value in some new data period, too.

Thomas mentioned that data-traces seem to work well in general, and users find them useful. Danmarks Nationalbank (Central Bank) has had some issues about "forgotten" data-traces, which ought to be investigated. For MAKRO users, note that data-traces are only present in `makrobk.gbk`, not in the (otherwise identical) `makrobk.gdx`.

Thomas showed a demo of the MAKRO html browser (a first test here).

Item 3. Status regarding Gekko 3.2

Gekko 3.2 is stable, well-tested and well-documented. Thomas advised to upgrade to 3.2 (or 3.3.x) if possible, and to base new projects on 3.2.

Item 4. Feedback regarding Gekko 3.2 and 2.4

Regarding models written in Gekko's .frm format, ADAM is being phased out, and the question is what happens in the longer run to models like for instance SMEC and MONA.

According to DØRS, Gekko 3.2 runs well and is stable, but there are some issues regarding mass-production of plot files, something about aspect ratios. This will be investigated, but it should be noted that the plot engine is unchanged (gnuplot), and if needed the old plot window can be revived with `option bugfix plot = no`. DØRS also reported that it sometimes seems like Gekko gets "tired" (slow) when the same application is used over a longer period (days), and the only remedy is to close and reopen Gekko (RESET/RESTART not enough). This could be RAM issue, a Gekko memory leak, or perhaps the upper part of the Gekko window filling up (try CLS).

DISP (internal equation browser) for MAKRO has improved lately. MAKRO users felt that using DECOMP, PLOT etc. in Gekko is now essentially as easy as it is regarding for instance ADAM-like models.

QNA feel that JDemetra+ is a bit heavy to use for seasonal adjustment, with point-and-click etc. Just using flat files as interface would probably be preferred. The Ministry of Economic Affairs has had an Excel add-in developed, based upon JDemetra+. Thomas will try to find out if this could somehow also be plugged into Gekko. The JDemetra+ settings used by QNA could probably be shared.

A user attempting to use array timeseries in Gekko 2.x was advised against it. Array-series are only very superficially supported in Gekko 2.x, and 3.2/3.3.x is advised.

The new PLOT window is popular in general, because of its ease of use. Michael asked if the new PLOT-window also suffers from the dreaded “red cross” sometimes seen in the old window because of missing data. This ought not to be the case, because the component has been rewritten.

Users felt that Thomas is generally fast to react regarding questions/advice.

Item 5. Prioritization of main lines regarding 2025-26

Regarding the prioritization process in general, see also Item 6.

Regarding (a2), arrow files, this will probably be done as part of a project financed by Danmarks Nationalbank (Central Bank). Array-series (a3), error messages (a7), and data tracing (a8) seem ok for now. Gekcel (a9) seems to work well. Regarding (a12), documentation, the Gekko 3.2 documentation (help system) has been completely revised, so this should be ok (the help system includes an introductory guide for beginners).

This leaves Python package (a1), html equation browser (a4), decomp/flowgraph (a5+a6), seasonal adjustment (a10), and interpolate (a11) for prioritization. Thomas said that he considers html browser (a4), decomp/flowgraph (a5+a6), and the in-built equation browser (DISP), as part of the same general component: model viewing and analysis.

It was decided to pursue the Python package (a1) development, starting out with the most simple interface and building from there. In parallel, the model viewing and analysis components will be refined, too, since there is a demand for this from MAKRO/ADAM/SMEC/MONA users. Important note: since the Python package depends upon a functioning Gekko application (the so-called Gekko-exe), both Gekko-package and Gekko-exe will be developed in parallel, and new Gekko features will benefit both. So Gekko-exe is *not* being phased out because of Gekko-package.

Regarding risk assessment and the 2025 roadmap (cf. Item 8): if migration to .NET Core is a necessary condition for both Gekko-exe and Gekko-package to run in a stable manner after 2030 (given the continued use of C# as a programming language for Gekko), some users felt that perhaps the migration to .NET Core (the so-called step 3 of the roadmap paper) ought to be done before – or at least not too long after – steps 1 and 2 of the roadmap paper, or perhaps in parallel. See Item 8 for details on that discussion.

