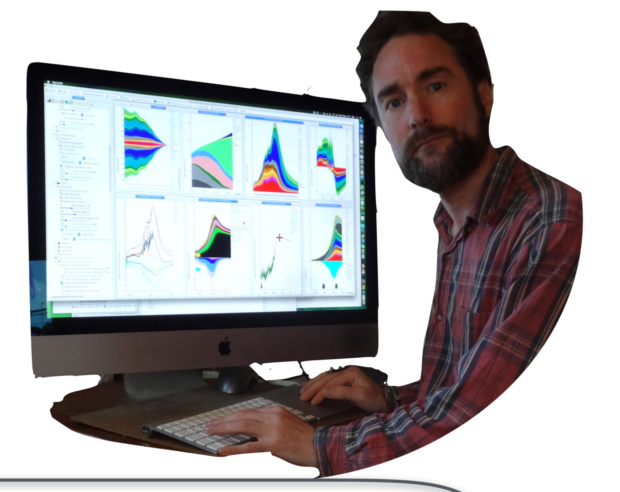
An interactive model revived with a scalable codebase, for exploring future pathways

This is about an interactive web model, so a static poster is a poor format to show it. Please try for yourself, just visit:

https://swim.benmatthews.eu/iamc2021

Dr Ben J.H. Matthews

ben@benmatthews.eu



* *You must use a **recently updated** web browser (since summer-21)! - needs latest javascript*
* This is a special preview for IAMC - it will be online just before this conference, will move later
* This web version is a prototype - work in progress - no documentation, parts missing or not working, still rapidly developing.
* Please visit again a few weeks later to see what's new

This tree illustrates some the parameters and plots available in the model, organised by module. More appear, depending on visible plots and options.

The core niche of the interactive Java Climate Model (JCM) was to enable stakeholders (anybody) to easily explore the relative sensitivity to diverse options and uncertainties. Developed since 2000, first as an applet and later via "webstart", this worked well until tech-giants blocked java in browsers (it profited neither app-stores nor cloud-servers).

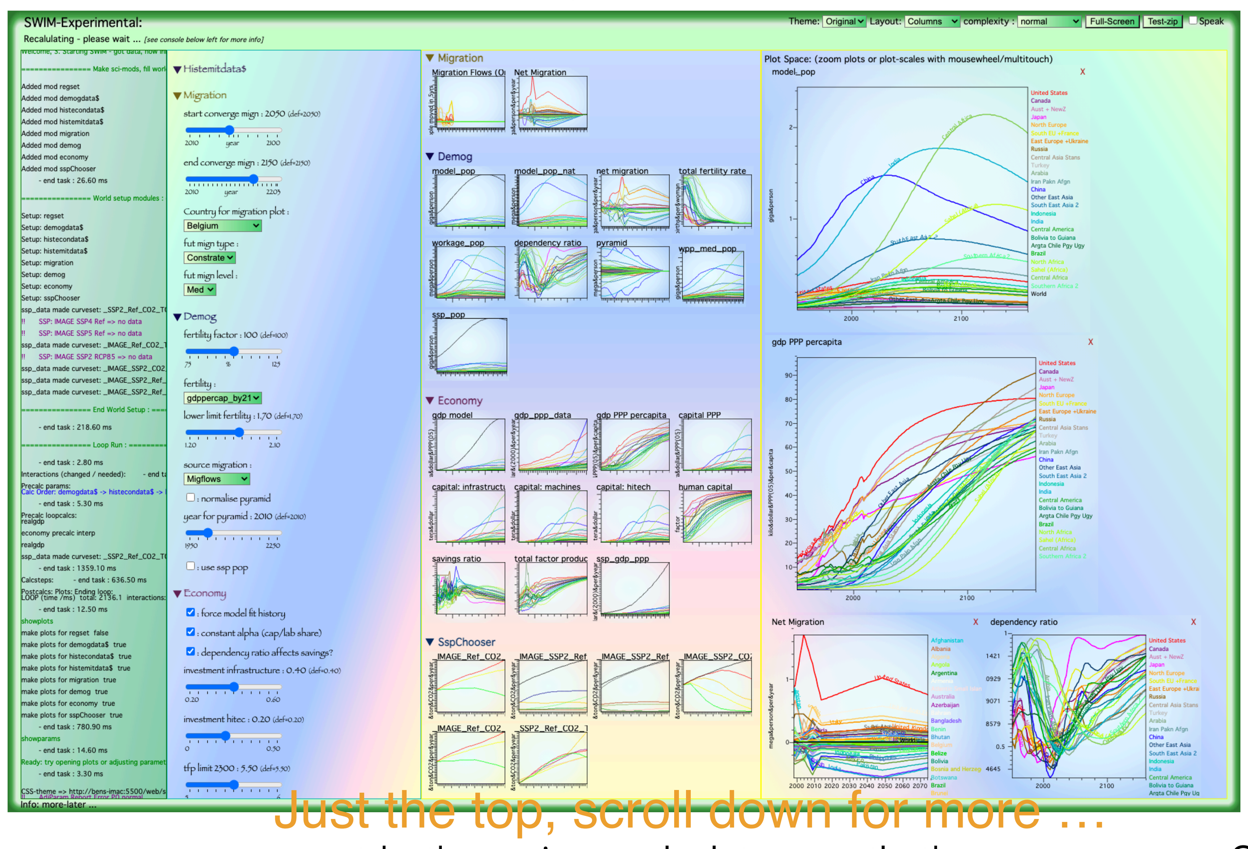
So how to revive this ? => old JCM translated to Scala-3 (new language launched May 2021):

