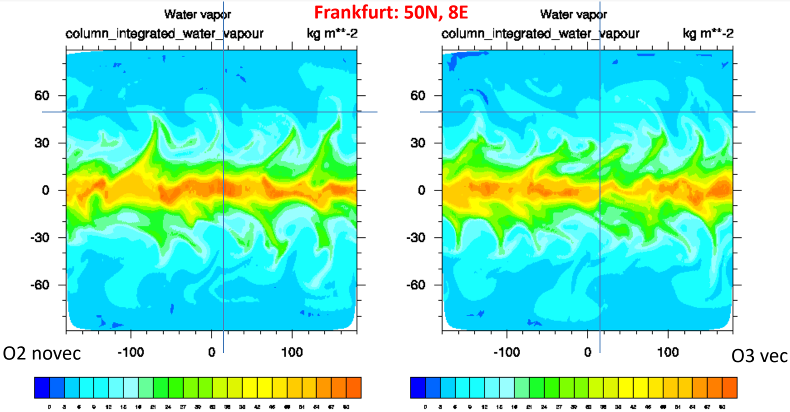
Reproducing Simulations

Nico Formánek
High Performance Computing Center (HLRS)

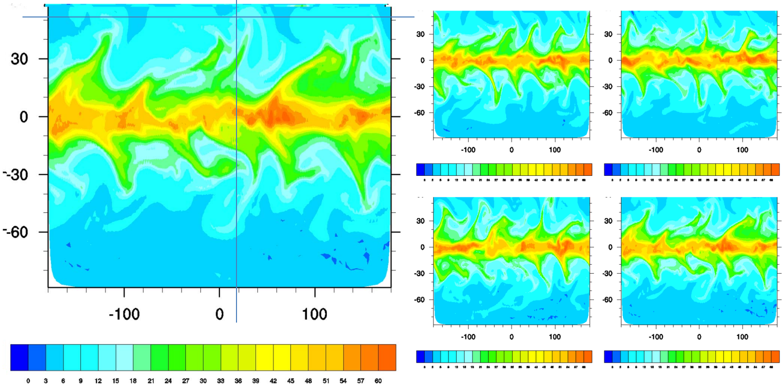
July 2nd, 2025

t_{sim}=100d O2: 10min30sec -> O3: 8m41sec => (+17,3%)

Frankfurt: 50N, 8E



t_{sim}=100d O3 -x AVX – 4 runs with identical input



The usual suspects

How to get Meaningless Answers in Scientific Computation (and what to do about it)

L. FOX, FIMA
Oxford University Computing Laboratory

1. Introduction

THE phrase "scientific computation" essentially means the numerical solution of problems of a scientific nature, as distinct from other computer applications such as "payroll," textual analysis, and so on. I am therefore talking to scientists and engineers, the people confronted with such problems.

Now those of us who try to teach numerical mathematics in universities or other centres of learning are frequently discouraged by the discovery that those who have problems to solve just will not learn how to solve them. Introduce computer programming to scientists and they flock to learn. Introduce numerical mathematics and they disappear in droves.

The undergraduate mathematics books they read will rarely include any serious numerical studies, and the research literature they read will also be lacking in this topic. Their teachers, both at undergraduate and graduate level, will rarely have more than a smattering of rather old numerical knowledge, and I have even seen could ever possibly have any meaning at all. Well (*a*) these *are* "your problems," (*b*) all the examples (although perhaps slightly simplified) have turned up in practice and (*c*) far from trying to spread panic I am trying to indicate that there is something to learn, that it is interesting, worth learning and not really very difficult except perhaps at a very advanced stage.

In what follows I give, in Part A, some illustrations of "how to get rubbish." Most of these cases have appeared in previous literature, and this is indicated where relevant. In Part B I try to indicate the sort of thing that is needed to overcome these problems, with a conclusion in Part C that varies from the deepest pessimism to a few gleams of hope.

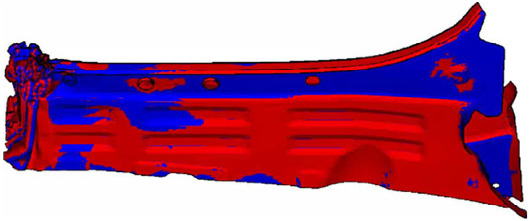
A. How to get rubbish

2. Your problem might be ill-conditioned (suffer from inherent instability)

Example 2.1 (reference 1, p. 75)
Consider the linear equations

- IEEE floats (not associative).
- Parallel computing (order of execution).
- Optimization.
- (Silent errors)

Non-reproducibility is expected



Variational crimes in FEM (Mangold et al. 2012), (Fillion and Corless 2021)

[Nonreproducibility] is more of a feature than a problem; it's thus possible to conclude that reproducibility **isn't a requirement that absolutely must be enforced**.

