

# Review questions

Edward L. Ionides, Aaron A. King, Qianying Lin

July 7, 2021

## Questions on Lesson 2: Simulation of stochastic dynamic models

**Question 2.1.** Scientifically, our conclusions should not depend on the units we choose, but we must get the details right. Suppose our data are two years of weekly aggregated case reports of a disease and we have a continuous time model solved numerically by an Euler timestep of size  $dt$ . Which of the following is a correct explanation of our options for properly implementing this in a `pomp` object called `po`?

- (A) The measurement times, `time(po)`, should be in units of weeks, such as  $1, 2, \dots, 104$ . The latent process can be modeled using arbitrary time units, say days or weeks or years. The units of  $dt$  should match the time units of the **latent** process.
- (B) The measurement times, `time(po)`, should be in units of weeks, such as  $1, 2, \dots, 104$ . The latent process can be modeled using arbitrary time units, say days or weeks or years. The units of  $dt$  should be in weeks (in practice, usually a fraction of a week) to match the units of the **measurement** times.
- (C) The measurement times do not have to be in units of weeks. For example, we could use `time(po)=1/52, 2/52, ..., 2`. The latent process and  $dt$  should use the same units of time as the measurement times.
- (D) The measurement times do not have to be in units of weeks. For example, we could use `time(po)=1/52, 2/52, ..., 2`. The latent process can also use arbitrary units of time, which do not necessarily match the units of the measurement times. The units of  $dt$  should match the units used for the **latent** process.
- (E) The measurement times do not have to be in units of weeks. For example, we could use `time(po)=1/52, 2/52, ..., 2`. The latent process can also use arbitrary units of time, which do not necessarily match the units of the measurement times. The units of  $dt$  should match the units used for the **measurement** times.

**Solution.** (C). For scientific calculations, you generally have to pick an arbitrary set of units and use it consistently. In `pomp`, this means that you have to use the same units for measurement times and within the latent process. For example, if your measurement times are in days (7,14,...) then rate parameters should have units  $\text{day}^{-1}$ . A latent transition with mean duration 1 week would have corresponding rate  $1/7\text{day}^{-1}$ .

**Question 2.2.** Suppose you obtain the following error message when you build your `pomp` model using Csnippets.

```
Error: error in building shared-object library from C snippets: in 'Cbuilder': compilation error:
cannot compile shared-object library '/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_be9007eb030e47cb34264e3e779b6da9.so':
status = 1
compiler messages:
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
-I'/Users/ionides/Library/R/x86_64/4.1/library/pomp/include'
-I'/Users/ionides/sbied/questions' -I/usr/local/include -fPIC -Wall -g -O2
```

```

-c /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_be9007eb030e47cb34264e3e779b6da9.c
-o /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_be9007eb030e47cb34264e3e779b6da9.o
/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_be9007eb030e47cb34264e3e779b6da9.c:39:5:
error: called object type 'int' is not a function or function pointer
    W = 0;
    ~
/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/5
In addition: Warning message:
In system2(command = R.home("bin/R"), args = c("CMD", "SHLIB", "-c", :
running command 'PKG_CPPFLAGS="-I'/Users/ionides/Library/R/x86_64/4.1/library/pomp/include'
-I'/Users/ionides/sbied/questions'" '/Library/Frameworks/R.framework/Resources/bin/R'
CMD SHLIB -c -o /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_be9007eb030e47cb34264e3e779b6da9.so
/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_be9007eb030e47cb34264e3e779b6da9.c
2>&1' had status 1

```

Which of the following is a plausible cause for this error?

- (A) Using R syntax within a C function that has the same name as an R function.
- (B) A parameter is missing from the `paramnames` argument to `pomp`.
- (C) Indexing past the end of an array because C labels indices starting at 0.
- (D) Using `beta` as a parameter name when it is a declared C function.
- (E) A missing semicolon at the end of a line.

