IFC-Bank of Italy Workshop on "Data science in central banking: enhancing the access to and sharing of data"

17-19 October 2023

Open-sourced central bank macroeconomic models[1]

Douglas Araujo,
Bank for International Settlements

[1] This contribution was prepared for the workshop. The views expressed are those of the authors and do not necessarily reflect the views of the Bank of Italy, the BIS, the IFC or the other central banks and institutions represented at the event.

Open-sourced central bank macroeconomic models

Douglas Araujo (douglas.araujo@bis.org)[1]

Abstract

Central banks and other financial policymakers rely on macroeconomic models to understand transmission channels of policy decisions, forecast the economy under different scenarios, and inform policy stances in a forward-looking way. Over time, some central banks have made the source code of their macroeconomic models publicly available, an interesting development in the decades-long increase in levels of central bank transparency. The advantages of open sourcing model code include enhanced predictability, accountability, streamlined collaboration eg with other central banks, improvements to the quality of the models and raised awareness about models. This paper proposes a framework to benchmark how macroeconomic models are open sourced according to criteria such as code accessibility, level of documentation, existence of version or vintage control, and openness to comments or proposed technical enhancements by the general public. Sharing source code of models appears to be a trend that is likely to pick up over time, as modern software engineering practices and higher levels of awareness of the advantages of open source software permeate more central banks' practices. However, choices on whether and how these models are disclosed may depend on availability of resources and objectives of the central bank. The paper concludes with practical, cost-mindful suggestions for central banks considering opening the codes of their macroeconomic models or improving their current levels of disclosure. Open-sourced macroeconomic models can be the next frontier in central bank communication.

Keywords: macroeconomic models, monetary policy, open source software, central bank communication.

JEL classification: E5, E60, C87.

1. Introduction

Central banks and other authorities have relied for decades on macroeconomic models to provide useful insights for decision-making. These models help to understand transmission channels of policy decisions, forecast the economy under different scenarios, and inform policy stances in a forward-looking way. But *model transparency*, ie the extent to which information about the models themselves is public, varies considerably across central banks, echoing the variation of transparency stances more broadly (Eijffinger and Geraats, 2006). This paper examines practices around one dimension of model transparency by central banks, namely whether and how the software codes for executing these models are publicly available. While the focus of this paper is on central banks and their models, which are mostly used for monetary policy purposes, other financial authorities are also in scope and will also feature in the analysis.

But why would code sharing in particular be a form of transparency? Central banks often publish academic papers describing their models; shouldn't that provide enough insight? A thorough description

[1] Economist, Bank for International Settlements. The author thanks Rodrigo Barbone Gonzalez, John Barrdear, Leon Berkelmans, Ben Cohen, Deniz Igan, Lauri Jantunen, Aaron Mehrotra, Benoît Mojon, Robert Munteanu, Rory Sarten, Olivier Sirello, Koji Takahashi, Ryland Thomas, Alexandre Tombini and seminar participants at the Irving Fisher Committee-Banca D'Italia's 3rd annual workshop on Data Science in Central Banking and at the Bank for International Settlements for helpful comments and suggestions. All errors are my own. This work does not necessarily represent the opinion of the Bank for International Settlements. This version is dated 11 March 2024.

of models' formulas and assumptions, as typically found in such papers, is indeed useful for interested practitioners, and an obvious requirement for model transparency. But publicly sharing the software code advances transparency even more. For one, it significantly lowers the bar for third-party users to reproduce results and more easily do scenario analysis and simulate different macroeconomic environments. Beyond that, opening model code also allows others to explore the model at a more fundamental level, testing alternative assumptions, formulas and parameter calibrations. All of the above would help spread understanding of how the model works, and could even lead to further improvements in macroeconomic modelling in the same way that much open source software benefits from user suggestions. These publicly available model codes are broadly taken as being "open source" for the purposes of this paper, although a more precise definition discussed below will help benchmark model code disclosure and inform suggestions for improving it.

Public sharing of macroeconomic model code is not an isolated phenomenon; far from it. Central banks are increasingly open sourcing some of their software in general (Araujo et al (2023c)), not just macroeconomic models. This trend can be broadly associated with the widespread internalisation of data science practices in central banking (Araujo et al (2023b)). Meanwhile, academia continues its longer-term movement towards code transparency as part of a sound and accountable scientific process. Also, key parts of macroeconomic modelling are often performed with open source software, such as Dynare (Juillard (1996)).[2] Taken together, open sourcing macroeconomic models is both a driver and a reflection of central banks modernising their macroeconomic modelling practice from these two complementary perspectives: the software or data science view (how they compute models, what are the data and computation requirements) and the academic or scientific view (how to reproduce or replicate models).

There are practical advantages to central banks that decide to open source model code. Beyond the transparency gains mentioned above, which accrue to society at large, authorities themselves benefit in several ways. First, model source code can promote enhanced predictability of policy, including from reputational gains as an accountable and transparent central bank, which can support policy credibility (Blinder (2000)). Collaboration with other central banks and international institutions is made easier by the dissemination of code. Improvements in model quality can even be obtained, for example, from macroeconomists – an audience that would be naturally interested in exploring model code – that offer suggestions to help the publishing central bank stay at the frontier of modelling practices. Other potential benefits include improved awareness about the models, leading to a clearer understanding of policy trade-offs by private sector economists, and a higher probability that newly-trained macroeconomists will already be familiar with the underpinnings of the main models.

Importantly, central banks use a suite of macroeconomic models as one input to inform policy, but not all models need to have their source code disclosed for central banks to start benefitting. As pointed out by Blanchard (2018) among others, different types of macroeconomic models are needed for policy and as a result, central banks typically have a portfolio of models.[3] Thus, gradually open sourcing one model at a time can help central banks do so in a strategic way. Once a model is open sourced, the key is really *how* they are made public. Imagine two extremes of model code disclosure: one central bank that simply shares the software with scant documentation in a hidden link in its website, and another one where the code is properly documented, easy to find and with clear channels to receive feedback or

[2] One tangible sign of the influence of Dynare in macroeconomic modelling is the large number of citations: one of its versions has more than 1000 citations as of drafting, with others summing up hundreds of references. Beyond that, the author's observation is that Dynare helped democratise macroeconomic modelling to generations of economists inside and outside of central banks.

[3] Sims (2002) describes how these models are used in the monetary policy process in a few major central banks. Post-Global Financial Crisis overviews of the evolution of macroeconomic models include Pescatori and Zaman (2011), Dou et al (2020) as well as Blanchard (2018), Hendry and Muellbauer (2018) and other papers in the same issue of the *Oxford Review of Economic Policy*. More recently, models exploring agent heterogeneity are increasingly being explored (Costa (2022)).

suggestions from any interested user. It is easy to see that the latter central bank would reap more benefits from code open sourcing. More broadly, how the code is shared influences the extent to which central banks and third-party users benefit, as well as the former's overhead and maintenance costs. A contribution of this paper is precisely to assess current model code disclosure practices, and identify the main actionable low-hanging fruits for central banks to better position themselves in this respect.[4]

But while model code disclosure can benefit authorities, in some cases a particular set of risks could be relevant. One concern is that publicly available models might crowd out private sector forecasts, or unwarrantedly reveal insights into monetary policy that interferes with policymaking. Experience in the US and other countries that have been sharing model code for some time mitigate these concerns. Another risk is that open sourcing model code could open up the central bank to technical criticism of its models, although that problem can be tackled with appropriate investment in model code quality and documentation, which is already to be expected in any case. Ultimately, model sharing might also require clear communication on the fact that particular models can have their results adjusted manually in the process of producing forecasts for monetary policy discussions. Even when these manual adjustments are considered sensitive information and therefore should not be shared, it would still be beneficial to publish its source code of the main model itself if it is not sensitive information, especially if it mechanically reflects economic equations already described in published papers.

This work rests on insights from two different literatures. The first stream, from the industrial organisation and management fields, focuses on open source software as community-based innovation (Lerner and Tirole (2002, 2005b)). This literature discusses strategic openness and open innovation by firms (West (2003), West and Gallagher (2006), Borges et al (2016)) and the motivations and drivers for voluntary participation in open source projects by third-party software developers (Shah (2006), von Krogh et al (2012)). This substantive body of prior work seeks to understand how entities – usually the private sector – decide to create or contribute to open source projects, and how software developers engage in these projects when they are often doing this as a public good (Johnson (2002)) without immediate economic remuneration. While that literature concentrates on analysing the incentives of firms such as software vendors and of private developers to distribute, use and contribute to open source software, the current paper is interested in central banks (and other financial authorities). Also in contrast to most of that literature, this paper is observational, not theoretical.

The other stream of previous work focuses on central banking policymaking, with particular attention to transparency and to macroeconomic modelling practice. Eijffinger and Gerats (2006) and Dincer and Eichengreen (most recently in 2013) show that central banks are steadily more transparent, a movement in tandem with their independence. Interestingly, already back in 2006 Eijffinger and Gerats highlight model code as a more advanced form of transparency when discussing the economic type of transparency (ie, the data, models and forecasts feeding into the policy function): "Only two central banks attain the maximum score of 3 on economic transparency, the Reserve Bank of New Zealand and the Bank of England. *The latter deserves special mention; it provides extensive documentation on its economic models, including the computer code for its macroeconometric model*" (my emphasis). Mishkin (2004) discusses how well-calibrated policy transparency can help central banks achieve their goals during crisis events. This theme has also been the subject of international guidance provided by the International Monetary Fund (IMF) to countries that volunteer for this assessment; see IMF (2020, 2023) for the most recent version of the Central Bank Transparency (CBT) code. This paper adds to this literature by concluding that open sourcing macroeconomic model code can be seen as the next frontier in central bank communication.

The next sections elaborate on the points above. Section 2 introduces concepts around code sharing that are important for the rest of the analysis. Section 3 places the topic of this paper in broader

[4] Similar exercises in central banking disclosure practice include Fracasso et al (2003), who study inflation reports, and the analyses of central banks' financial stability reports by Muñoz et al (2012).

context, including other software codes shared by central banks and how the discussion is evolving in academia. Section 4 elaborates the opportunities and challenges of model code disclosure mentioned in the introduction. Section 5 proposes criteria to benchmark code sharing of macroeconomic models by central banks and other authorities, and Section 6 presents the main results, with Section 7 assessing current central bank practices in light of those criteria. Section 8 draws on these examples to offer practical advice on how to improve the level of code-sharing for those central banks that wish to do so, and the final section concludes. Importantly, this paper does not judge or evaluate the models themselves, and neither does it compare models or their outputs with each other. The present exercise compiles publicly-available information from the macroeconomic models published by some central banks and assess the extent to which information is available and the model is easy to replicate.

2. Defining open source

Arguably the most intuitive definition of what constitutes "open source code" is very simple: code that is publicly available.[5] In the case of macroeconomic models, this would be a text file containing instructions for the computer to important any necessary data when applicable, estimate the model equations, and produce some form of output, which can be analytical (such as impulse response functions or variance decompositions) or quantitative, such as forecasts. As for the instructions themselves, they are made in one more programming languages, defined as a specific set of commands that is both readable by expert humans and understood by modelling software such as Dynare (Juillard (1996)), more generally-applicable scripting languages (such as Julia and Python) or written in compiled languages such as C++ or Rust.