(Diethelm, 2012)

Should we enforce reproducibility?

When it comes to HPC, when comparing results of the same computer experiments, **we need: Bitwise identical results** *for repeatability: achieving exactly the same results from run to run under the same conditions.*

(Antunes, Hill, 2024)

Should we enforce reproducibility?

> *When it comes to HPC, when comparing results of the same computer experiments,* **we need: Bitwise identical results** *for repeatability: achieving exactly the same results from run to run under the same conditions.*
>
> *(Antunes, Hill, 2024)*

The HPC community reached no consensus about:

1. If failure of bit-wise reproduction is a problem.
2. What kind of reproducibility we should expect instead.

The central epistemological principle of (computational) modeling

> *[...] computational errors should be analyzable in the same terms as modeling and experimental errors. By that we mean that if truncation, discretization, and roundoff errors are small compared to the modeling and experimental error, then for all we know, our approximate numerical answer can be the right one.*

(Fillion and Corless, 2014)

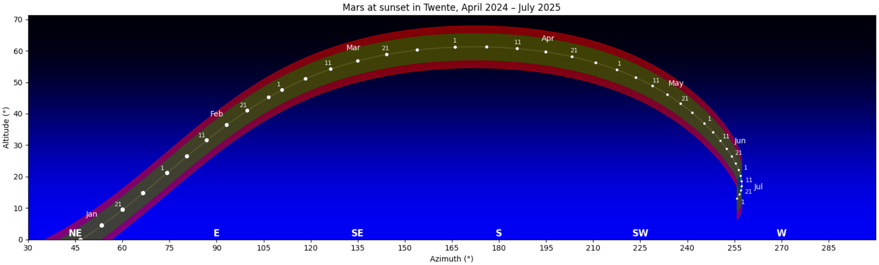
The central epistemological principle of (computational) modeling

> *[...] computational errors should be analyzable in the same terms as modeling and experimental errors. By that we mean that if truncation, discretization, and roundoff errors are small compared to the modeling and experimental error, then for all we know, our approximate numerical answer can be the right one.*

(Fillion and Corless, 2014)

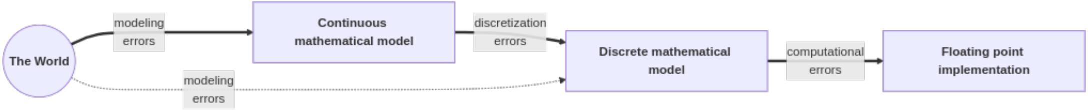
- Everything is fine if **modeling error** $\approx$ **computational error**.

Illustration of the principle

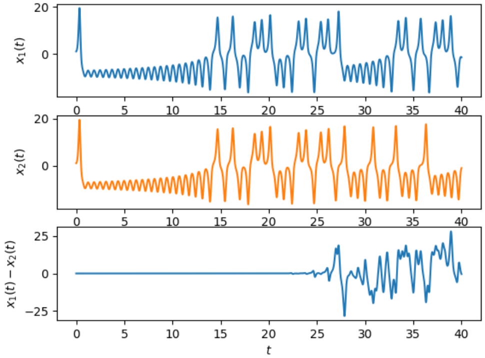


Consistent with Epicycles, Oberservational error.

Errors are getting mixed in (computational) modeling



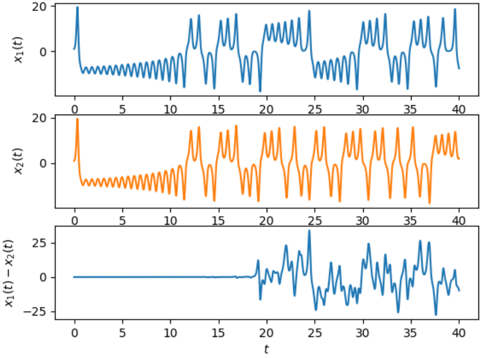
Lorenz 1963 - Example



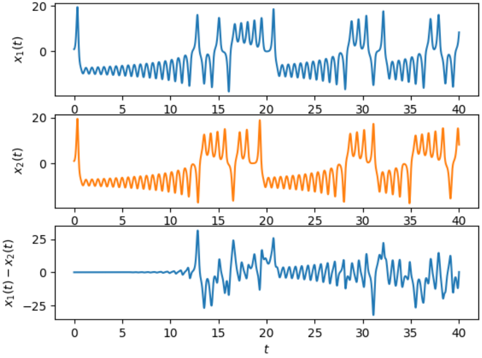
Two numerical solutions, inital conditions differing in 1/100000