**Solution.** (E). The error message was produced by the code below. `pomp` passes on the C compiler error message for you to inspect. Note the missing semicolon at the line end before `W=0`;

```

sir1 <- sir()
sir2 <- pomp(sir1, statenames=c("S", "I", "R", "cases", "W"),
  paramnames=c(
    "gamma", "mu", "iota",
    "beta1", "beta_sd", "pop", "rho",
    "S_0", "I_0", "R_0"
  ),
  rinit=Csnippet("
    double m = pop/(S_0+I_0+R_0);
    S = nearbyint(m*S_0);
    I = nearbyint(m*I_0);
    R = nearbyint(m*R_0);
    cases = 0
    W = 0;"
  )
)

```

**Question 2.3.** Suppose you obtain the following error message when you build your `pomp` model using `Csnippets`.

```

Error: error in building shared-object library from C snippets: in 'Cbuilder': compilation error:
cannot compile shared-object library '/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_b675d99e691eda865610f570058ea3be.so': status = 1
compiler messages:
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
-I'/Users/ionides/Library/R/x86_64/4.1/library/pomp/include' -I'/Users/ionides/sbied/questions'
-I/usr/local/include -fPIC -Wall -g -O2
-c /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_b675d99e691eda865610f570058ea3be.c
-o /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_b675d99e691eda865610f570058ea3be.o
/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_b675d99e691eda865610f570058ea3be.c:33:16:
error: use of undeclared identifier 'pop'; did you mean 'pow'?
    double m = pop/(S_0+I_0+R_0);
                ^~~
                pow
/Applications/
In addition: Warning message:
In system2(command = R.home("bin/R"), args = c("CMD", "SHLIB", "-c", :
  running command 'PKG_CPPFLAGS="-I'/Users/ionides/Library/R/x86_64/4.1/library/pomp/include'
-I'/Users/ionides/sbied/questions'" '/Library/Frameworks/R.framework/Resources/bin/R' CMD SHLIB
-c -o /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_b675d99e691eda865610f570058ea3be.so
/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_b675d99e691eda865610f570058ea3be.c 2>&1' had status 1

```

Which of the following is a plausible cause for this error?

- (A) Using R syntax within a C function that has the same name as an R function.
- (B) A parameter is missing from the `paramnames` argument to `pomp`.
- (C) Indexing past the end of an array because C labels indices starting at 0.
- (D) Using `beta` as a parameter name when it is a declared C function.
- (E) A missing semicolon at the end of a line.

**Solution.** (B). The code generating this error is below. Here, `pop` is intended to be passed as a parameter, but it is missing from the `paramnames` argument. It could alternatively be defined as a global variable using the `globals` argument to `pomp`.

```

sir3 <- pomp(sir1, statenames=c("S", "I", "R", "cases", "W"),
  paramnames=c(
    "gamma", "mu", "iota",
    "beta1", "beta_sd", "rho",
    "S_0", "I_0", "R_0"
  ),
  rinit=Csnippet("
    double m = pop/(S_0+I_0+R_0);
    S = nearbyint(m*S_0);
    I = nearbyint(m*I_0);
    R = nearbyint(m*R_0);
    cases = 0
    W = 0;")

```

```
)  
)
```

**Question 2.4.** Suppose you obtain the following error message when you build your pomp model using Csnippets.

```
Error: error in building shared-object library from C snippets: in 'Cbuilder': compilation error:
cannot compile shared-object library '/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_77886fb66d95b4b9904440d86a4425b3.so': status = 1
compiler messages:
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
-I'/Users/ionides/Library/R/x86_64/4.1/library/pomp/include'
-I'/Users/ionides/sbied/questions' -I/usr/local/include -fPIC -Wall -g -O2
-c /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_77886fb66d95b4b9904440d86a4425b3.c
-o /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_77886fb66d95b4b9904440d86a4425b3.o
/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_77886fb66d95b4b9904440d86a4425b3.c:39:36:
error: too many arguments to function call, expected 2, have 3
    rep = nearbyint(rnorm(1,mean,sd));
                        ~~~~~~      ~
/Librar
In addition: Warning message:
In system2(command = R.home("bin/R"), args = c("CMD", "SHLIB", "-c", :
running command 'PKG_CPPFLAGS="-I'/Users/ionides/Library/R/x86_64/4.1/library/pomp/include'
-I'/Users/ionides/sbied/questions'" '/Library/Frameworks/R.framework/Resources/bin/R'
CMD SHLIB -c -o /var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_77886fb66d95b4b9904440d86a4425b3.so
/var/folders/fv/pt62sh2d6_gf9fp3t7b466vr0000gr/T//RtmpD16GmG/
5324/pomp_77886fb66d95b4b9904440d86a4425b3.c 2>&1' had status 1
```

Which of the following is a plausible cause for this error?