one Scala3 codebase => compiles to two views:

Scala.js web version

with new Scala-Javascript graphical interface

Easy access => better for exploring system, illustrating points
Open in a web browser, no installation, it just works
Test relative sensitivity to policy options and scientific uncertainties
Response time to parameter adjustments: ~ 0.5-5s

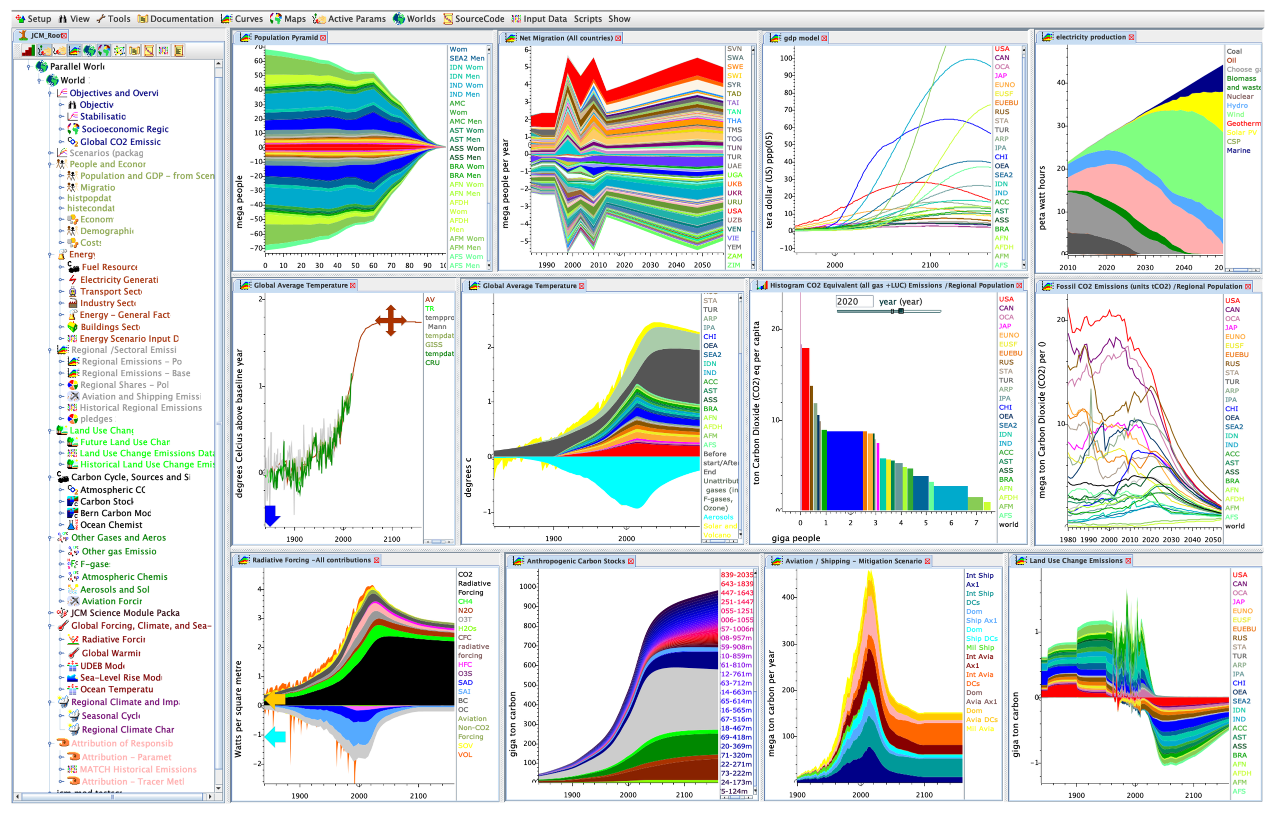


Just the top, scroll down for more ...

JVM desktop version

applying original java GUI

Faster => better for systematic analysis
Run scripts, probabilistic analysis of thousands of variants
Simpler debugging for code in development
Response time to to parameter adjustments: ~ 0.1-1s



Not a small project ...:

~ 50 science modules,
~ 22000 lines of code
(excl comments /blank/libraries)
~ 250 parameters
(user-adjustable: see far left)
~ 165 plots (curve-sets)
(excl derivatives eg. ratios)
~ 100 history data-tables
~ 50k words doc (to-fix)

yet both versions calculate exactly the same curves. Scala3 - strongly typed, multi-paradigm, yet with readable syntax, is both robust and flexible

A dynamic system, not a data or scenario visualiser

Instant dynamic response to infinite parameter combinations, everything is connected, re-calculates in your computer (not on a server)

Real models, not trivial functions: for example... :

Demography: - 5yr age-groups for each country, including migration
Electricity supply: - Dynamic cost-driven investments
Land use change: - Biome change maps, explore uncertainty in historical fluxes
Policies: - National pledges, diverse equitable sharing options
Scenarios: - Inverse stabilisation scenarios, blend with SSPs
Chemistry: - Atmospheric interactions, ocean carbonate /pH
Forcing: - All greenhouse gases and aerosols, incl aviation cirrus
Ocean carbon / heat: Diffusion + upwelling with~ 40 ocean layers, tuned to GCMs

Not a small project... :

~20 years development - in Belgium, Switzerland, Brazil, Denmark, Norway
(conversion to Scala-3 last seven months)