A more complete and useful definition of open source code is provided by the Open Source Initiative (2023), encompassing a range of tenets. The most important points are that the code should be freely shared and distributed, in a way that allows for modifications and derived works. Access to source code should not discriminate against persons or groups, and the software should not be restricted to only certain uses (eg, only academic or policy applications). The source code license apply equally to all users, without specific carve-outs or other ad hoc instructions for particular users. These and other terms apply to promote a broad and fair access to the source code when it is considered "open". In other words, an open model code would enable a third party outside the central bank or agency in question to run their own local copy of the macroeconomic model, without having to request access or inquire for the code and wait a response. To be sure, open sourcing does not grant access to external parties to the central bank's internal systems. The public version of the code itself does not even need to be stored in the central bank servers: there are sites dedicated to hosting other entities' codes.

As noted above, whether a particular software code can be considered open source depends on the terms of its *license*. Software licenses are legal documents that describe the permissible uses, modifications or redistributions of the code. Typically, in the absence of a license these actions by the end-user would normally constitute copyright infringement, but software creators, in this case the central bank, can use licenses to explicitly carve them out – with or without conditions. While there are multiple types of standardised licenses, and software creators are free to create their own, open source is usually taken as a synonym for free software in the sense that its price is zero.

Open source software across all domains has seen widespread growth across countries in the number of unique programmers and the quantity of software code worked (Graph 1). This provides a favourable background for central banks willing to share model code, because the increasing number of

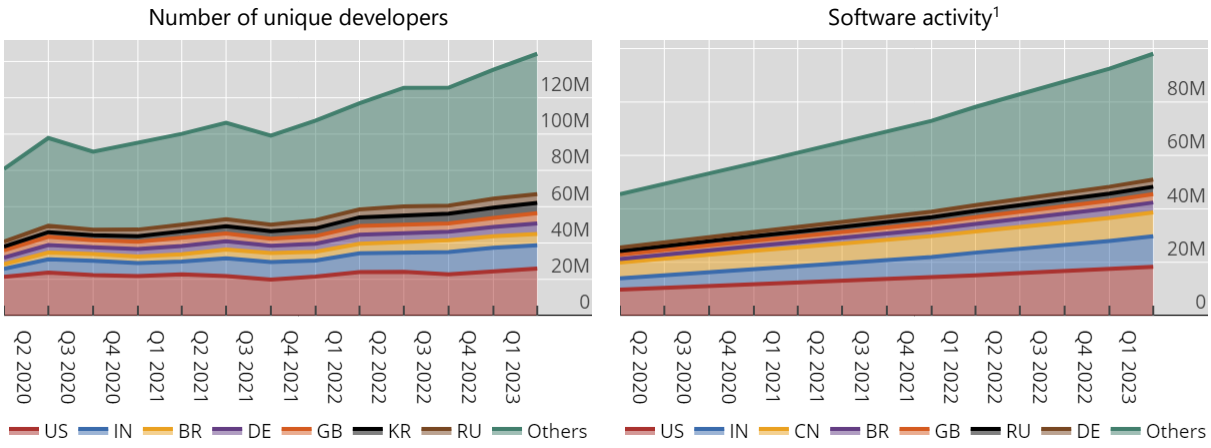
[5] A long-running quip in the computer science lore states in contrast that "all software code is open if you know Assembly", the low-level computer instruction language to which programmes are usually compiled.

people that is familiar with open source suggests that macroeconomists would also be increasingly able to engage with open sourced macroeconomic model code.

Growth in open source software activity, by location

Top 7 jurisdictions according to most recent period; all others aggregated in single series

Graph 1



Information as of 24 November 2023. [1] Software activity is defined as the number of "git pushes", which roughly corresponding to saving in a central repository one or more edited code files (ie, containing added, removed or changed code).

Source: GitHub Innovation Graph Metrics.

3. Broader context of open source and central banking

3.1 Central bank open source software

Many if not all central banks are users of multiple open source software for key IT and data science tasks, and some are also themselves sharing their own code (Araujo et al (2023)). Interestingly, the type of code being shared is heterogenous, as can be seen in the non-exhaustive list of central bank open source repositories found in Table 1.

Central bank public code repositories by type

Best efforts list of official central bank repositories[1]

Table 2

Institution	Internal utilities	Payments, CBDC, open banking	Research replication	Macroeconomic models
Central Bank of Brazil		X		
National Bank of Belgium	X			
Bank of Canada			X	X
Central Bank of Colombia			X	
National Bank of Denmark			X	

European Central Bank				X	X
Bank of France	X				
Bank of England	X			X	
National Bank of Greece			X		
Bank of Italy	X			X	X
Dutch Central Bank	X				
Bank of Norway			X		
Federal Reserve Bank of Boston (collaboration with MIT)			X		
Federal Reserve Bank of New York				X	X

Information as of 22 September 2023. [1] The helpful list of central bank open source repositories kept by the Dutch Central Bank (DNB (2023)) is kindly acknowledged.

Source: Author compilation, authorities' websites.

Repositories that contain internal utilities include formatting and styling software or data cleaning and wrangling tools. The latter in particular can be more easily used by other central banks, which often share similar data science use cases (Araujo et al (2023b)). The repositories including code related to payments, CBDC and open banking feature both experimental and production-level code. An example of the former is the Bank of Norway's sandbox for CBDC smart contracts, while the latter is illustrated by the application programming interface (API) for the Brazilian retail fast payment system Pix (Duarte et al (2022)). The research replication repositories are usually associated with specific publications. And the macroeconomic model repositories are part of the sample studied in this paper; other central banks make model code in other ways rather than in code repositories.

Similar to the repositories above, BIS work in this area also illustrates the more intense role of central banks as providers of open source software. In particular, BIS OpenTech, sdmx.io and the BIS Innovation Hub's Project Ellipse[6] (in partnership with the Monetary Authority of Singapore) all promote sharing of source code, either within the central banking community or with the broader public. Other BIS research work is open source, eg the gingado machine learing library (Araujo (2023)).

3.2 Disclosure of model code in the context of broader academic transparency

Beyond central bank transparency, disclosure of model code is also in line with broader developments in academic transparency. A number of widely-read academic journals in economics enforce some form of transparency and replicability. The definition used in this work follows the National Academies of Sciences, Engineering, and Medicine (2019): *reproducibility* is the ability to obtain consistent results with the same input data, computational steps and conditions of analysis, while *replicability* is the ability to obtain consistent results across studies that look at the same scientific question but each with its own data.

Vilhuber (2020) offers a comprehensive historical review of the evolution of code and data transparency in academic journals, most notably through the early imposition of "data availability policies". One notable evolution in this direction is the creation in December 2022 of the Data and Code Availability Standard (Koren et al (2022)), or DCAS, which defines criteria for availability of data, code, supporting materials and sharing. The DCAS was originally created by data editors of the American Economic

[6] https://github.com/bis-ih-ellipse-dkp.

Association[7], the Royal Economic Society[8], the *Review of Economic Studies*, the *Canadian Journal of Economics* and *Economic Inquiry* and later adopted by the Econometric Society[9]. The latter is also an example of a newly established (since July 2023) role of a Data Editor tasked with enforcing submission of complete reproduction packages, that should be well documented for future use. Interestingly, they go beyond the DCAS in some areas, such as calling for data formats to be readable also in non-proprietary formats.

In other words, open sourcing model code supports central banks alignment with best practices, promoting both reproducibility and replicability. In addition, this practice can help central bank research related to macroeconomic modelling reach wider impact due to alignment with leading journal practices.

3.3 Other macroeconomic model code sharing initiatives

Beyond central banks and government agencies, there are other academic-led compilations of macroeconomic model code that are worth mentioning. These resources are worth consulting for their wealth of model code, and because they can serve as starting material or benchmark models for central banks that are considering to leverage openly available code to add to their model portfolios.

The Macroeconomic Model Data Base (MMB), a product from the Macroeconomic Model Comparison Initiative (MMCI) (Taylor and Wieland (2012)) is a system that combines multiple models in a way that enables a comparison of their results. Interestingly, the website also includes guidelines and tools for authors that want to contribute models and a forum for discussion – both tools that help build a community around the models. Pfeifer (2023) and the Macrosimulation webpage (Prante and Kohler (2024)) also offer compilations of relevant macroeconomic model codes, gathering various papers and their respective Dynare code in the former case and R and Python codes in the latter collection. And because it is hosted in a public code repository system that includes a page for users to describe issues and discuss potential solutions, users can interact with one another and with the repository maintainer. Similarly, Cesa-Bianchi's (2024) webpage contain replications of landmark papers.

4. Opportunities from public disclosure of models

The advantages of open sourcing model code include enhanced predictability; more streamlined collaboration with other central banks and with international finance institutions; potential improvements to the quality of the models; and raised awareness about models' characteristics and limitations. As discussed in more detail below, the multiple ways in which central banks can opt to share code certainly affect the extent to which these opportunities are accrued. But as a community of macroeconomists start to form around a collection of open-sourced models, these opportunities can help shape research in the area and even facilitate central bank economists to incorporate any relevant crowd-sourced marginal improvements to their own models.

Making the model source code available significantly adds to model transparency, which can clarify communication and with this, improve the predictability of monetary policy, as observed by Blinder et al (2008). For example, Fracasso et al (2003) show that the quality of inflation reports lowers the

[7] Publisher of the *American Economic Review*, *AER: Insights*, *American Economic Journal: Applied Economics*, *Economic Policy*, *Macoreconomics*, and *Microeconomics*.

[8] Publisher of the *Econometrics Journal* and the *Economic Journal*.

[9] Publisher of the journals *Econometrica*, *Quantitative Economics* and *Theoretical Economics*.

uncertainty of private agents. It can only follow that sharing model code would further contribute to policy predictability, as it lowers the costs for reproduction and replication. The magnitude of the effects are less clear ex ante, as they probably depend on the starting point: a central bank that is already very transparent with respect to its models' characteristics and outputs, and how they feed into the policy process, might not reap much additional gains in predictability. Still, open sourcing models contribute to reinforce an appropriate level of disclosure and help maintain predictability, including from enhanced credibility (Blinder (2000)) that can arise from the enhanced level of transparency that is to share model code.

Model code disclosure can also directly facilitate collaboration with other central banks and international organisations. A simple use case is model comparison. Smets (1995) compared the monetary policy transmission mechanism across macro models of different central banks. If central banks as a rule made their models available, a broader range of stakeholders could compare models. Collaboration can also benefit the process of calibrating models, ie manually setting the values for some of the structural parameters in a model instead of estimating them from data. Calibration is usually done when a parameter is already well established in the literature, or when it is challenging to estimate, which calls for a better understanding of the effects of these parameters on the data-driven estimates (eg, Iskrev (2019)). For some central banks, it might also be challenging to estimate a structural parameter *de novo* due to shorter data histories, or regime and structural changes. Model code sharing can alleviate these challenges by significantly lower the bar for central bank experimentation with other policy models.