- Modeling target: atmospheric convection.
- Continuous model: system of differential equations.
- Discretization and Numerics: scipy . solve_ivp

Lorenz 1963 - Example

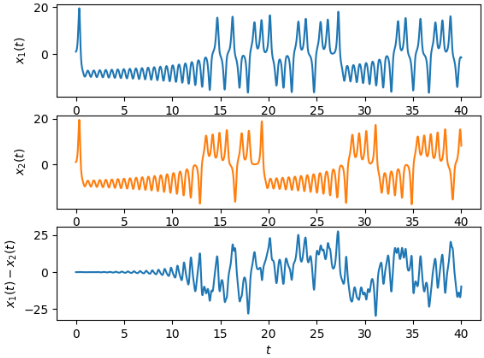


Numerical solutions with method='BDF'



Numerical solutions with method='LSODA'

Lorenz 1963 - Example



Solution 1 with method='RK45', Solution 2 with method='LSODA'

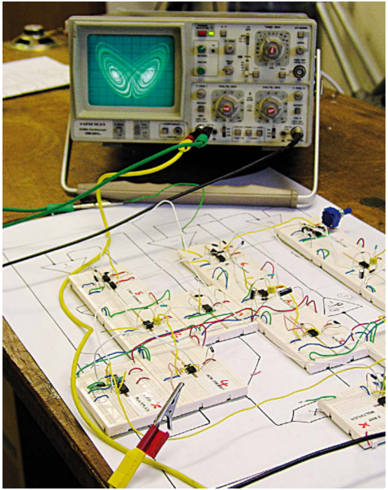
- Is the atmosphere sensibly dependent on initial conditions (SDIC)?
- Is the initial value problem ill-conditioned?
- Are the numerics unstable?

Why is error control important?

> *The situation has the whiff of paradox. We may think a system is sensitively dependent because of computer trials: but then, if it is sensitively dependent, why trust the computer trials?*
> *(Smith, 1998)*

- Computer methods mix all the errors in the modeling process.
- Can we disentangle them?
- Who is to blame? (reality? model? numerics? everyone?)

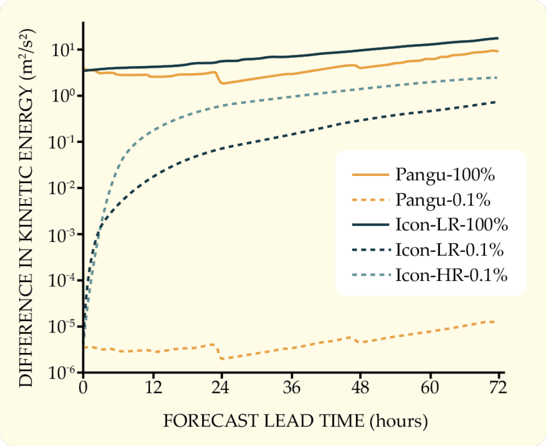
Some ways to disentangle error



Analog computer implementing Lorenz 63

- Prove that method is numerically stable (stability often depends on problem).
- Prove that continuous model is ill-conditioned/well-conditioned (often not known how).
- Establish that target system has property independently (from computer method).
- Compare to experiments.

Current debate in weather modeling



The so called real butterfly effect. (see `https://doi.org/10.1063/pt.eike.hsbz`)

- Classical physics based models show real butterfly effect.
- ML models do not.
- Long standing conviction among meteorologists: real butterfly effect is property of the atmospheric system.
- Recently: Perhaps not, because ML models do not show it.

How much reproducibility should we expect?

The level of reproducibility of a computing method depends on:

1. The numerical properties of the computing method.
2. The mathematical properties of the continuous model.
3. The properties of the modeling target.

If we cannot disentangle them, we cannot tell how much reproducibility we can expect.