- (A) Using R syntax within a C function that has the same name as an R function.
- (B) A parameter is missing from the `paramnames` argument to `pomp`.
- (C) Indexing past the end of an array because C labels indices starting at 0.
- (D) Using `beta` as a parameter name when it is a declared C function.
- (E) A missing semicolon at the end of a line.

**Solution.** (A). The code producing the error is below. Within Csnippets, the C versions of R distribution functions are available but they have slightly different syntax from their more familiar R cousins.

```
sir4 <- pomp(sir1,statenames=c("S","I","R","cases","W"),
  paramnames=c(
    "gamma","mu","iota",
    "beta1","beta_sd","pop","rho",
    "S_0","I_0","R_0"
  ),
  rmeasure=Csnippet("
    double mean, sd;
    double rep;
```

```

    mean = cases*rho;
    sd = sqrt(cases*rho*(1-rho));
    rep = nearbyint(rnorm(1,mean,sd));
    reports = (rep > 0) ? rep : 0;"
  )
)

```

**Question 2.5.** Let  $V_n$  be a Markov process and let  $W_n = h(V_n)$  for some function  $h$ . Which of the following statements are true?

- i)  $W_n$  is a Markov process for all choices of  $h$ .
- ii)  $W_n$  is a Markov process for some choices of  $h$ .
- iii)  $W_n$  is not a Markov process for any choice of  $h$ .
- iv) If  $V_n = (X_n, Y_n)$  where  $X_n$  and  $Y_n$  are a POMP model, and  $h(X_n, Y_n) = X_n$  then  $W_n$  is a Markov process.
- v) If  $V_n = (X_n, Y_n)$  where  $X_n$  and  $Y_n$  are a POMP model, and  $h(X_n, Y_n) = Y_n$  then  $W_n$  is a Markov process.

- (A) i,iv,v
- (B) ii,iv
- (C) ii,v
- (D) iii
- (E) None of the above

**Solution.** (B)

### Questions on Chapter 3: Likelihood for POMP

**Question 3.1.** Suppose that 10 replications of a particle filter, each using  $10^3$  particles, runs in 15 minutes with no parallelization. To look for a more precise likelihood evaluation, you consider running 20 replicates, each with  $10^4$  particles. How many minutes will this take, if you distribute the calculation across 4 cores?

- (A) 50
- (B) 60
- (C) 75
- (D) 120
- (E) 300

**Solution.** (C). Using the linear dependence, also called proportionality, of the computing effort on various algorithmic parameters, we calculate

$$5 \times (10000/1000) \times (20/10) \times (1/4) = 75.$$

**Question 3.2.** A particle filter is repeated 5 times to evaluate the likelihood at a proposed maximum likelihood estimate, each time with  $10^4$  particles. Suppose the log likelihood estimates are  $-2446.0$ ,  $-2444.0$ ,  $-2443.0$ ,  $-2442.0$ ,  $-2440.0$ . Which of the following is an appropriate estimate for the log likelihood at this parameter value and its standard error.

- (A) Estimate =  $-2443.0$ , with standard error 1.0

- (B) Estimate =  $-2443.0$ , with standard error 2.2
- (C) Estimate =  $-2443.0$ , with standard error 5.0
- (D) Estimate =  $-2441.4$ , with standard error 2.2
- (E) Estimate =  $-2441.4$ , with standard error 1.4

**Solution.** (E). Answers (A), (B) and (C) estimate using a mean on the log scale. However, the particle filter provides an unbiased likelihood estimate on a natural scale but not on a log scale. Note that the particle filter also has some bias for most quantities on a natural scale, which reduces to zero as the number of particles tends to infinity, but it happens to be unbiased for the likelihood. The standard error for the log of the mean of the likelihoods can be computed by the delta method or a jack-knife, for example using the `logmeanexp` function in `pomp`.

```
ll <- c(-2446, -2444, -2443, -2442, -2440)
mean(ll)

[1] -2443

sd(ll)

[1] 2.236068

sd(ll)/sqrt(length(ll))

[1] 1

library(pomp)
logmeanexp(ll, se=TRUE)

              se
-2441.422198    1.380688
```

**Question 3.3.** What is the log likelihood (to the nearest unit) of the Dacca cholera data for the POMP model constructed in `pomp` via

```
d <- dacca(deltaI=0.08)
```

with cholera mortality rate 8% and other parameters fixed at the default values.