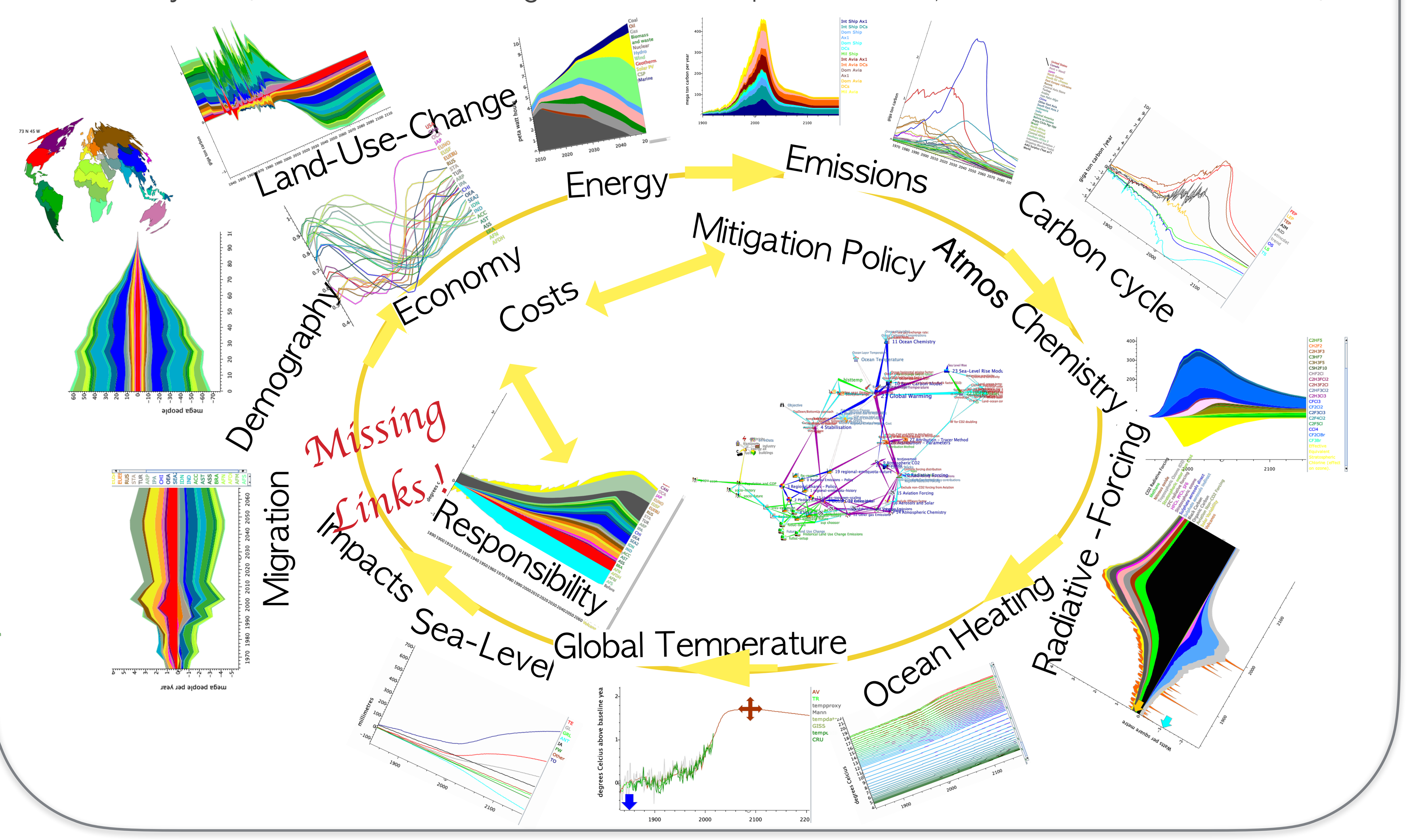
Timescale: 1750 - 2250, 1yr steps, smooth discontinuity history-future
Regions: calculates by nations or diverse region-sets - subnational planned
Feedbacks e.g. climate => soil respiration, economy => fertility
Flexible loop order =>top-down/bottom-up, dependencies =>efficient recalculation

Yet all this now just works in a web browser, even on a tablet or phone - although you need a big screen to appreciate it.

(Tablets offer new ways to scroll and zoom to explore the system space - I'm working on this - please visit again later)

