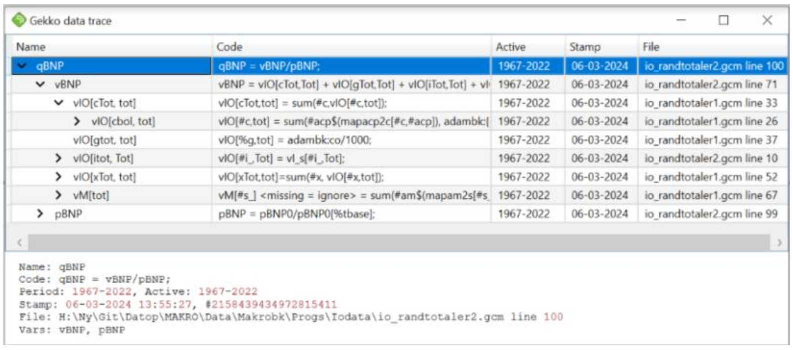
Re item 2. Editor’s report regarding of 2023-24[1]

Generally, in this period, quite a lot of work has gone into data revision programs related to the MAKRO and ADAM models/databanks, and Gekko/GAMS questions related to that. Also, work on migrating the ADAM data revision programs from Gekko 2 to 3 is ongoing.



Name	Code	Active	Stamp	File
qBNP	qBNP = vBNP/pBNP;	1967-2022	06-03-2024	io_randtotaler2.gcm line 100
vBNP	vBNP = vIO[cTot,Tot] + vIO[gTot,Tot] + vIO[iTot,Tot] + vI	1967-2022	06-03-2024	io_randtotaler2.gcm line 71
vIO[cTot, tot]	vIO[cTot,tot] = sum(#c,vIO[#c,tot]);	1967-2022	06-03-2024	io_randtotaler1.gcm line 33
vIO[cbol, tot]	vIO[#c,tot] = sum(#acp$(mapacp2c[#c,#acp]), adambk:(	1967-2022	06-03-2024	io_randtotaler1.gcm line 26
vIO[gtot, tot]	vIO[%g,tot] = adambk:co/1000;	1967-2022	06-03-2024	io_randtotaler1.gcm line 37
vIO[itot, Tot]	vIO[#i_Tot] = vI_s[#i_Tot];	1967-2022	06-03-2024	io_randtotaler2.gcm line 10
vIO[xTot, tot]	vIO[xTot,tot]=sum(#x, vIO[#x,tot]);	1967-2022	06-03-2024	io_randtotaler1.gcm line 52
vM[tot]	vM[#s_] <missing = ignore> = sum(#am$(mapam2s[#s_	1967-2022	06-03-2024	io_randtotaler1.gcm line 67
pBNP	pBNP = pBNP0/pBNP0[%tbase];	1967-2022	06-03-2024	io_randtotaler2.gcm line 99

```
Name: qBNP
Code: qBNP = vBNP/pBNP;
Period: 1967-2022, Active: 1967-2022
Stamp: 06-03-2024 13:55:27, #2158439434972815411
File: H:\Ny\Git\Datop\MAKRO\Data\Makrobk\Progs\Iodata\io_randtotaler2.gcm line 100
Vars: vBNP, pBNP
```

Apart from that, the largest development point in this period has been the data tracing module, released officially in March 2024 (cf. the data trace window above). Data tracing traces interlinked series statements back in time, kind of like DECOMP for data construction. Traces work through Gekko databanks too and make it possible to quickly get an idea of how particular numerical values are arrived at, including timestamps, .gcm file + line numbers etc. Traces link up and form a so-called directed acyclical graph rather than a tree structure, complicating the underlying Gekko source code quite a bit. But the data tracing module seems to be stable now, and traces do not seem to affect the running speeds much (they do make the .gbk databanks larger, though).

Program fencing has been implemented: fencing restricts Gekko to read/write files from designated folders, in a whitelist/blacklist fashion. Together with the root() function, this makes cloning or moving a system of .gcm files from one folder to another easier and more worry-free.

Gekko warnings have been improved, so that not too many similar warnings will flood the Gekko output window while running Gekko programs. Instead, the number of similar warnings are limited regarding the output window, but pooled together and shown at the end of a Gekko session (clickable links). See the possibilities in the user manual, under OPTION (“option global warnings …”).

INTERPOLATE now has the Denton method built in, making it possible to convert from for instance annual to quarterly data. However, Denton-Cholette or similar might be advisable regarding levels. Also, the laspchain() function has been improved and now also works for quarterly data (using so-called annual overlap), in addition to supporting d-prices (values in previous period’s prices). The hope is that INTERPOLATE and laspchain() can solidify into kinds of standards regarding these things. This is still work in progress.

[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.

There is a new `gekko version` … statement, designating which Gekko versions are intended for a particular .gcm file. Especially helpful for file structures containing a mix of Gekko 2 and 3 programs.

The Gekko editor has a long-term love-hate relationship with GAMS scalar models (“unfolded” GAMS equations), but it seems that the problem with exogenous variables and the so-called holdFixed option in GAMS has found a solution. GAMS scalar models are useful for decomposition, residual checks, and more.

The MAKRO data revision programs have been running in Gekko for about a year now, and these programs are a good test of Gekko array-timeseries in general, and all the logic surrounding these (like $-conditions etc.). A number of smaller Gekko enhancements were implemented due to these MAKRO programs (see the Gekko changelog for details on smaller enhancements).

Local options: as a new possibility, any global option may now also be put into the <…> field for “local” use.

A Gekko 3.2 version is imminent. Leading up to 3.2, care has been taken to ensure that Gekko 3.2 is fully compatible with Gekko 3.0/3.1.x (the most critical parts of Gekko have not undergone any heavy surgery since Gekko 3.0, like for instance the parser etc.). So 3.2 should hopefully be very stable and can be released as soon as the help system has been read through and refreshed.

Regarding Gekko 2.4, fencing, root() and program dependency tracking have been back-ported.

Gekko risk assessment

See separate paper on this (in Danish), to be discussed under agenda item 9. The paper describes how to mitigate risks regarding Gekko as a well-running system towards 2030.