- (A) -3764
- (B) -3765
- (C) -3766
- (D) -3767
- (E) -3768

**Solution.** (A), calculated as follows:

```
d <- dacca(deltaI=0.08)
library(doParallel)
my_cores <- detectCores()
registerDoParallel(my_cores)
bake(file="Q9.rds", {
```

```

foreach(i=1:32,.combine=c) %dopar% {
  library(pomp)
  logLik(pfilter(d,Np=10000))
}
}) -> cholera_loglik
logmeanexp(cholera_loglik,se=TRUE)

              se
-3764.2030946   0.1502043

```

**Question 3.4.** Effective sample size (ESS) is one of the main tools for diagnosing the success of a particle filter. If you plot an object of class `pfilterd_pomp` (created by applying `pfilter` to a `pomp` object), the ESS is displayed. Suppose one or more time points have low ESS (say, less than 10) even when using a fairly large number of particles (say,  $10^4$ ). What is the proper interpretation?

- (A) There is a problem with data, perhaps an error recording an observation.
- (B) There is a problem with the model which means that it cannot explain something in the data.
- (C) The model and data have no major problems, but the model happens to be problematic for the particle filter algorithm.
- (D) At least one of (A), (B) and (C).
- (E) Either (A) or (B) or both, but not (C). If the model fits the data well, the particle filter is guaranteed to work well.

**Solution.** (D). An example of a situation where the model fits the data well, but filtering is hard, arises when the measurement error is small relative to the process noise. In this case, the particles are scattered by the process noise and very few of them are compatible with the data due to the precise measurement. Thus, almost all the particles must be discarded as unfeasible given the data, corresponding to a low ESS.

## Questions on Chapter 4: Inference via iterated filtering

**Question 4.1.** When carrying out inference by iterated particle filtering, the likelihood increases for the first 10 iterations or so, and then steadily decreases. Testing the inference procedure on simulated data, this does not happen and the likelihood increases steadily toward convergence. Which of the following is the best explanation for this?

- (A) One or more random walk standard deviation is too large.
- (B) One or more random walk standard deviations is too small.
- (C) The model is misspecified, so it does not fit the data adequately.
- (D) A combination of the parameters is weakly identified, leading to a ridge in the likelihood surface.
- (E) Too few particles are being used.

**Solution.** (C). All the other issues can cause inference problems, but likely would cause similar problems on simulated data.

When there is a reproducible and stable phenomenon of decreasing likelihood, it generally indicates that the unperturbed model is a worse fit to the data than the perturbed model. Recall that the likelihood calculated by iterated filtering at each iteration corresponds to the model with perturbed parameters rather than the actual postulated model with fixed parameters. If the perturbed model has higher likelihood, it may mean that the data are asking to have time-varying parameters. It may also be a signature of any other weakness in the model that can be somewhat accommodated by perturbing the parameters.

**Question 4.2.** People sometimes confuse likelihood profiles with likelihood slices. Suppose you read a figure which claims to construct a profile confidence interval for a parameter  $\rho$  in a POMP model with four unknown parameters. Which of the following confirms that the plot is, or is not, a properly constructed profile confidence interval. The code producing the plot is available to you as an Rmarkdown file.

- (A) The CI is constructed by obtaining the interval of rho values whose log likelihood is within 1.92 of the maximum on a smoothed curve of likelihood values plotted against  $\rho$ .
- (B) The code involves evaluation of the likelihood but not maximization.
- (C) The points along the  $\rho$  axis are not equally spaced.
- (D) The smoothed line shown in the plot is close to quadratic.
- (E) (A) and (D) together.