Beyond central banks, allowing third-party users to examine the model by open sourcing its code can lead to improvements in the quality of the model, both due to *ex ante* and *ex post* factors. The former comprise the strong incentives for a central bank that commits to publish model source code to invest resources before the publication to ensure that the code is of high quality and the model itself can withstand more detailed scrutiny. At the very least, the model's shortcomings would be communicated more explicitly, which is also a desirable outcome. As for the latter, model improvement can also occur once its code is published. As argued by Johnson (2006), the quality of an open source project can increase compared to a similar closed source initiative because proper peer review of codes are more prevalent in open codebases, and also because individuals share more freely ideas for improvement, given they will not necessarily be the ones to implement them. Del Negro et al (2015) make a similar point when justifying the rationale for open sourcing their Julia-based DSGE code: it is a form of validating, and improving, model code beyond in-house run tests of code accuracy. More broadly, once a sufficiently numerous pool of stakeholders becomes familiar with its inner workings, contributions by third parties in the form of suggested code modifications, testing routines (to make sure future versions of the model run as expected or "break" visibly), or improved documentation start to gain momentum. This is likely to occur especially if early contributions are constructively dealt with by central banks, either by accepting them (with or without requested revisions) or by clearly even if briefly explaining why the suggestion in question is not taken on board. [10]

But attracting active model contributors might be more challenging, at least at first while a culture of accepting third-party contributions is not widespread. An important obstacle is in the numbers: the set of potential contributors is a niche group of macroeconomists with sufficient knowledge of central bank models and of programming to know where these models can be improved. Given there is not much code in the open or easily accessible even if strictly speaking it is published, many macroeconomists might not

[10] The open source literature also identifies an incentive-related argument. In settings where economic agents compete with each other, the social benefits of open innovation could be associated with free-riding, reducing incentives for the innovator (in this case, the central bank), as highlighted by Schmidt and Schnitzer (2003) and Saint-Paul (2003). But in the case at hand, central banks will continue to use macroeconomic models as key ingredients of monetary policy, regardless of whether the source code is closed or open. Hence, the incentives for continuous improvement of models remain even when it is shared. In fact, the interests of the central bank and of society at large in the achievement of central bank mandates, such as control of inflation, are (or should be) aligned, so opening model code should promote good faith contributions.

be familiar with the practice of central bank modelling (even if they are with the theory). On top of that, central banks lack the track record to accept specific inputs from third parties that do not flow directly from another central bank, international financial institution such as the IMF or the BIS, or consulting economists that are hired for that purpose. In the spirit of Lerner and Tirole (2005b), those limitations are offset by a potentially strong set of private incentives for potential contributors to provide high quality input, if they had the chance. Contributions are individually named and fully transparent, and are therefore seen and assessed by peer macroeconomists. The central bank responsible for the model code needs to approve each contribution, which provides a "seal of approval" to that particular piece of code. And in addition to the central bank in question, private contributions would be potentially seen by other central banks as well as by other interested stakeholders such as private sector economists (eg from private forecasters, bank economic research departments) or from academia, reinforcing the signal about that contributor's ability.

Another advantage from public model disclosure is that stakeholders, such as private sector economists, can have a more concrete understanding of how central banks use models to view the economy and monetary policy transmission mechanisms. This can improve the way that the private sector interprets and receives policy announcements, especially due to the ability to adapt and run their own analyses based on central bank models. Opening model codes can thus be a practical way to help central banks reap some of the expected benefits from disclosure of policy function or predicted policy paths (eg, Rudebusch and Williams (2008)) in a less likely way to be wrongly interpreted as a policy commitment. In particular, sharing model code can contribute de-mystify central bank models as all-knowing black boxes that lead policymaking, and instead promote the more realistic image of mathematical simplifications that offer insights that feed into a larger policy debate.

5. Challenges related to model code disclosure

Naturally, open sourcing model code can also have challenges and even risks. I classify them into four main types, along the economic, policy, operational and reputational perspectives.[11]

One plausible risk as an open source macroeconomic model becomes widely used by forecasters and other private sector economists is similar to the phenomenon modelled by Morris and Shin (2002): the (now publicly available) forecasts or interpretations of policy transmission channels resulting from a model disclosure may be welfare decreasing depending on how noisy they are compared to private information. More specifically, forecasts resulting from open source central bank models could lead private agents to rationally overreact to this new information, even in situations where these agents had more precise information about the economy. This overreaction would stem from the coordinating effect arising from the unique role of central banks, and to paraphrase Morris and Shin, could be even more detrimental if the forecast from the open source model was faulty in the first place. Ideally this can be addressed by appropriate clarity by the central bank on the fact that model forecasts and estimated transmission channels are usually a *starting point*, not a direct ingredient in policy decisions. In other words, central bankers do not mechanistically follow model results. A related coordination risk occurs if the central bank models become the standard way the private sector sees the economy – while this may well occur given a facilitated implementation of the official models, having the code at hand would also enable private agents to more easily experiment with different alternatives for calibration or even model architectures, thus promoting more variation in private forecasts.

[11] Beyond the potential drawbacks of code disclosure, one practical challenge could be related to contractual or other restrictions central banks might face in sharing data that would feed into these models. However, this can be overcome with synthetic data.

From a policy dimension, another risk of open sourcing model code is that it might undesirably reveal insights into central bank thinking that could interfere with the ability to conduct monetary policy. Of course, model outputs are important ingredients of policy decisions, but definitely not the only ones. These macroeconomic models are tools to help policymakers put together how they see the economy working from its various components. Still, in a way, this is related to the Lucas critique (1976): macroeconomic models often make assumptions about market reactions, which could themselves be shaped by model disclosures. However, this does not seem to have been the case in practice, even for open-source models of widely followed economies such as the US.

Operational challenges include the preparation of the code for disclosure in a way that is consistent with central banks' strong reputation in economic analysis, and ways to properly deal with feedback. Code disclosure requires central banks to invest upfront in ensuring that the code is complete, working, properly structured and ideally well documented. Granted, this investment would be ideal even for internal code, but the need to maintain a reputation for high quality analyses could reflect on an added investment if the code is made public.

Another potential concern is that publicly available model code could open up the model for criticism, potentially harming a central bank's technical reputation. While in the end modelling efforts would always benefit from critical input, especially from informed third-parties that were not involved in the original modelling effort, the tolerance for criticism might be different between central banks and over time. This is also related to how central banks react to spontaneous feedback received on the model code. As usual in open source projects of relevance, third-party developers occasionally suggest code improvements, point out bugs or simply ask technical questions related to reproducing the model. The need to respond can impose a burden for many central banks, especially if its jurisdiction counts with specific requirements in analogy with "freedom of information" requests. A remedy to this is to ensure the model is well documented *ex ante*, thus precluding any need for generic responses, while also engaging with technical suggestions in a way that balances staff availability at the central bank with the benefits from leveraging crowd-sourced talent.

6. Criteria

I use two groups of criteria in assessing central bank open source models. The first set of criteria are related to the transparency objective of sharing code: allowing others to see how the model is implemented, understand its various features and inputs, and be able to reproduce it. The second group of criteria go beyond just the transparency objective, helping build and promote a community of users of the code: practices around software choice and versioning, how easy it is to contribute with issues or code suggestions, and whether third party users have clarity on the permitted uses via an explicit license. The following subsections elaborate these criteria.

In this section, each criterion is written in bold, with specific questions to help judge compliance in italics. A normal-faced text follows with more detail, eg its rationale or how it relates to other criteria.

6.1 Core criteria for code transparency

The most important criteria to promote code transparency are related to open access (how easy it is to find and access the code), documentation, and how easy it is to reproduce the model.

Open access

- **Easy to find**: *is the code available in a location on the central bank's website that is easy for interested users to find, such as a dedicated repository or as accompanying material to the paper describing the model?* Learning about the existence of model source code is of primary importance for sharing. Because there isn't yet a culture of sharing code, it is reasonable to assume many potentially interest users of model code might not know it is available, let alone how to find it, unless it is easy to find.
- **Free access**: *can the source code be freely accessed and downloaded by anyone without the need to register with the central bank or ask for its permission?* Even if some aspects of code are not free to run (eg, use of commercial software), access to the code file itself may be free. Still, some central banks might conceivably want to restrict access to the model code in order to have a tighter control on its usage. Even in these cases, the central bank might find it more convenient and a better cost-benefit mix to post the code for free access while fine-tuning its license, as discussed below.

Documentation

- **Academic description**: *is an academic document (either published in a journal or as working paper or occasional paper) available with a full description of the model?* A detailed composition of the model is obviously a pre-requisite for proper model transparency. One common and useful way this can be done is by writing a detailed description of the model in academic format.
- **Application programming interface (API) documentation**: *are all user-facing methods, classes, variables and other programming objects adequately documented?* For models that are executed in environments based on graphical user interfaces (GUI), this criteria amounts only to pointing to the different necessary files, as the code execution itself occurs in the GUI.
- **Input data documentation**: *are the expected data formats well documented, such that a new user can know what data goes in, in which format, and whether any data transformations (seasonal adjustment, logarithm, de-meaning, etc) are expected?* Ideally, documentation includes a complete list of all necessary input data, and would inform users about sources of the data used to estimate the original model, preferably official sources such as central bank themselves or statistical agencies, or vendors when needed. Input data documentation needs to be very precise to avoid ambiguity during reproduction. For example, whether a particular series is seasonally adjusted or not. A very helpful practice related to this is including data that supports a minimal use case. Regardless of whether this data would be real or simulated (eg, to avoid breaching data distribution agreements that might fall on vendor data), offering users a minimally reproducible example not only advances documentation of the model itself but helps users to understand how it is run and to test it in case users make changes to the code.
- **Output documentation**: *are the formats of the model results well described, so that the user can form an expectation on the type of output to be expected?* In cases where the code offers reproduction for a published paper, the output is usually a table or graph and thus can be considered to be well-documented. When this is not the case, the model should be clear about what type of output can be expected.

Reproduction and replication

- **End-to-end execution**: *is it possible to run the model end-to-end with the available code and instructions?* This criteria does not require an exact reproduction of central bank *results*, as it may depend on data that is either available only administratively or on vendor data subject to

distribution limitations. Rather, this criteria points to an ability to *execute* the code beginning-to-end with only the distributed code and any other requirement obtainable through clearly documented instructions, such as data or specific hardware.

- **Reproduction or replication instructions, tutorial or vignette available**: *does the model include explicit instrucitons for reproduction of central bank results or replication with other data, from download to final result?* These documents are meant to guide users, including those that have never interacted with this code before. They include a narrative description, coupled with examples, of all steps necessary to execute the code. For central banks, as for other software creators in general, writing vignettes can be a way to improve the code, as it puts developers in the shoes of first-time users, a point made by Wickham and Bryan (2023).
- **Minimum requirements**: *are the minimum hardware and software requirements clearly stated?* Naturally, a full list of requirements is impractical, but at least a major version of the software language used to run the model is required (it is likely the main constraint in any case).