Regarding seasonal adjustment (a10), Thomas will investigate whether there is a stand-alone for X13, to supplement the existing X12A in Gekko. If there is, and if it is reasonably simple to use it in Gekko similar to X12A from the Census Bureau, this ought to be tried out. Alternatively, JDemetra+ could be tried (with their so-called “cruncher”).[2]

Regarding interpolation (disaggregating from lower to higher frequencies (a11)), the new Gekko interpolate features ought to be tried out and evaluated by users (including the so-called ‘olsette’ method), before further resources are used. In that regard, (a10) is prioritized over (a11).

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

Thomas was elected as editor for the next period.

The steering committee meetings are a bit formal regarding election of editor, etc. This, however, has to do with the so-called “foundation” and the project being open-source (and the editor trying to guide to project in one direction). But things have changed since the first steering committee meeting in 2011, and nowadays the editor is typically on-site weekly, talking regularly to the two current main project sponsors or their representatives.

In the light of this, and given the budget constraint on the editor’s available time, the editor envisioned some changes going forward: (a) Keep a yearly meeting in September. (b) Make sure the main sponsors are present at the meeting. (c) Invite anyone that normally receives news about Gekko (and steering committee meeting invites) to this open meeting. (d) Keep the meeting material simple and to the point: an agenda and a brief editor’s report. (e) Keep the minutes brief and to the point.

Ingeborg mentioned that in reality, the steering committee meeting is more of a user forum nowadays, because the priority list (Item 5) is more or less being continuously agreed upon between the editor and the primary contributors. Still, suggestions and feedback are always very welcome, of course, from any user – and such feedback is important regarding Gekko’s features and choosing the right direction. The Gekko editor is as always willing to implement new features based on user needs, provided the necessary funding is available.

In general, most users felt that the volume of written meeting material etc. could be adjusted, but they would very much like the open meetings to continue going forward, in order to be able to discuss issues/projects regarding Gekko etc., and perhaps as a side effect exchange information regarding modeling etc. in general.

It was suggested that one could consider using AI to automatically produce minutes.

[2] Update: there *is* a X13 exe file available for Windows, with a simple file interface like X12A. But the so-called “cruncher” for JDemetra+ is also a possibility, and probably not that difficult. In practice, X13 and JDemetra+ are expected to yield very similar results, but because QNA already use JDemetra+, being able to reproduce official QNA figures will be advantageous in itself.

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

Middle of September 2026 was suggested. In the light of what was mentioned under Item 6, regarding the next September 2026 meeting, Thomas will select a date and announce it before the summer vacation of 2026.

Item 8. Gekko Roadmap 2025 (includes risk assessment)

Note: As mentioned under Item 5, any development of a Gekko-package (for instance in Python) will not affect the availability of Gekko-exe (stand-alone application). Any new Gekko features will benefit both Gekko-package and Gekko-exe, and both will be released in parallel.[3] Users enjoy the clickable Gekko popup windows (like plot, decomp, flowgraphs), and it is the intention to provide these for both Gekko-package and Gekko-exe.

As also mentioned under Item 5, some users felt that perhaps the migration to .NET Core ought to be done first (or at least not too late), because without this migration, the stability of both Gekko-exe and Gekko-package could be an issue after 2030. Thomas was asked how many resources this migration could demand. He said that 80% of the code can be machine translated or used without changes, but the question is always the last 20%, especially regarding the GUI (graphical user interface) and external components used. Migrating the main Gekko window to vector graphics (WPF) could take some time, though. Given sufficient interest, Thomas could try to estimate the cost of .NET Core migration, where of course automatic translators and AI can help out.