**Solution.** (B).

If the researchers calculate a sliced likelihood through the MLE and tell you it is a profile, but you are concerned they might have constructed a slice by mistake, it is hard to know without looking at the code. (A) is the proper construction of a profile if the points are maximizations over the remaining parameters for a range of fixed values of rho. However, if the code does not involve maximization over other parameters at each value of rho, it cannot be a proper profile. It could be a slice accidentally explained to be a profile, and with a confidence interval constructed as if it were a profile.

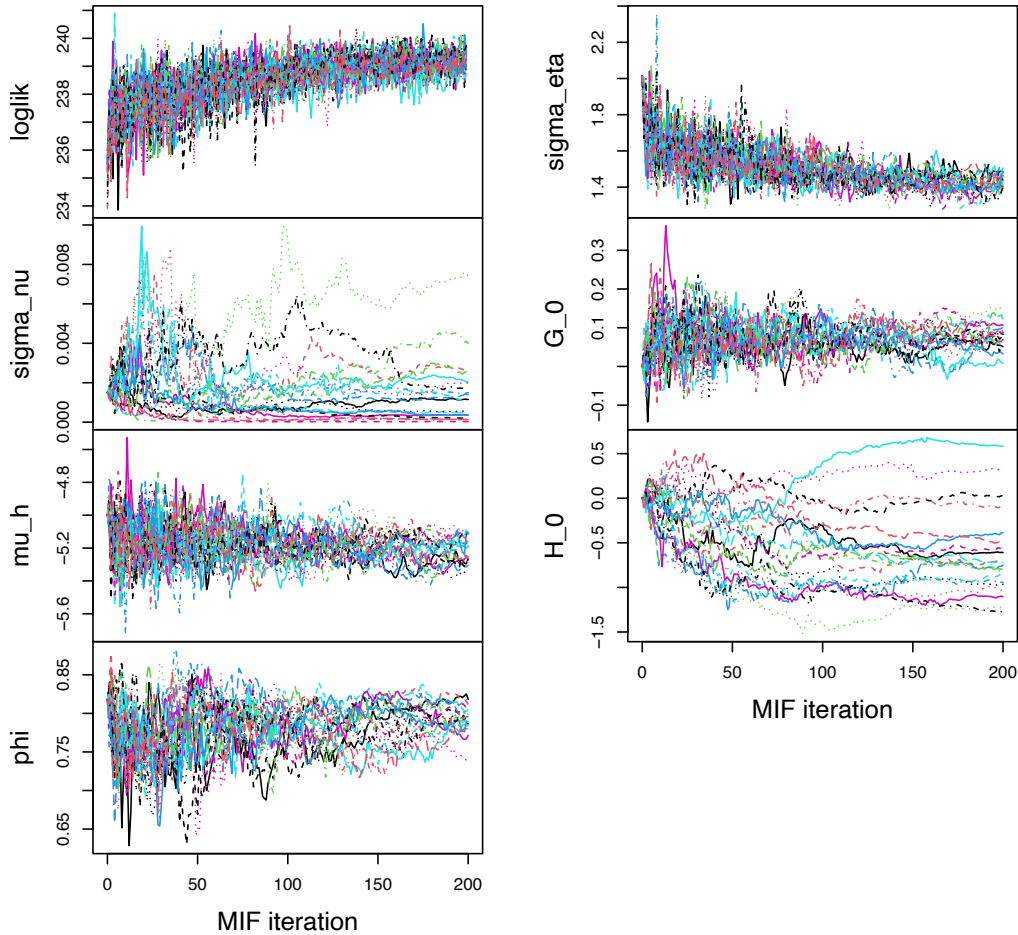
**Question 4.3.** Which of the following are true?

- (A) A profile likelihood must lie *above* every slice.
- (B) Confidence intervals can be read from likelihood slices.
- (C) A poor man's profile must lie above the true profile.
- (D) A poor man's profile must lie below the true profile.

**Solution.** (A) is true. (B) is not true in general. If our model depends on a single scalar parameter, then a slice and a profile are the same thing. (C) is false. (D) is true.



MIF2 convergence diagnostics



**Question 4.4.** The iterated filtering convergence diagnostics plot shown above come from a [student project](#). What is the best interpretation?