6.2 Additional criteria for community-based benefits

Software

The choice of programming language has implications not only for the performance of the macroeconomic models (Aruoba and Fernández-Villaverde (2015)), but also on the ability of third-party users to understand and execute the code. Of particular importance is the difference between software that is proprietary and accessible only at a cost compared to free software (usually itself open source).[12] Similarly, models that do not run across the most widely used operating systems (Windows, macOS and Linux) create roadblocks for an important subset of users.

- **Software open availability**: *is the programming language itself free and easy to acquire?* The software might not need to be open source, as long as it is accessible to a broad range of users.
- **Technology neutrality**: *can the model be run in a broad variety of operating systems, including Linux, macOS and Windows?* Some software used for econometric modelling may be executable across multiple platforms (especially macOS and Windows) even when the programming language itself is not free. Alternatively, even a free and open source programming language could in practice not be available for all platforms.
- **Testing**: *does the code contains tests that ensure that new code, including any adjustments made by users, can be tested against to ensure it runs?* Given that model codes are often complex in the sense that they take multiple input series, perform non-trivial mathematical operations on these series, and then generate some output, including testing routines in the code would ensure that all these steps occur as appropriate. This is particularly important as data manipulation can suffer from silent errors, ie those situations where the software executes as expected (ie, the code "does not break") but because of formatting or similar issues the data was not in the expected format, leading the code to execute in a completely different way as intended.
- **Explicit code versioning**: *is the version of the code explicitly stated?* Users reproducing or replicating the model need to be certain which version of the model they are working with, which is especially important as macroeconomic tools evolve and get incorporated into newer version of models.

[12] In some cases, like MatLab, there are suitable open source alternatives (Octave, in this case).

- **Availability of past versions**: *can users freely and easily access past versions of the model in the same website?* Access to versioned code helps users be certain of the model version that they are executing. This is especially important in the case of macroeconomic modelling code, since a number of central banks tend to update their models every few years to take stock of lessons learned from previous years, and potentially of advances in modelling.

Contribution

- **Contact with software maintaners**: *is some form of contacting a person or team responsible for the model software clearly available to all, eg in email format or in a discussion board?* While opening up a contact channel to the broad public can seem daunting for some central banks, it is one important way to ensure contributions come in without the central bank actively exerting effort. Naturally, this form of contact does not need to be an e-mail mailbox – which almost calls for a reply – but a forum or even a commenting field similar to how pictures are shared in various social networks. The point of this criterion is to encourage central banks to offer some effective way for the broader public to provide their written feedback to model software maintainers.
- **Third party contributions possible**: *can third party users independently propose changes to the code and documentation?* An evolution of the communication possibility is to allow third party users to literally propose improvements to the code or to the documentation.
- **Contribution guidelines**: *does the model have explicit guidelines for contributors, or a contributor agreement?* Of course, this is more valid for the codes that are shared in a way that offers opportunities for collaboration. In those cases, it is important to offer consistency between actions and the guidelines, as they often diverge (Elazhary et al (2019)).

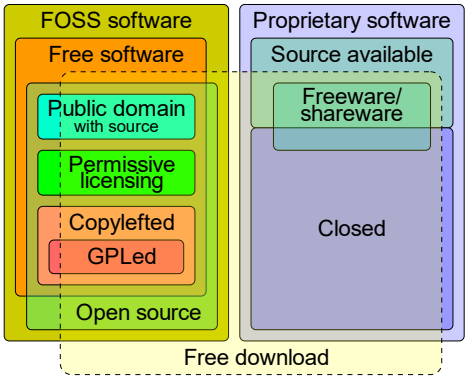
License

- **Explicit open source license**: *is the license explicit and is it compatible with the open source definitions?* Over the years, several standardised open source licenses were created with the purpose of setting different terms that are considered open source. These standard licenses are widely known, usually offer legal teams comfort around their meaning and implications in different circumstances, and thus can more easily be used off-the-shelf.[13] Still, some specific circumstances might call for a customised license. While they might not be as easily understood as the standard licenses, and thus should not be preferred as they impose a cost on each user to check the conditions, custom-made licenses can offer central banks reasonable flexibility in how they open source model code. In any case, it is important to understand different types of software and source code availability that licenses imply. Graph 2 overviews license concepts.

[13] One widely used catalogue of standard licenses is the Linux Foundation's SPDX, available at https://spdx.org/licenses/.

Software license concepts according to the Free Software Foundation

Graph 2



[1] FOSS: free and open source software. GPLed: software under the GNU General Public License (GPL).

Source: Wikipedia.

7. Current central bank practices

A best-efforts stock-take of central banks with open-sourced model code is presented in Table 2. The models listed were sourced mainly from the author's own familiarity with some of these codes; crowd-sourced contributions in a public repository, and from searching the most recent macroeconomic models by the central banks that already share other types of code, listed in Table 1.

Some observations are in order. First, not many countries populate this list: Table 2 is short compared to the population of central banks. This was not a comprehensive stock take, and there could be other central banks that do share code but were hard to find. Still, the models of some central banks that share other, non-model code (Table 1) are not open source. For example, the Central Bank of Brazil's SAMBA model (Fasolo et al (2023)) or the National Bank of Belgium's BEMGIE model (de Walque et al (2023)) have been recently overhauled but their codes are not publicly available. The Danish central bank's MONA model (2003) and the Bank of Norway's NEMO model (2019) also do not seem to have publicly available code. These models and others are well-documented, so the absence of public code might represent either an explicit decision to retain code confidentiality or simply an inertia from past practices. Second, once a country's central bank does decide to open source model code, then it is common for more than one model to be publicly shared. Case in point is the United States, perhaps the first country with a central bank to publish model code approximately one decade ago, has no less than four open source models. Together, these observations point to a possible effect at the jurisdiction level on the decision to share model code: perhaps culture, legal framework, or a combination of these and other factors prevent central banks from benefitting from opening their macroeconomic model code.

Other works in this space that are worth mentioning are the Bank of Canada's StateSpaceEcon.jl Julia package (St-Pierre (2023)) and the Bank of Italy's Python-based Black-it package (Benedetti et al (2022)). They are not macroeconomic models themselves, but well-documented open source libraries for coding such models.[14]

[14] Codes available respectively at github.com/bankofcanada/StateSpaceEcon.jl and github.com/bancaditalia/black-it.

Open-sourced central bank models used in the assessment

Best-efforts list of official models with available code

Table 3

Jurisdiction	Institution	Model name	Model ID[1]	Reference
Australia	Reserve Bank of Australia	MARTIN	AU-01	Ballantyne et al (2019)
Chile	Central Bank of Chile	MSEP	CL-01	Arroyo Marioli et al (2020), BCC (2020)
Chile	Central Bank of Chile	XMAS	CL-02	García et al (2019), García and Guerra-Salas (2020), BCC (2020)
Denmark	Ministry of Finance[2]	MAKRO	DK-01	Bonde et al (2023)
European Union	European Central Bank	BEAR	EU-01	Dieppe and van Roye (2023)
European Union	European Commission[2]	Output Gap Model	EU-02	
Finland	Bank of Finland	Aino	FI-01	Kilponen et al (2016)
France	Ministry of Economics and Finance[2]	Mésange	FR-01	Dufernez et al (2017)
France	Ministry of Economics and Finance[2]	Opale	FR-02	Daubaire et al (2017)
France	Ministry of Economics and Finance[2]	Modèle Saphir	FR-03	Amoureux et al (2018)
Germany	Federal Ministry for Economic Affairs and Climate Action[2]	-	DE-01	BMWK and BMFin (2022)
Iceland	Central Bank of Iceland	QMM	IS-01	Daníelson et al (2019)
Japan	Bank of Japan	Q-JEM	JP-01	Hirakata et al (2019)
Sweden	Riksbank	Ramses II	SE-01	Adolfson et al (2013)
United Kingdom	HM Treasury and Office for Budget Responsibility[2]	OBR model	GB-01	Office for Budget Responsibility (2013)
United States	Federal Reserve Bank of New York	DSGE.jl	US-01	
United States	Federal Reserve Bank of New York	Nowcasting	US-02	Bok et al (2017)
United States	Federal Reserve Board	FRB/US	US-03	Brayton et al (2014), Laforte (2018)
United States	Federal Reserve Board	EDO	US-04	Chung et al (2010)

Information as of 22 September 2023. [1] Model ID is used throughout the text to refer to each model. [2] Other official financial agencies such as Ministries of Finance are also in scope of this compilation.

Source: Author compilation, authorities' websites.

8. Assessment of open-sourced model code

I assessed each code against the criteria identified above (Table 3). When in doubt, I took a positive view of compliance with the criteria – for example, a simple documentation of the usage interface can be taken

as documentation of the API, even if rudimentary. This is because the purpose of this assessment is to take a broad view of the main practices in official macroeconomic model code disclosures. Models judged to fulfil the criteria are therefore also encouraged to continuously improve.

Characteristics of open-sourced macroeconomic model codes

Best efforts list of official models with available code

Table 4

	Core criteria									Additional criteria								
	Open access		Documentation				Reprod./Replic.			Software					Contribution			Lic.
Model ID[1]	Easy to find	Free access	Academic	API	Input data	Output data	End-to-end	Instructions	Requirements	Open software	Tech neutrality	Testing	Explicit version	Past versions	Contact	Third parties	Guidelines	Explicit license
AU-01	X	X	X	X		X	X	X	X									X
CL-01	X	X	X	X	X	X	X	X	X	X	X				X			
CL-02	X	X	X	X		X	X	X	X	X	X				X			
DE-01		X	X	X	X		X			X			X	X				
DK-01	X	X	X	X	X		X		X		X		X	X	X	X		X
EU-01	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X
EU-02	X	X		X	X		X		X				X	X				
FI-01	X	X	X	X	X	X	X		X	X	X							
FR-01	X	X	X	X	X	X	X	X	X				X	X	X	X		X
FR-02	X	X	X	X	X	X	X	X	X				X	X	X	X		X
FR-03	X	X				X	X	X	X				X	X	X	X		X
IS-01	X	X	X	X	X	X	X	X										X
JP-01	X	X	X	X	X	X	X	X	X						X			X
SE-01	X	X	X	X			X	X		X	X				X	X		
GB-01	X	X	X	X	X		X		X									
US-01	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
US-02	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X		X
US-03	X	X	X	X	X		X	X	X	X	X							
US-04	X	X	X	X			X			X	X							X

Information as of 22 September 2023.[1] See Table 2.

Source: Author compilation, authorities' websites.

8.1 Core criteria

Open access

The vast majority of model code can be classified as having open access. Having the code publicly available is used in this paper as a definition of open source, so by design all models comply with this criterium. In terms of the ease of access, DE-01 could be better.

Documentation

In some cases, models list the **input data** used to estimate the model including their sources. Other cases document input data, but dispersed in the academic documentation of the model instead of a single location that is easy to consult. In a few cases, authors embed the data in the distributed model. One helpful practice, observed for example in AU-01, is to distribute simulated data when the original series are under commercial protection, specifying which ones are simulated.