Designed for feedbacks, completing the loop

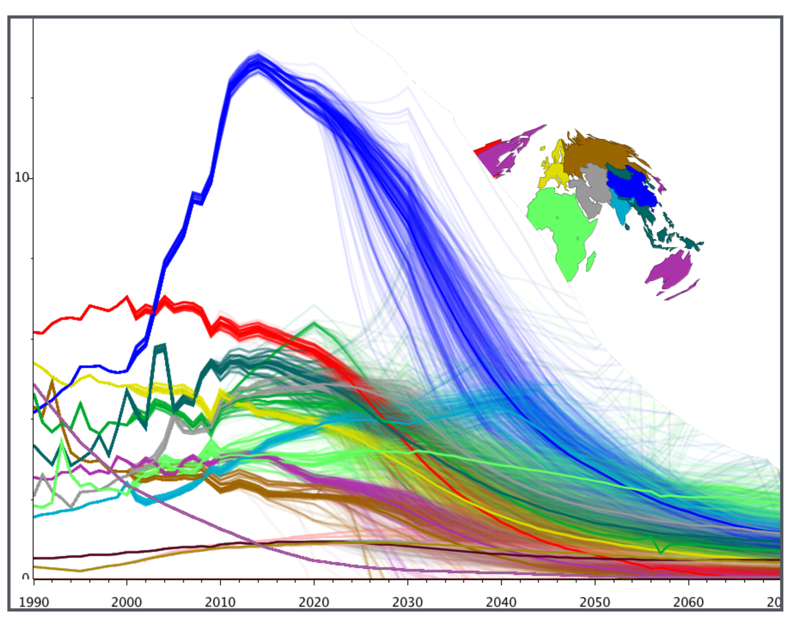
Explore response from "drivers" to "impacts" - e.g adjust a demography parameter, and see the change up to the sea-level .... BUT: the loop is not complete - sea-level should impact population (especially migration) and economy etc. The core loop step structure was designed to enable such feedbacks, but we needs impacts modules calibrated from more detailed analyses. (JCM had some old regional climate / impacts modules, but removed as too outdated)