- (A) Everything seems to be working fine. The likelihood is climbing. The replicated searches are giving consistent runs. The spread of convergence points for  $\sigma_\nu$  and  $H_0$  indicates weak identifiability, which is a statistical fact worth noticing but not a weakness of the model.
- (B) The consistently climbing likelihood is promising, but the failure of  $\sigma_\nu$  and  $H_0$  to converge needs attention. Additional searching is needed, experimenting with **larger** values of the random walk perturbation standard deviation for these parameters to make sure the parameter space is properly searched.
- (C) The consistently climbing likelihood is promising, but the failure of  $\sigma_\nu$  and  $H_0$  to converge needs attention. Additional searching is needed, experimenting with **smaller** values of the random walk

perturbation standard deviation for these parameters to make sure the parameter space is properly searched.

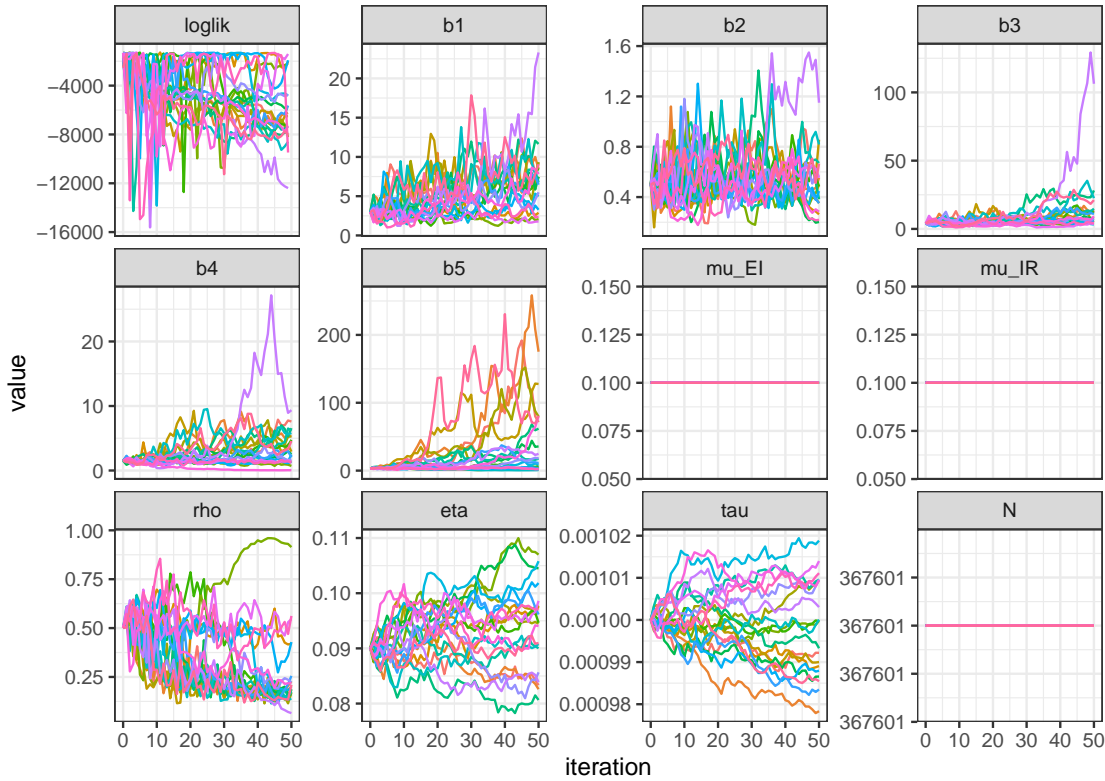
- (D) The consistently climbing likelihood is promising, but the failure of  $\sigma_\nu$  and  $H_0$  to converge needs attention. This indicates weak identifiability which cannot be solved by improving the searching algorithm. Instead, we should change the model, or fix one or more parameters at scientifically plausible values, to resolve the identifiability issue before proceeding.
- (E) Although the log likelihood seems to be climbing during the search, until the convergence problems with  $\sigma_\nu$  and  $H_0$  have been addressed we should not be confident about the successful optimization of the likelihood function or the other parameter estimates.