Reproduction and replication

Open source models are already being used in academia by third-party authors as a way to obtain credible counterfactuals for macroeconomic variables. Gross and Leigh (2022) use AU-01, re-estimated up to 2019 and adapted to use only publicly available data, to estimate counterfactuals for nominal cash rate, (trimmed-mean) inflation, real GDP, unemployment gap, wages and exchange rate. Blanchard and Summers (2020) use US-03 to simulate the US economy under a neutral fiscal policy.

8.2 Additional criteria

One general remark about the additional criteria is that model codes stored in dedicated repository websites already benefit from many useful characteristics in terms of open source code. In addition to making the code easy to find and access (core criteria), it automatically includes explicit versioning and availability of past. Moreover, such repositories also provide a useful platform for contact with code maintainers and other code users, and allow for suggestions from third parties to improve code and documentation. These code repositories also facilitate a clear definition of a use license and implementing testing routings (in the form of "continuous integration", a technique that ensures that every new version of the code must be able to execute some testing or production-like routines).

Software

The software choices across models are heterogenous, but with a clear tendency to gravitate to Matlab (which has an open source alternative, GNU Octave) and EViews. In terms of **software open availability**, many models run on software that is freely available. But in some cases central banks use commercial software that is not freely available, such as GAMS (General Algebraic Modelling System)[15] and EViews. Table 4 presents an overview of the software languages used by the models studied in this paper.

Most common programming languages for macroeconomic models

Best efforts list of official models with available code

Table 5

Languages	Open source	Multi-platform	Observations	Models
Matlab (or GNU Octave)	No (Yes)	Yes	Most models with Dynare	CL-01, CL-02, EU-01, SE-01 US-02, US-04

[15] Interestingly, and laudably, GAMS Software GmbH has an explicit and public open source strategy that acknowledges the value of open source to the firm's software, and also mentions that the company seeks to contribute code to open source software.

EViews	No	No		AU-01, GB-01, JP-01[2], US-03[2]
Portable TROLL	No	Yes (raw code; GUI[3] only available for Windows)		FR-01, FR-02
Julia	Yes	Yes		US-01[1]
SAS	No	Yes		FR-03
EUCAM	No	No		EU-02
GAP	No	No		DE-01
GAMS	No	No	APIs[3] available for Python, Java and other languages	DK-01

Information as of 22 September 2023. [1] Based on previous Matlab code. [2] Also has a Python version. [3] GUI: graphical user interface, API: application programming interface.

Source: Author compilation, authorities' websites.

Dynare (Juillard (1996)) appears to be the reason why most models use Matlab. It is a time-tested, well-documented open source software that estimates a wide range of models used by central banks. A whole community of macroeconomists, many of them active in central banks, participate in its forums and in the events organised by the Dynare community. There is even a working paper series dedicated to models developed with this software.

Two purpose-specific languages feature in the studied sample: Portable TROLL[16] and GAP. The former is a C-based software that is reportedly executable across multiple platforms, with a graphical user interface (GUI) in Windows. It is developed and marketed by Intex Solutions, Inc. Some European models (DE-01, EU-02) run on GAP, a programme developed by the European Commission based on the EU output gap methodology. While GAP itself is publicly accessible, it requires Microsoft Excel and MatLab, neither of which are free. And although both have free and open source alternatives, such as LibreOffice for the former and Octave for the latter, it is not clear whether GAP would execute on them.

Using GAP as an example, **technology neutrality** is not really present in all models. For example, the GAP model works only on Windows (since it is assumed or known to be the most widely professionally used OS by European agencies). Similarly, model codes that rely on EViews are only runnable on Windows. On the other side, many models are based on Julia, which is runnable on multiple OSs.

In general, **testing** is not included in the source codes. While model codes are not made for autonomous execution, instead serving the purpose of directly interacting with analysts and economists, the lack of embedded testing routines (such as unit testing) can be problematic because it does not include a key way to check whether code changes or improvements maintain an appropriately executing file. In other words, the model code is not guaranteed to work. As mentioned above, it is possible to set up testing in practical ways when the code is hosted in a repository website. For example, EU-01 and US-02 are tested automatically at every new code contribution that is pushed to the central repository.

For many open source models, **explicit code versioning** and **availability of past versions** (if any) is lacking. While such models are not expected to be updated at a high frequency, over time as third-party use of models increase, explicitly adequately version can be important. One interesting case is the European Central Bank's PERFORM system, which includes a git repository, a suite of software to run models, and schedulers that execute code testing and quality control (see Box 20 of Work Stream on Eurosystem Modelling (2021)).

[16] TROLL stands for Time-shared Reactive OnLine Laboratory.

Contribution

Central banks currently do not score well on **contribution guidelines**, and in fact, on anything that might harness one of the key advantages of having source code accessible by all: the power of contributions.

Unless the academic paper authors are considered to be the code maintainers, which is not always the case, there is often no way to contact the people responsible for model code. This is of course made worse by the fact that in many cases the model white paper is not even identifiably authored. In any case, even with a relevant contact, third parties willing to spend their time reporting bugs, suggesting code improvements, or simply asking questions about the code are often left without a clear, official way to contact maintainers.

Naturally, opening up such contact risks creating a wrong impression about availability of resources to follow up on these forms of contact mentioned above. But it does not have to be this way. The central bank might be upfront about third parties not expecting a response. Or alternatively, the central bank might either set up, or use an existing forum-like system where any third parties can post their comments. Many code repositories make such a forum available by default. Another advantage of these fora for contact between central bank staff code maintainers and third parties interested in the code is that all communication is public, documented, and identified.

License

Model **licenses** are an important area where some models can further develop. There are many standard licenses developed over time by the software community, but the software creator might wish to create a custom license. While describing details of the main open source licenses is outside the scope of this paper, one important message is that absence of a license does not mean that third parties can freely use it. On the contrary, lack of licenses formally precludes broader use of open source software because other people are not granted the permission to copy and adapt the code, for example. In some cases, a "disclaimer" instead of a license is available. When this disclaimer included at least some general notices about permitted usage or about any potential liabilities from the model, it was considered to be similar to a license.

The selection of a licensing model is a critical decision that can significantly impact how widely used the macroeconomic model is. The standard, 'off-the-shelf' licenses mentioned above offer a distinct advantage in this context, as they have undergone extensive scrutiny and acceptance within the software community. This pre-validated status of such licenses mitigates the legal and operational uncertainties typically associated with custom licensing agreements. While concerns about commercialising copies of macroeconomic models may be less pronounced in the context this paper explores, the strategic choice of a well-established open source license can still play a pivotal role. It can ensure legal clarity and broad acceptability, fostering a collaborative and transparent development environment that might attract third-party talent even from outside the central bank jurisdictions' borders. Correspondingly, Lerner and Tirole (2005a) show that the decision about the license both depends on, and influences, the community expected to be developed around the model code.

9. Enhancing availability of code

For those central banks that wish to put existing code in the public domain or to improve their current levels of accessibility of the code they make public, below are practical, cost-mindful suggestions:

- **Set up a code repository**. It could be self-hosted or, more easily, hosted in specialised servers such as GitHub, GitLab, etc. Another advantage is that they facilitate other auxiliary characteristics

of a good software repository, such as explicitly choosing a license and facilitating contribution by creation of issues or even code suggestions.

- **Create a page listing available code**. A central page in the authority's website could concentrate its available model code (and potentially other source code as well, such as for replicating papers). This would be easy to set up and relatively inexpensive to maintain, given that publication of new source code is not a frequent event. It can also help "cross-polinate" ideas, by helping users that are interested in one code find other code made available by the authority.
- **Create and share tutorials on how to replicate and adapt the models**. Ideally these tutorials would include steps from scratch, such as setting up the environment, accessing the necessary data, running the model, and when relevant changing parameters to those desired by the user.
- **Consider adapting code requiring proprietary software to open source solutions**. This would promote higher third party engagement, facilitate sharing with other central banks and external institutions, and even would facilitate new staff being onboarded to projects. Moving the code from one language to another (known as "transpiling") would be mostly an one-off investment. It could be partially or totally offset by the cost of giving up paid licenses, and could be simplified through the use of language artificial intelligence (AI) models, especially those focused in coding.

10. Conclusion

More than two decades ago, Blinder et al (2001) argued that central banks should better disclose how their models worked. Specifically, they suggested "well-chosen words supplemented by a few key numbers" would be more informative then "masses of equations" because the latter are far from transparent. This paper makes the case that model *code* can be even better, lists some central banks that are already doing this, analyses their practices and suggests specific actions to better harness this practice.

If model codes are publicly disclosed, comparisons of model responses such as Fisher et al (1988) and Smets (1995) can be more easily accomplished. This could enrich the literature on comparative analysis of monetary policy: for example, if replications of the models of two central banks respond the same way to similar shocks, then diverging rate decisions between them might suggest different monetary policy stances. At the same time, other views suggest that transparency might at some point be excessive if it interferes with policy effectiveness – along these lines, source code should be viewed, similar to broader policy transparency, as a means to an end (Mishkin, 2004). In any case, episodes like the 2007-09 Global Financial Crisis and more recently the post-Covid inflation forecast errors have prompted macroeconomics into collective soul-searching (Caballero (2010), Stiglitz (2011), Chahad et al (2022), Borio et al (2023), Koch and Noudeldin (2023)). Having a range of model codes in the open could facilitate diagnosing what is needed to prevent future errors, including by backtesting whether improved models would have been helpful ex ante.

Since macroeconomic models embody the academic work of a public institution, open sourcing of model code can also be seen as part of broader transparency and accountability movements in academia and in administration. The first one is "open science": the practice of sharing more openly code, data and even research plans. Open science acceptance and practice in economics has grown substantially over the last decade, following other social sciences (Ferguson et al, 2023). And the other is "open government", the idea that administration data should be made open and accessible to the extent possible (Attard et al, 2015). Part of the prominence of both of these "open" movements is due to confidence crises in the scientific and political worlds. In contrast, open sourcing models can help shield central banks from losses in their analytical capabilities. In addition, it can be seen as a way of transferring back to society

technologies funded with public money, as long as such disclosure does not interfere with the effectiveness of monetary policy.

This work is also related to the influence exercised by central bank forecasts on private agents (eg, Ehrmann et al (2012), Hubert (2015) and Pedersen (2015)). If the publication of central bank forecasts directly affects private agents regardless of performance, then it can be argued that sharing the code can lead to two effects: first, it would make the effect more diffuse since agents would be able to form their own views using the same model as the central bank. And second, whenever forecasts are indeed published, agents would be able to benchmark that against the hard output of the model, thus inferring the "human adjustment" component of the forecast. Whether this is beneficial or not is a matter to be explored in future research.