Thomas said that young (newly educated) users prefer a coding environment that they know already, that is, for instance running Python from VS Code (or R from RStudio, etc.). Thomas also had to make note of the fact that Statistics Denmark encourage Python or R package use, and that the MAKRO model runs from inside a Python environment (using the so-called gamY component from DREAM-tools). Therefore, Python makes sense to start out with.

A user asked if it's possible in Gekko-package to "jump around" and execute lines of code non-linearly in VS Code, like in the main Gekko window (thereby providing a user friendly so-called REPL). This can be done easily in VS Code by simply using `Shift` + `Enter` (marking a block of lines to execute is possible, too), which will then execute that particular piece of Python code, representing for instance one or more Gekko statements. That workflow is much like the default behaviour of the main Gekko window, except for the use of `Shift`. So, regarding this, Gekko-package programs do not need to be executed in a linear fashion, but this is of course possible, too (debugging line by line).

[3] Unless the value added parts of Gekko are rewritten in for instance C++ (for Python), and Gekko-exe is scrapped. In the 2025 roadmap, migrating to for instance C++ is mentioned as a possible "ambitious" stage 4 in the Gekko-package project, but a major drawback is mentioned too: "A general drawback related to using another system language than C# for Gekko-package is that in that case, Gekko-exe and Gekko-package would diverge regarding the code base". Because C# is not that much slower than C++, and because C#.NET is actually a quite popular ecosystem, stage 4 may never materialize.

The so-called Python.NET package allows you to both call C# (.NET) from Python and vice versa. For some tasks, Gekko could probably benefit from the use of other Python packages, like for instance Pandas/Polars for dataframes, NumPy for arrays/matrices, etc. In that case, Python.NET can be used to call these Python packages from inside C#, and since the calls are implemented via C#, such features will be available for both Gekko-package and Gekko-exe. Regarding Gekko-package, users can of course make their own functions in Python, to interact with Gekko in different ways (for instance, making these functions emit suitable Gekko code in the form of text strings). Such developments will not be available for Gekko-exe, but this is not conceptually different from the fact that the development of VBA functions used together with the Gekko Excel add-in (Gekcel) does not benefit Gekko-exe per se. Still, a GitHub repository for Gekko-package could be established, so that valuable Python functions that interact with Gekko can be shared among users.[4]

For existing Gekko users thinking about migrating (parts of) their Gekko systems to Python, a simple Gekko-package could provide a rather easy-to-implement stepping stone regarding migration to Python, after which deeper integration into Python could be implemented in a stepwise fashion.[5]

Item 9. Other potential users

It was mentioned that The Danish Ministry of Taxation have tried to install Gekko, but were not allowed because of IT policy. Thomas will contact them.

A delegation from the High Commission for Planning (Morocco) are using Gekko for an economic model, in a project together with Statistics Denmark (ADAM).

Item 10. Status regarding documentation, help systems, etc.

As mentioned, the help system has been fully revised, including its introductory user guide section. User felt that code examples are helpful in general, and the editor mentioned that quite a lot more examples had been added while revising the help system for Gekko 3.2.

[4] So-called “pull requests” could be useful, with for instance the Gekko editor validating these before merging them in. This repository would mainly contain Python code, in contrast to the official Gekko repository that mainly contains C# code.

[5] As mentioned in section 2.2 of the roadmap: “… a Gekko-Python package could act as a stepping stone regarding the migration of existing systems into Python, and it would definitely be much easier, less error-prone and less time consuming than porting a large Gekko-exe system directly from Gekko syntax to for instance Python/Pandas/Polars/NumPy, which are not particularly well suited for timeseries handling.” Regarding data flows between Gekko and Python, csv, xlsx, arrow or gdx files could be used to communicate with Python dataframes (or more ambitious: provide a Python API for Gekko .gbk files).

Regarding courses, examples, etc., new users were advised to take a look at the Gekko User Guide (part of the Gekko help system). This is intended for new users. Some of the Gekko users have internal course material available and would probably not mind sharing it.