**Solution.** (A). All searches are finding parameters with consistent likelihood. The discrepancies of a few log likelihood units put the parameter values within statistical uncertainty according to Wilks's Theorem. Therefore, the spread in the parameter estimates reflects uncertainty about the parameter given the data, rather than a lack of convergence.

That perspective suggests that the goal of the Monte Carlo optimizer is to get close to the MLE, measured by likelihood, rather than to obtain it exactly. Independent Monte Carlo searches can be combined via a profile likelihood to get a more exact point estimate and a confidence interval.

Wide confidence intervals, also called weak identifiability, are not necessarily a problem for the scientific investigation. Some parameters may be imprecisely estimable, while others can be obtained more precisely, and part of the analysis is to find which is in each category. It may also be of interest to investigate what extra precision can be obtained on one parameter by making assumptions about the value of another, as in (D), but this is not mandatory for proper inference.

Overall, the convergence plots here look good. The plots show that the searches are all started from a single high likelihood starting point. Now this has been done successfully, a natural next step would be to start some searches from more diverse starting points to look for any global features missed by this local search.



**Question 4.5.** The iterated filtering convergence diagnostics plot shown above come from a [student project](#), calculated using  $10^3$  particles. What is the best interpretation?

- (A) Everything seems to be working fine. There is a clear consensus from the different searches concerning the highest likelihood that can be found. Therefore, the search is doing a good job of maximization. Occasional searches get lost, such as the purple line with a low likelihood, but that is not a problem.
- (B) The seaches obtain likelihood values spread over thousands of log units. We would like to see consistent convergence within a few log units. We should use more particles and/or more iterations to achieve this.
- (C) The seaches obtain likelihood values spread over thousands of log units. We would like to see consistent convergence within a few log units. We should compare the best likelihoods obtained with simple statistical models, such as an auto-regressive moving average model, to look for evidence of model misspecification.
- (D) The seaches obtain likelihood values spread over thousands of log units. We would like to see consistent convergence within a few log units. We should look at the effective sample size plot for the best fit we have found yet, to see whether there are problems with the particle filtering.
- (E) All of (B), (D) and (D).

**Solution.** (E). The authors of this project were able to show evidence of adequate global maximization for their model, but their maximized likelihood was 47 log units lower than ARMA model. The wide spread in likelihood, thousands of log units, shown in this convergence plot suggests that the numerics are not working smoothly. This could mean that more particles are needed:  $10^3$  particles is relatively low for a particle filter. However, if the model fit is not great (as revealed by comparison against a benchmark) this makes the filtering harder as well as less scientifically satisfactory. If the model is fitting substantially below ARMA benchmarks, it is worth considering some extra time on model development. Identifying time points with low effective sample size can help to identify which parts of the data are problematic for the model to explain.

## Questions on Chapter 5: Measles

**Question 5.1.** Two models are fitted to case counts on an epidemic. Model 1 is an SIR POMP model with a negative binomial measurement model, and model 2 is a linear regression model estimating a cubic trend. The log likelihoods are  $\ell_1 = -2037.91$  and  $\ell_2 = -2031.28$  respectively. Which of the following do you agree with most?

- (A) We should not compare the models using these likelihoods. They correspond to different model structures, so it is an apples-to-oranges comparison.
- (B) We can compare them, but the difference is in the 4th significant figure, so the likelihoods are statistically indistinguishable.
- (C) The linear model has a noticeably higher likelihood. Our mechanistic model needs to be updated to beat this benchmark before we can responsibly interpret the fitted model. If a simple regression model has higher likelihood than a more complex mechanistic model, one should prefer the simpler model.
- (D) The linear model has a noticeably higher likelihood. The mechanistic model is somewhat validated by being not too far behind the simple regression model. We are justified in cautiously interpreting the mechanistic model, while continuing to look for further improvements.
- (E) The log likelihoods cannot properly be compared as presented, but could be if we used a Gaussian measurement model for the POMP (or a negative binomial generalized linear model instead of least squares for the regression).

**Solution.** (D).

Why not (A)? Likelihoods of different models for the same data can be compared. Likelihood ratio tests using Wilks's theorem specifically require nested models, but in other contexts (such as AIC and the Neyman-Pearson lemma) the models being compared by likelihood do not need to have any particular relationship.