Public code sharing can be considered the next frontier in central bank communication, building on a multi-decade evolution (Blinder et al (2008)). Sharing model code provides exact, customisable implementations, which offers central banks the opportunity to be better listeners (Macklem and Vardy, (2023)) by standardising the language around its open sourced models and also observing how private sector users adapt these models. Model code also facilitates central banks' outreach to a broader audience, a more technical one, even if not the general public (Haldane and McMahon (2018)). As central banks consider what type of forecasts to provide to the public, with arguments in favour of publishing a range of forecasts based on different scenarios (Goodhart, 2023), open sourcing model code could allow users the possibility of running the models and testing their own scenarios. Model code sharing could also perhaps be seen as contributing to one of the pillars of central bank transparency as codified by the IMF (2020, 2023), namely the second pillar "*Transparency in policies*" through its sub-item "*policy decisions*".[17]

Central banks have been increasingly active open source code providers, and have been relying on open source tools for many of their most advanced data analytical tasks, such as machine learning (Araujo et al (2023a, 2023c)). Open sourcing macroeconomic model code in a way that maximises its benefits for central banks themselves and for societies would be another significant step in this direction. The diversity in model code disclosure practices suggests that there is still ample need for research to understand the drivers, implications and strategies on open sourcing macroeconomic model code.

References

Adolfson, M, S Laséen, L Christiano, M Trabandt and K Walentin (2013): "*Ramses II – model description*", Riksbank Occasional Paper Series No 12.

Amoureux, V, I Benoteau and A Naouas (2018): "*Le modèle de microsimulation Saphir*", DG Trésor Working Paper No 2018/06, September.

Araujo, DKG (2023): "*gingado: a machine learning library focused on economics and finance*", BIS Working paper No 1122.f

Araujo, DKG, G Bruno, J Marcucci, R Schmidt and B Tissot (2023a): "*Machine learning applications in central banking*", Journal of AI, Robotics & Workplace Automation, v.2, n.3, p.271-293.

Araujo, DKG, G Bruno, J Marcucci, R Schmidt and B Tissot (2023b): "*Data science in central banking: applications and tools*", IFC Bulletin No 59.

[17] In more detail, this would be related to subprinciple 2.1.3, ie transparency about supporting analysis for monetary policy.

Araujo, DKG, S Nikoloutsos, R Schmidt and O Sirello (2023c): "*Central banks as users and providers of open-source software*", Box in IFC Bulletin No 59.

Aruoba, SB, J Fernández-Villaverde (2015): "*A comparison of programming languages in macroeconomics*", Journal of Economic Dynamics and Control, v.58, September, p.265-273.

Arroyo Marioli, F, F Bullano, J Fornero and R Zúñiga (2020): "*Semi-structural forecasting model*", Banco Central de Chile Working Paper No 866, January.

Attard J, F Orlandi, S Scerri and S Auer (2015): "*A systematic review of open government data initiatives*", Government Information Quarterly, v.32, n.4, p.399-418.

Ballantyne A, T Cusbert, R Evans, R Guttmann, J Hambur, A Hamilton, E Kendall, R McCririck, G Nodari and D Rees (2019): "*MARTIN has its place: a macroeconometric model of the Australian economy*", Reserve Bank of Australia Research Discussion paper, RDP 2019-07.

Benedetti, M, G Catapano, F De Sclavis, M Favorito, A Glielmo, D Magnanimi and A Muci (2022): "*Black-it: a ready-to-use and easy-to-extend calibration kit for agent-based models*", Journal of Open Source Software, v.7, n.79, p.4622.

Bank of Norway (2019): "*A complete documentation of Norges Bank's policy model NEMO*", 22 March.

Blanchard, OJ (2018): "*On the future of macroeconomic models*", Oxford Review of Economic Policy, v.34, n.1-2, p.43-54.

Blanchard, OJ and LH Summers (2020): "*Automatic stabilizers in a low-rate environment*", American Economic Association Papers and Proceedings, v.110, p.125-130, May.

Blinder, AS (2000): "*Central-bank credibility: why do we care? How do we build it?*", American Economic Review, v.90, n.5, p.1421-1431.

Blinder, AS, C Goodhart, P Hildebrand, D Lipton and C Wyplosz (2001): "*How do central banks talk?*", Geneva Report on the World Economy 3, Centre for Economic Policy Research, London.

Blinder, AS, M Ehrmann, M Fratzscher, J De Haan and D-J Jansen (2008): "*Central bank communication and monetary policy: a survey of theory and evidence*", Journal of Economic Literature, v.46, n.4, p.910-945.

Bok, D, D Caratelli, D Giannone, AM Sbordone and A Tambalotti (2017): "*Macroeconomic nowcasting and forecasting with big data*"

Bonde M, J Ejarque, G Høegh, E Partsch, P Stephensen, T Vasi (2023): "*MAKRO model documentation*", paper available at www.dreamgruppen.dk.

Borio, C, B Hofmann and E Zakrajšek (2023). "*Does money growth help explain the recent inflation surge?*" BIS Bulletin no 67, Bank for International Settlements.

Brayton, F, T Laubach and D Reifschneider (2014): "*The FRB/US model: a tool for macroeconomic policy analysis*", FEDS Notes, April.

Bundesministerium für Wirtschaft und Klimaschutz – BMWK and Bundesministerium der Finanzen – BMFin, Germany (2022): "*Ergänzende Unterlagen zur Bestimmung des gesamtwirtschaftlichen Produktionspotenzials*", 26 January.

Caballero, RJ (2010): "*Macroeconomics after the crisis: time to deal with the pretense-of-knowledge syndrome*", Journal of Economic Perspectives, v.24, n.4, p.85-102, Fall.

Central Bank of Chile – BCC (2020): "*Uso de modelos macroeconómicos en el Banco Central de Chile*", paper available at bcentral.cl.

Cesa-Bianchi, A (2024): "*Replications*", available at: https://sites.google.com/site/ambropo/replications. Access on 11 March 2024.

Chahad, M, A Hofmann-Drahonsky, B Meunier, A Page and M Tirpák (2022): "*What explains recent errors in the inflation projections of Eurosystem and ECB staff?*" ECB Economic Bulletin no 3/2022, European Central Bank.

Chung, HT, MT Kiley, J-P Laforte (2010): "Documentation of the Estimated, Dynamic, Optimization-based (EDO) model of the U.S. economy: 2010 version", Federal Reserve Board Finance and Economics Discussion Series (FEDS) No 2010-29.

Costa, R (2022): "*Heterogeneity in macroeconomics – implications for monetary policy*", Remarks at XXV Annual Conference of the Central Bank of Chile, Santiago, 21-22 November.

Danmarks Nationalbank (2003): "*MONA – a quarterly model of the Danish economy*".

Daubaire, A, G Lefebvre and O Meslin (2017): "*La maquette de prévision Opale2017*", DG Trésor Working Paper No 2017/06, May.

Daníelson, Á, L Elíasson, MF Gudmundsson, SJ Haraldsdóttir, LS Kro, ThG Pétursson, ThS Sveisson (2019): "*QMM – a quarterly macroeconomic model of the Icelandic economy Version 4.0*", Central Bank of Iceland Working Paper No 82, December.

De Nederlandsche Bank (DNB) (2023): "*GitHub pages of institutions*", available at: https://github.com/DeNederlandscheBank/general-information/blob/main/github_pages_institutions.md

De Walque, G, T Lejeune, A Rannenberg and R Wouters (2023): "*BEMGIE: Belgian Economy in a Macro General and International Equilibrium Model*", National Bank of Belgium Working Paper Research, no 435, March.

Del Negro, M, M Giannoni, P Li, E Moszkowski and M Smith (2015): "*The FRBNY DSGE model meets Julia*", Liberty Street Economics, 3 December.

Dieppe, A and B van Roye (2023): "*The BEAR toolbox*", website available at: www.ecb.europa.eu/pub/research/working-papers/html/bear-toolbox.en.html. Accessed on 22 Sept 2023.

Dincer, N and B Eichengreen (2014): "*Central bank transparency and independence: updates and new measures*", International Journal of Central Banking, v.10, n.1, p.189-253.

Dou, WW, AW Lo, A Muley and H Uhlig (2020): "*Macroeconomic models for monetary policy: a critical review from a financial perspective*", Annual Review of Financial Economics, v.12, p.95-140, November.

Duarte, A, J Frost, L Gambacorta, P Koo Wilkens and HS Shin (2022): "Central banks, the monetary system and public payment infrastructures: lessons from Brazil's Pix", BIS Bulletin No 52, 23 March.

Ehrmann M, SCW Eijffinger and M Fratzscher (2012): "*The role of central bank transparency for guiding private sector forecasts*", The Scandinavian Journal of Economics, v.114, n.3, p.1018-1052.

Eijffinger, SCW and PM Geraats (2006): "*How transparent are central banks?*", European Journal of Political Economy, v.22, p.1-21.

Elazhary, O and M-A Storey, N Ernst, A Zaidman (2019): "*Do as I do, not as I say: do contribution guidelines match the GitHub contribution process?*", arXiv, mimeo.

Fasolo, AM, E Araujo, M Valli Jorge, A Kornelius and LSG Marinho (2023): "*Brazilian macroeconomic dynamics redux: shocks, frictions, and unemployment in SAMBA model*", Latin American Journal of Central Banking, in press.

Ferguson, J, R Littman, G Christensen, EL Paluck, N Swanson, Z Wang, D Birke and J-H Pezzuto (2023): "*Survey of open science practices and attitudes in the social sciences*", Nature Communications, v.14, n.5401.

Fisher, PG, SK Tanna, DS Turner, KF Wallis and JD Whitley (1988): "Comparative properties of models of the UK economy", National Institute Economic Review, v.125, p.69-87, August.

Fracasso, A, H Genberg and C Wyplosz (2003): "*How do central banks write? An evaluation of inflation reports by inflation targeting central banks*", Geneva: International Center for Monetary and Banking Studies; London: Centre for Economic Policy Research.

García, B, S Guarda, M Kirchner, R Tranamil (2019): "*XMas: an extended model for analysis and simulation*", Banco Central de Chile Working Paper No 833, May.

García, B, J Guerra-Salas (2020): "*On the response of inflation and monetary policy to an immigration shock*", Banco Central de Chile Working Paper No 872, April.

Goodhart, C (2023): "Should central banks abandon single point forecasts?", VoxEu, 29 August.

Gross, I and A Leigh (2022): "*Assessing Australian monetary policy in the twenty-first century*", Economic Record, v.98, n.322, p.271-295, September.

Haldane, A and M McMahon (2018): "*Central bank communications and the general public*", AEA Papers and Proceedings, v.108, p.578-583.

Hendry, DF and JNJ Muellbauer (2018): "*The future of macroeconomics: macro theory and models at the Bank of England*", Oxford Review of Economic Policy, v.34, n.1-2, p.287-328

Hirakata, N, K Kan, A Kanafuji, Y Kido, Y Kishaba, T Murakoshi and T Shinohara (2019): "*The Quarterly Japanese Economic Model (Q-JEM): 2019 version*", Bank of Japan Working Paper Series, No 19-E-7, June.

Hubert, P (2015): "*Do central bank forecasts influence private agents? Forecasting performance versus signals*", Journal of Money, Credit and Banking, v.47, n.4, p.771-789.

International Monetary Fund (2020): "*The central bank transparency code*", IMF Policy Paper 2020/038, 4 July.

International Monetary Fund (2023): "*Central bank transparency code review – guidance note*", IMF Policy Paper 2023/050, 31 October 2023.