Probabilistic stabilisation analyses

A model structure designed for rapid response to parameter adjustment, is also efficient for systematic analysis combining thousands of calculations. Combined with an iterative inverse algorithm to find temperature stabilisation pathways, this was applied (way back in 2003) to make the first probabilistic analysis of pathways below 2C, with carbon and climate model variants constrained by historical data. More recently (2016) a similar approach was applied to 1.5C and 1.75C, generating ranges of regional pathways based on a range of equitable sharing criteria, as well as scientific uncertainties. A 1.75C variant is shown below (outdated, but illustrates the method).

Similar inverse stabilisation analysis, comparing stabilisation of concentration, radiative forcing and temperature ("shifting the burden of uncertainty") helped influence the shift in global policy towards temperature goals. This also influenced the development of the parallel RCP+SSP scenario structure, also with the hope of completing the loop (left) (the author participated in that IPCC - initiated process, which in turn helped motivate the creation of IAMC). So, interactive web tools can grow and lead to greater outcomes.



Some Obvious Questions:

✦ *Was it easy to convert JCM to Scala.js?* No... *Is complete?* Almost (to fix: ocean eigenvectors+luc polygons). *Was it worth it?* Yes (flexible & readable code, smart compiler, scala.js makes it visible again)

✦ *Why develop a js-web version before updating old science / data ?* To check scala-js sufficiently fast and robust, to prove concept as tool only useful if visible. Updating important but have done it before - historical datasets, model tuning and "factors" (climate, chemistry, costs etc.)

✦ *Why calculate everything client side, the fashion is clouds?* Developed for interactivity. Scala code easily moves client <=> server, so later could adapt as efficient - eg if slow web + fast computer => more client side or vice versa, frequently adjusted => client, rarely changed (historical) => server.

✦ *What about zooming within countries as promised in the abstract?* Coming, later but it's important, far more people in Uttar Pradesh or Sichuan than european nations... Can you help with datasets?

✦ *JCM specialised in equitable stabilisation pathways, is this relevant post-COP26 heading to net zero?* Grand policies need implementation detail, such tools help put local efforts in global context.

✦ *Where's the documentation / translations ?* JCM had >50,000 words docn, auto-adapting to model state, and plot-labels in many languages. It's broken but will return.

✦ *Can I see / change the source code?* Later - when the structure settles down. *Will the desktop app be downloadable?* Later - current focus on web version.

✦ *Who funds this?* Nobody yet. *So why...?* It's what I can do, long story... *Can I help?* Please get in touch !

✦ *What is it called now?* Needs new name (no longer java, may diversify beyond climate). Temporarily "Scalable World Interactive Model" (SWIM) *(read abstract for explanation "scalable").*

Next steps ...?

Many todos, priority can depend on your feedback and collaboration

★ Update core science to be consistent with AR6
★ Develop impacts => migration to complete the loop (above)
★ Update pledges (NDCs) post COP26, towards COP28 stocktake.
★ Develop variant to contribute to new scenarios process
★ Develop future LUC options, more on agriculture, methane, biodiversity
★ Subdivide large countries with provincial detail especially relevant to
  ★ Energy transition
  ★ Migration, including (but not only) climate-related
★ Re-develop educational version (recall early "applet" JCM used for teaching in many countries) - questions auto-adapting to student level
★ Combine long history of model with author's experience in UNFCCC/IPCC to illustrate changing scenarios and policy visions over 25 years (roll back time to help youth see what might-have-been, not only "bla-bla-bla")?
★ Fuzzy-control policy responses (as if people don't believe models!)

So please tell me what you think... (email above)