Multi-Model Communication Project
University of Wisconsin-Madison

Gershgorin-Majda 2010 System Simulator

Jason Turner

15th December, 2020

Contents

1. Introduction

This code is a test example for the multi-model forecast Kalman filter, as initially formulated in November 2020. In particular, we assume linear, Gaussian models and utilize some linear combination of our models to create a psuedo-truth signal for each model that that model observes. The purpose of this code, at least initially, is to be able to run the Gershgorin-Majda 2010 system, as well as the additive and multiplicative models.

This document is inteded to be supplemental documentation for the `README.md` files included throughout the code repository.

1.1 The Gershgorin-Majda 2010 System

Although this is not the official name for this test system, it was first proposed by Gershgorin and Majda in [1]. Specifically, the exactly solvable test model is

$$\frac{d\,u(t)}{dt} = (-\gamma(t) + i\,\omega)\,u(t) + b(t) + f(t) + \sigma\,\dot{W}(t), \tag{1}$$

$$\frac{d\,b(t)}{dt} = \left(-\gamma_b + i\,\omega_b\right)\left(b(t) + \widehat{b}\right) + \sigma_b\,\dot{W}_b(t), \tag{2}$$

$$\frac{d\,\gamma(t)}{dt} = -d_\gamma\left(\gamma(t) + \widehat{\gamma}\right) + \sigma_\gamma\,\dot{W}_\gamma(t), \tag{3}$$

where $u(t)$ and $b(t)$ are complex-valued and $\gamma(t)$ is real-valued. The terms $b(t)$ and $\gamma(t)$ represent additive and multiplicative bias corrections terms. Also, ω is the oscillation frequency of $u(t)$, $f(t)$ is external forcing, and σ characterizes the strength of the white noise forcing $\dot{W}(t)$. The parameters γ_b and d_γ represent the damping and parameters σ_b and σ_γ represent the strength of the white noise forcing of the additive and multiplicative bias correction terms, respectively. The parameters $\widehat{b}$ and $\widehat{\gamma}$ are the stationary mean bias correction values of $b(t)$ and $\gamma(t)$, correspondingly, and ω_b is the frequency of the additive noise. Note that the white noise $\dot{W}_\gamma(t)$ is real-valued while the white noises $\dot{W}(t)$ and $\dot{W}_b(t)$ are complex-valueds and their real and imaginary parts are independent real-valued white noises.

In [1], they consider an oscillatory external forcing

$$f(t) = A_f\,e^{i\,\omega_f\,t} \tag{4}$$

with A_f the strength og the external forcing and ω_f its frequency. For the purpose of testing some parameter estimation techniques, Gershgorin and Majda considered an additive model of the original Gershgorin-Majda 2010 system where there is only additive bias correction

$$\frac{d\,u(t)}{dt} = \left(-\overline{d} + i\,\omega\right)\,u(t) + b(t) + f(t) + \sigma\,\dot{W}(t)\,, \tag{5}$$

$$\frac{d\,b(t)}{dt} = \left(-\gamma_b + i\,\omega_b\right)\,\left(b(t) + \widehat{b}\right) + \sigma_b\,\dot{W}_b(t)\,, \tag{6}$$

where $\overline{d}$ is the mean value of the damping, as well as a multiplicative model where there is only multiplicative bias correction

$$\frac{d\,u(t)}{dt} = \left(-\gamma(t) + i\,\omega\right)\,u(t) + f(t) + \sigma\,\dot{W}(t)\,, \tag{7}$$

$$\frac{d\,\gamma(t)}{dt} = -d_\gamma\,\left(\gamma(t) + \widehat{\gamma}\right) + \sigma_\gamma\,\dot{W}_\gamma(t)\,. \tag{8}$$

2. Compiling and Running

This code utilizes CMake as a build system, and was designed for compilation on Ubuntu 20.04.1. In particular, here are the requirements for compiling and running the code:

- CMake (at least version 3.16).
- netCDF-Fortran (at least version 4.8.0).

Once you have these, simply navigate to the `build` subdirectory and enter the standard CMake commands

```
cmake ..
cmake --build .
```

This will compile the code, build the headers and `NAMELIST`, and create the executable `gershgorin_majda_10`. To run the code, enter the command

```
./gershgorin_majda_10
```

and an output file `out.nc` will be generated.

3. Code Structure

In the most overview sense, the simulation code contains a single driver subroutine that calls subroutines to perform the following tasks:

1. Initialize the simulation, including reading the `NAMELIST`.

2. Execute the simulation.
3. Finalize the simulation.

Acknowledgements

This project took advantage of netCDF software developed by UCAR/Unidata (http://doi.org/10.5065/D6H70CW6) and CMake software.

References

[1] B. Gershgorin, J. Harlim, and A. Majda, *Test models for improving filtering with model errors through stochastic parameter estimation*, Journal of Computational Physics, 229 (2020), pp. 1–31.