Why not (B)? Likelihood ratios have statistical meaning, which corresponds to differences of log likelihoods. The likelihood is a dimensional quantity, whereas the likelihood ratio is dimensionless. The units used correspond to a scientifically arbitrary additive constant to the log likelihood, which disappears after taking differences.

Why not (C)? If our only goal were to find a predictive model, then (C) could be a reasonable position. Usually, we want to find a model that also has interpretable structure, leading to understanding of the system or estimating the effect of interventions. A simple regression model cannot do those things, even if it fits a bit better. If the mechanistic model fits much worse than simple alternatives, it is not providing a reasonable explanation of the data, suggesting that there may be important things missing from the model specification. Quite likely, with some persistence, a mechanistic specification will beat a simple off-the-shelf statistical model.

**Question 5.2.** A compartment model is first implemented as a system of ordinary differential equations (ODEs). This leads to qualitatively reasonable trajectories, but poor likelihood values. The researchers add stochasticity in an attempt to improve the fit of the model by interpreting the ODEs as rates of a Markov chain. The likelihood, maximized by iterated particle filtering, remains poor compared to ARMA benchmarks. In addition, the effective sample size for the particle filtering is low at many time points despite even using as many as  $10^4$  particles. Which of the following is the most promising next step?

- (A) Increase to  $10^5$  particles, moving the computations to a cluster if necessary.
- (B) Add noise to one or more rates to allow for overdispersion.
- (C) Try adding extra features to the model to capture scientific details not present in the original model.
- (D) Experiment with variations in the iterated filtering procedure; maybe more iterations, or a different cooling schedule.
- (E) To address the possibility of reporting errors, see if the model fits better when the most problematic data points are removed.

**Solution.** (B). All the possibilities are worth consideration. However, adding noise in rates to give flexibility in mean-variance relationships is commonly an important part of developing a stochastic model. The simple compartment model interpretation of a ODE as a Markov chain is determined by the rates and therefore does not have free parameters to describe variance. There is some variance inherent in the Markov chain (demographic stochasticity) but additional variability may be needed. It will be hard to investigate the other possibilities if the model has not been given enough stochasticity to explain the variability in the data, so including overdispersion should be an early step. Note that overdispersion can be included in both the process model and the measurement model.

**Question 5.3.** You fit an SEIR model to case reports of an immunizing disease from a city. The resulting confidence interval for the mean latent period is 12–21 days, but clinical evidence points to a latent period averaging about 7 days. Which of the following is the most appropriate response to this discrepancy?

- (A) The latent period may be confounded with some unmodeled aspect of the system, such as spatial or age structure. The model estimates an effective latent period at the population level, which may not perfectly match what is happening at the scale of individuals. One should be cautious of making a causal interpretation of models fitted to observational data.
- (B) The discrepancy shows that something is substantially wrong with the model. Extra biological detail must be introduced with the goal of bringing the estimated parameter back in line with the known biology of the system.

- (C) The discrepancy is problematic, but fortunately can readily be fixed. Since we know the clinical value of this parameter with reasonable accuracy, we should simply use this value in the model rather than estimating it.
- (D) If the model fits the data statistically better than any known alternative model, then we have to take the estimated parameter at face value. It is certainly possible that the estimates in the literature correspond to some different population, or different strain, or have some other measurement bias such as corresponding to severe cases resulting in hospitalization. The discrepancy does not show that our model was wrong.
- (E) This discrepancy suggests that we should take advantage of both C and D above by putting a Bayesian prior on the latent period. By quantifying the degree of our skepticism about the previously established clinical value of 7 days, we can optimally combine that uncertainty with the evidence from this dataset.

**Solution.** (A).

Transferring parameter estimates between scales is hard. An example is the difficulty of reconciling micro and macro economics. It is generally not possible to guarantee that a parameter means exactly the same thing in models at different scales. (A) acknowledges this. The other answers, in various ways, assume that there should be a single parameter value that describes the system at all scales. We have some sympathy with (D), since it is reasonable to try to gain biological understanding by investigating why the fitted model is successful at explaining the data.