Iskrev, N (2019): "*What to expect when you're calibrating: measuring the effect of calibration on the estimation of macroeconomic models*", Journal of Economic Dynamics and Control, v.99, p.54-81.

Johnson, JP (2004): "*Open source software: private provision of a public good*", Journal of Economics & Management Strategy, v.11, n.4, p.637-662.

Johnson, JP (2006): "*Collaboration, peer review and open source software*", Information Economics and Policy, v.18, n.4, p.477-497.

Juillard, M (1996): "*Dynare: a program for the resolution and simulation of dynamic models with forward variables through the use of a relaxation algorithm*", CEPREMAP Working Paper No 9602.

Koch, C and D Noureldin (2023): "*How we missed the inflation surge: an anatomy of post-2020 inflation forecast errors*", IMF Working Paper no 2023/102.

Koren, M, M Connolly, J Llull and L Vilhuber (2022): "*Data and Code Availability Standard [Version 1.0]*", Website, https://datacodestandard.org https://doi.org/10.5281/zenodo.7436134.

Laforte, J-P (2018): "*Overview of the changes to the FRB/US model*", FEDS Notes, December.

Lerner, J and J Tirole (2002): "*Some simple economics of open source*", The Journal of Industrial Economics, v.50, n.2, p.197-234.

Lerner, J and J Tirole (2005a): "*The scope of open source licensing*", The Journal of Law, Economics, and Organization, v.21, n.1, p.20-56, April.

Lerner, J and J Tirole (2005b): "*The economics of technology sharing: open source and beyond*", Journal of Economic Perspectives, v.19, n.2, p.99-120, Spring.

Lucas, R (1976): "*Econometric policy evaluation: a critique*", in: Carnegie-Rochester Conference Series on Public Policy, v.1, 19-46.

Macklem, T and J Vardy (2023): "*20 years of central bank communications, and lessons for the future*", in: Central banking in the Americas: Lessons from two decades, BIS Papers No 143.

Morris, S and HS Shin (2002): "*Social value of public information*", American Economic Review, v.92, n.5, p.1521-1534.

Mishkin, FS (2004): "*Can central bank transparency go too far?*", NBER Working Paper No 10829.

Muñoz, S, S Jahjah, M Cihak, ST Sharifuddin, KI Tintchev (2012): "*Financial stability reports: what are they good for?*", IMF Working Paper WP/12/1.

National Academies of Sciences, Engineering, and Medicine (2019): "*Reproducibility and replicability in science*", Washington, DC: The National Academies Press. https://doi.org/10.17226/25303.

Office for Budget Responsibility, United Kingdom (2013): "*The macroeconomic model*", Briefing paper No 5, October.

Open Source Initiative (2023): "*Open source definition*", available at: https://opensource.org/osd/

Pedersen, M (2015): "*What affects the predictions of private forecasters? The role of central bank forecasts in Chile*", International Journal of Forecasting, v.31, n.4, p.1043-1055.

Pescatori, A and S Zaman (2011): "*Macroeconomic models, forecasting and policymaking*", Federal Reserve Bank of Cleveland Economic Comentary No 2011-19, October.

Pfeifer, J (2023): "*DSGE_mod: a collection of Dynare models*", computer software, available at: https://github.com/JohannesPfeifer/DSGE_mod

Prante, F and K Kohler (2024): "*Macrosimulation*", available at: https://macrosimulation.org. Accessed on 11 march 2024.

Rudebusch, GD and JC Williams (2008): "*Revealing the secrets of the temple: the value of publishing central bank interest rate projections*", In: Campbell, JY (Ed.), *Asset Prices and Monetary Policy*, p.247-289.

Saint-Paul, G (2003): "*Growth effects of non-proprietary innovation*", Journal of the European Economic Association: Papers and Proceedings, v.1, n.2-3, p.429-439.

Schmidt, KM and M Schnitzer (2003): "*Public subsidies for open source? Some economic policy issues of the software market*", Harvard Journal of Law & Technology, v.16, n.2, p.473-505.

Shah, SK (2006): "*Motivation, governance, and the viability of hybrid forms in open source software development*", Management Science, v.52, n.7, p.1000-1014.

Silvo, A and F Verona (2020): "*The Aino 3.0 model*", Bank of Finland Research Discussion Papers No 9/2020.

Sims, C (2002): "The role of models and probabilities in the monetary policy process", Brookings Papers on Economic Activity, 2002, v.2002, n.2, p.1-40.

Smets, F (1995): "*Central bank macroeconometric models and the monetary policy transmission mechanism*", In: *Financial structure and the monetary policy transmission mechanism*, Central Bank, BIS p.394.

St-Pierre, NL, B Bejanov and J Jensen (2023): "*What we learned building a modelling language for macroeconomics*", JuliaCon 2023, presentation, 28 July.

Stiglitz, JE (2021): "*Rethinking macroeconomics: what failed, and how to repair it*", Journal of the European Economic Association, v.9, n.4, p.591-645, August.

Taylor, JB, V Wieland (2012): "*Surprising comparative properties of monetary models: results from a new model database*", Review of Economics and Statistics, v.94, n.3, p.800-816.

von Krogh, G, S Haefliger, S Spaeth and MW Wallin (2012): "*Carrots and rainbows: motivation and social practice in open source software development*", MIS Quarterly, v.36, n.2, p.649-676.

West, J (2003): "*How open is open enough? Melding proprietary and open source platform strategies*", Research Policy, v.32, n.7, p.1259-1285.

West, J and S Gallagher (2006): "*Challenges of open innovation: the paradox of firm investment in open-source software*", R&D Management, v.36, n.3, p.319-331.

Wickham, H and J Bryan (2023): "*R packages*", 2nd edition, O'Reilly Media.

Work stream on Eurosystem modelling (2021): "*Review of macroeconomic modelling in the Eurosystem: current practices and scope for improvement*", European Central Bank Occasional Paper Series No 267, ECB Strategy Review.

Open-sourced central bank macroeconomic models

Douglas Araujo

3rd IFC and Bank of Italy Workshop on "Data Science in Central Banking"

18 Oct 2023

Disclaimer

The views in this presentation do not necessarily represent the views of the Bank for International Settlements or of the central banks and other authorities mentioned.

All errors are my own.

This is work in progress. Additional or better information about central banks or ministries that share macroeconomic model code is welcome.

Overview

- Macroeconomic models help central banks (and other policymakers such as MinFins) to:
  - understand transmission channels of policy decisions
  - forecast the economy under different scenarios, and
  - inform policy stances in a forward-looking way

- Over time, some central banks open sourced the code of their macroeconomic models
  - interesting development in the decades-long increase in levels of central bank transparency

- This paper:
  - benchmarks how macro models are open sourced, and
  - offers practical, cost-mindful suggestions for central banks considering open sourcing models

- *Evaluating* or *categorising* the models themselves are not in scope

What is open source code?

- Simple definition: **code is publicly available**

- More complex definition (according to Open Source Initiative):
  - code is publicly available…
  - … for free…
  - … for anyone to download…
  - … using any technology… *(does not have to work in all technologies though)*
  - … and to use in their own software or derived work.

Why open sourcing model code?

- Models are already described in working papers / white papers
  - assumptions
  - building blocks
  - estimation methods
- Ideally, all relevant information on models should already be public in these papers
  - still, not always clear how to 100% replicate them from scratch just by reading papers
- Code takes transparency to next level
  - major step towards reproducibility
  - enables testing different assumptions, data, scenarios, formulas, etc
  - helps to ensure coding quality

Open source model code in context

- Central bank transparency
  - Moving from explaining decisions to sharing the objective part of “thought process”
- Academic transparency in the economics profession
  - Data and Code Availability Standard – DCAS (Koren et al, 2022)
  - Top journals: data editor, reproducibility, **code**
- Central bank code
  - Public central bank repositories (includes also non-model codes)

Central bank public code repositories by type

Best efforts list of official central bank repositories[1] Table 1

Institution	Internal utilities	Payments, CBDC, open banking	Research replication	Macroeconomic models
Central Bank of Brazil		X		
National Bank of Belgium	X			
Bank of Canada			X	X
Central Bank of Colombia			X	
National Bank of Denmark			X	
European Central Bank			X	X
Bank of France	X			
Bank of England	X		X	
National Bank of Greece		X		
Bank of Italy	X		X	X
Dutch Central Bank	X			
Bank of Norway		X		
Federal Reserve Bank of Boston (collaboration with MIT)		X		
Federal Reserve Bank of New York			X	X

Information as of 22 September 2023. [1] The helpful list of central bank open source repositories kept by the Dutch Central Bank is kindly acknowledged.

Source: Author compilation, authorities' websites.

Implications of model reproducibility

- Central bank models *inform* policy, they don't *decide* policy
  - Policy = **model output** + **qualitative expert knowledge** + **judgement**
  - Model code only informative about the first component

- Some central banks might be more willing to share code for *simulation models* to understand transmission channels rather than production-grade *forecast models*

- Model code can be helpful to other central banks, in particular those in smaller countries:
  - adapting for their own domestic economy; or
  - as workhorse model for major economies, helping them evaluate the external scenario

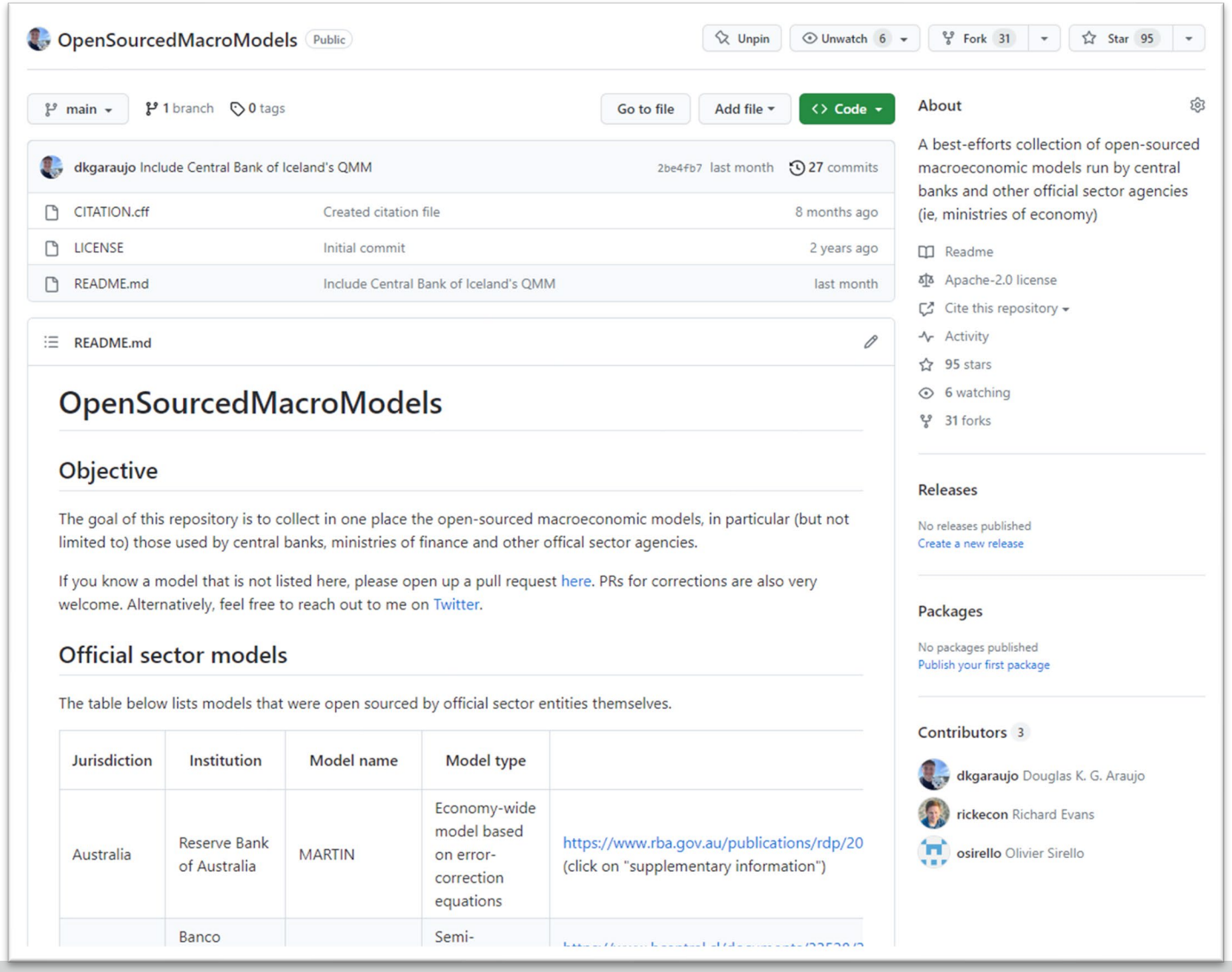
Who shares model code currently?

- Australia – RBA
- *Canada – Bank of Canada
- Chile – Banco Central de Chile
- Denmark – Min Finance
- European Union – ECB, European Commission
- Finland – Bank of Finland
- France - Min of Economics and Finance
- Germany - Min of Economic Affairs and Climate Action
- Iceland – Central Bank of Iceland
- *Italy – Banca D'Italia
- Japan – Bank of Japan
- Sweden - Riksbank
- UK - HM Treasury
- US - FRB, FRBNY

* not a model itself, but a modelling package

If you know another **open source model**, please let me know!

Who shares model code currently?



OpenSourcedMacroModels Public

Unpin | Unwatch 6 | Fork 31 | Star 95

main | 1 branch | 0 tags | Go to file | Add file | Code

dkgaraujo Include Central Bank of Iceland's QMM 2be4fb7 last month 27 commits

CITATION.cff	Created citation file	8 months ago
LICENSE	Initial commit	2 years ago
README.md	Include Central Bank of Iceland's QMM	last month

README.md

OpenSourcedMacroModels

Objective

The goal of this repository is to collect in one place the open-sourced macroeconomic models, in particular (but not limited to) those used by central banks, ministries of finance and other offical sector agencies.

If you know a model that is not listed here, please open up a pull request here. PRs for corrections are also very welcome. Alternatively, feel free to reach out to me on Twitter.

Official sector models

The table below lists models that were open sourced by official sector entities themselves.

Jurisdiction	Institution	Model name	Model type	
Australia	Reserve Bank of Australia	MARTIN	Economy-wide model based on error-correction equations	https://www.rba.gov.au/publications/rdp/20 (click on "supplementary information")
	Banco		Semi-	

About

A best-efforts collection of open-sourced macroeconomic models run by central banks and other official sector agencies (ie, ministries of economy)

Readme
Apache-2.0 license
Cite this repository
Activity
95 stars
6 watching
31 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

dkgaraujo Douglas K. G. Araujo
rickecon Richard Evans
osirello Olivier Sirello

Check out the repo!

Open-sourced central bank models used in the assessment

Best-efforts list of official models with available code

Table 2

Jurisdiction	Institution	Model name	Model ID[1]	Reference
Australia	Reserve Bank of Australia	MARTIN	AU-01	Ballantyne et al (2019)
Chile	Central Bank of Chile	MSEP	CL-01	Arroyo Marioli et al (2020), BCC (2020)
Chile	Central Bank of Chile	XMAS	CL-02	García et al (2019), García and Guerra-Salas (2020), BCC (2020)
Denmark	Ministry of Finance[2]	MAKRO	DK-01	Bonde et al (2023)
European Union	European Central Bank	BEAR	EU-01	Dieppe and van Roye (2023)
European Union	European Commission[2]	Output Gap Model	EU-02	
Finland	Bank of Finland	Aino	FI-01	Kilponen et al (2016)
France	Ministry of Economics and Finance[2]	Mésange	FR-01	Dufernez et al (2017)
France	Ministry of Economics and Finance[2]	Opale	FR-02	Daubaire et al (2017)
France	Ministry of Economics and Finance[2]	Modèle Saphir	FR-03	
Germany	Federal Ministry for Economic Affairs and Climate Action[2]	-	DE-01	BMWK and BMFin (2022)
Iceland	Central Bank of Iceland	QMM	IS-01	Daníelson et al (2019)
Japan	Bank of Japan	Q-JEM	JP-01	Hirakata et al (2019)
Sweden	Riksbank	Ramses II	SE-01	Adolfson et al (2013)
United Kingdom	HM Treasury and Office for Budget Responsibility[2]	OBR model	GB-01	Office for Budget Responsibility (2013)
United States	Federal Reserve Bank of New York	DSGE.jl	US-01	
United States	Federal Reserve Board	FRB/US	US-02	Brayton et al (2014), Laforte (2018)
United States	Federal Reserve Board	EDO	US-03	Chung et al (2010)

Information as of 22 September 2023. [1] Model ID is used throughout the text to refer to each model. [2] Other official financial agencies such as Ministries of Finance are also in scope of this compilation.

Source: Author compilation, authorities' websites.

Benchmarking model open sourcing

Core criteria

- Open access
  - Easy to find
  - Free access
- Documentation
  - Academic description
  - API documentation
  - Input data documentation
  - Output documentation
- Replication
  - End-to-end execution
  - Instructions/tutorial/vignette
  - Minimum requirements

Additional criteria

- Software
  - Software open availability
  - Testing
  - Technology neutrality
  - Availability of past versions
  - Explicit code versioning
- Contribution
  - Contact with software maintainers
  - 3rd party contributions possible
  - Contribution guidelines
- License
  - Explicit open source license

Characteristics of open-sourced macroeconomic model codes

Best efforts list of official models with available code

Table 3

	Core criteria									Additional criteria									
	Open access		Documentation				Replication			Software					Contribution			Lic.
Model ID[1]	Easy to find	Free access	Academic	API	Input data	Output data	End-to-end	Instructions	Requirements	Open software	Tech neutrality	Testing	Explicit version	Past versions	Contact	Third parties	Guidelines	Explicit license
AU-01	X	X	X	X		X	X	X	X									X
CL-01	X	X	X	X	X	X	X	X	X	X	X				X			
CL-02	X	X	X	X		X	X	X	X	X	X				X			
DE-01		X	X	X	X		X			X			X	X				
DK-01	X	X	X	X	X		X		X		X		X	X	X	X		X
EU-01	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X
EU-02	X	X		X	X		X		X				X	X				
FI-01	X	X	X	X	X	X	X		X	X	X							
FR-01	X	X	X	X	X	X	X	X	X				X	X	X	X		X
FR-02	X	X	X	X	X	X	X	X	X				X	X	X	X		X
FR-03	X	X				X	X	X	X				X	X	X	X		X
IS-01	X	X	X	X	X	X	X	X										X
JP-01	X	X	X	X	X	X	X	X	X						X			X
SE-01	X	X	X	X			X	X		X	X				X	X		
GB-01	X	X	X	X	X		X		X									
US-01	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
US-02	X	X	X	X	X		X	X	X	X	X							
US-03	X	X	X	X			X			X	X							X

Information as of 22 September 2023. [1] See Table 2.

Source: Author compilation, authorities' websites.

Core criteria: general comments

- Most models fulfil core criteria for code sharing
- Main “core” gaps found in:
  - Documentation of data
  - Instruction usage
- These can be expected to improve over time, as interest by third parties increase
  - Could also be good for internal purposes, eg onboarding of new economists

Most common programming languages for macroeconomic models

Best efforts list of official models with available code

Table 4

Languages	Open source	Multi-platform	Observations	Models
Matlab (or Octave)	No (Yes)	Yes	Most models with Dynare	CL-01, CL-02, EU-01, SE-01 US-03
EViews	No	No		AU-01, GB-01, JP-01[2], US-02[2]
Portable TROLL	No	Yes (raw code; GUI is only available for Windows)		FR-01, FR-02
Julia	Yes	Yes		US-01[1]
SAS	No	No		FR-03
EUCAM	No	No		EU-02
GAP	No	No		DE-01
GAMS	No	No	APIs available for Python, Java and other languages	DK-01

Information as of 22 September 2023. [1] Based on previous Matlab code. [2] Also has a Python version.

Source: Author compilation, authorities' websites.

Additional criteria: contribution

- Judging by "revealed preferences", contribution from third-parties does not seem to be a driver of code sharing up until this point
- Understandably, code sharing to improve transparency of CB models (which are arguably the best for each particular economy), rather than to elicit feedback or improvements
- This could be an avenue to explore, but will probably require cultural shifts

Additional criteria: license

- Ideally, publicly available code should have an explicit license – even if it is not open source
  - Licenses protect both contributors and users
- But, so far many model codebases are not licensed
  - Lack of license: no official permission to copy, distribute or modify
  - If lack of license is intentional, should be explicit
- In practice most users will not care, but other official users (eg, central banks) will
- Wide variety of off-the-shelf licenses, and they can also be custom-made

Choose an open source license

An open source license protects contributors and users. Businesses and savvy developers won't touch a project without this protection.

{ Which of the following best describes your situation? }

I need to work in a community.

Use the **license preferred by the community** you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to **add a license**.

I want it simple and permissive.

The **MIT License** is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

Babel, **.NET**, and **Rails** use the MIT License.

I care about sharing improvements.

The **GNU GPLv3** also lets people do almost anything they want with your project, *except* distributing closed source versions.

Ansible, **Bash**, and **GIMP** use the GNU GPLv3.

Practical advice to enhance code availability

- Make it easier to find
  - Listing available code or point to repository in one place in the central bank's website
- Make it easier to use
  - Create tutorials and examples
  - Be explicit about the license
- Make it easier to contribute
  - Set up a code repository
- Make it more accessible and future-proof
  - Consider adapting code requiring proprietary software to OS languages (AI can help)

Tentative conclusions

- While some central banks share code for many years now, discussion on its benefits and challenges is still scarce
- This work hopes to shed light on this practice, to help CBs harness the benefits from transparency when that is convenient
- Code-sharing can boost CB cooperation and technical assistance, but it can also be a benefit for internal staff

Many thanks!
Question, feedback, criticisms…?