

# Homework 8, due 11:59pm Monday April 4

STATS/DATASCI 531, Winter 2022

## Solutions

**This is a group assignment for which you will collaborate with your final project group.** You should submit one homework report for your group. This report should be written in Rmd, and you should submit both the Rmd source and a version compiled to HTML. For each question, you should give an answer (a letter or true/false claims) followed by some text explaining your answer. Most of the questions do not need any coding, but you can show your code when appropriate.

The goal is to think about some issues that might arise in final projects, and indeed some of the questions alert you to common issues arising in previous 531 final projects.

If you poke around, you may find solutions to these questions. That is not against the rules, following the usual source attribution requirements for the course that all sources are acceptable as long as they are properly acknowledged. The requirement, as in other homeworks, is that you explicitly list all sources you consulted and you make clear what you learned from them. For group work, it is your responsibility to make sure that every source consulted by every group member is listed at the end of the report and referenced properly where used within the report.

It may be simplest for this homework if you avoid consulting solutions online. If you do obtain answers online, you should find ways to go beyond the sources to make your own contribution.

These questions are based on material developed for a course on [Simulation-based Inference for Epidemiological Dynamics](#). Note that this vague acknowledgement of a large but unspecified intellectual debt is not intended as a role model for homework reports; it is okay in this context but insufficient to explain your own contribution in a course assignment.

---

**Question 1.** From a scientific perspective, conclusions should not depend on the units we choose. However, we must get the details straight to correctly describe a POMP model and its `pomp` representation. Suppose our data are two years of weekly aggregated case reports of a disease and we have a continuous time model solved numerically using an Euler timestep of size  $dt$ . Which one 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.** 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 one of the following is the most 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 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 one of the following is the most 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 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 one of the following is the most 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 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.**

**Question 6.** 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 7.** 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`.

```
l1 <- c(-2446, -2444, -2443, -2442, -2440)
mean(l1)
[1] -2443
sd(l1)
[1] 2.236068
sd(l1)/sqrt(length(l1))
[1] 1
library(pomp)
logmeanexp(l1, se=TRUE)
              se
-2441.422198    1.380688
```

**Question 8.** 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
-3764.2247020  0.1251129
```

**Question 9.** 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.

**Question 10.** 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 one 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 11.** 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. Suppose that the code producing the plot is available to you as an Rmarkdown file. Which one of the following confirms that the plot is, or is not, a properly constructed profile confidence interval.

- (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) Both 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 12.** For each of the following, say whether the statement is true or false.

- (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

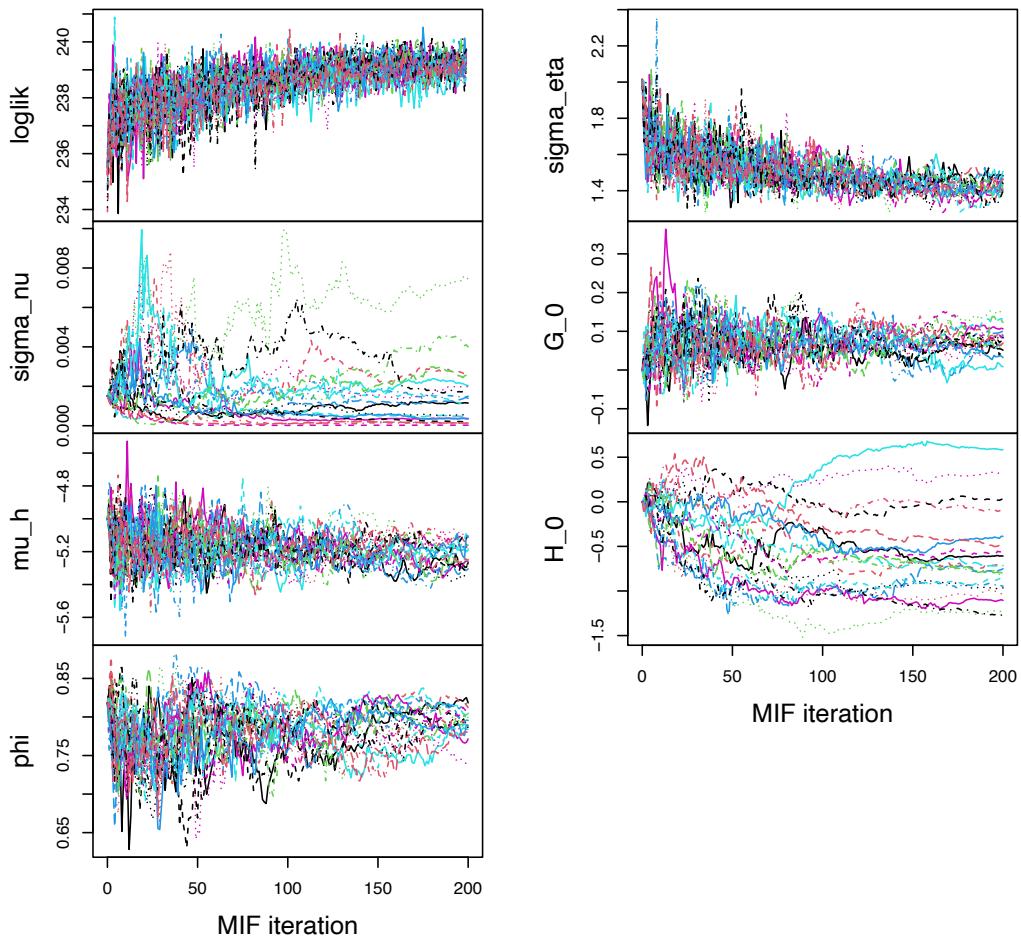


Figure 1: Iterated filtering convergence diagnostic plot for [Question 13](#)

**Question 13.** The iterated filtering convergence diagnostics plot in Fig. 1 comes 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 14.** The iterated filtering convergence diagnostics plot in Fig. 2 comes from a [student project](#), calculated using  $10^3$  particles. Which one of the following is the best interpretation of this diagnostic plot?

- (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 searches 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.

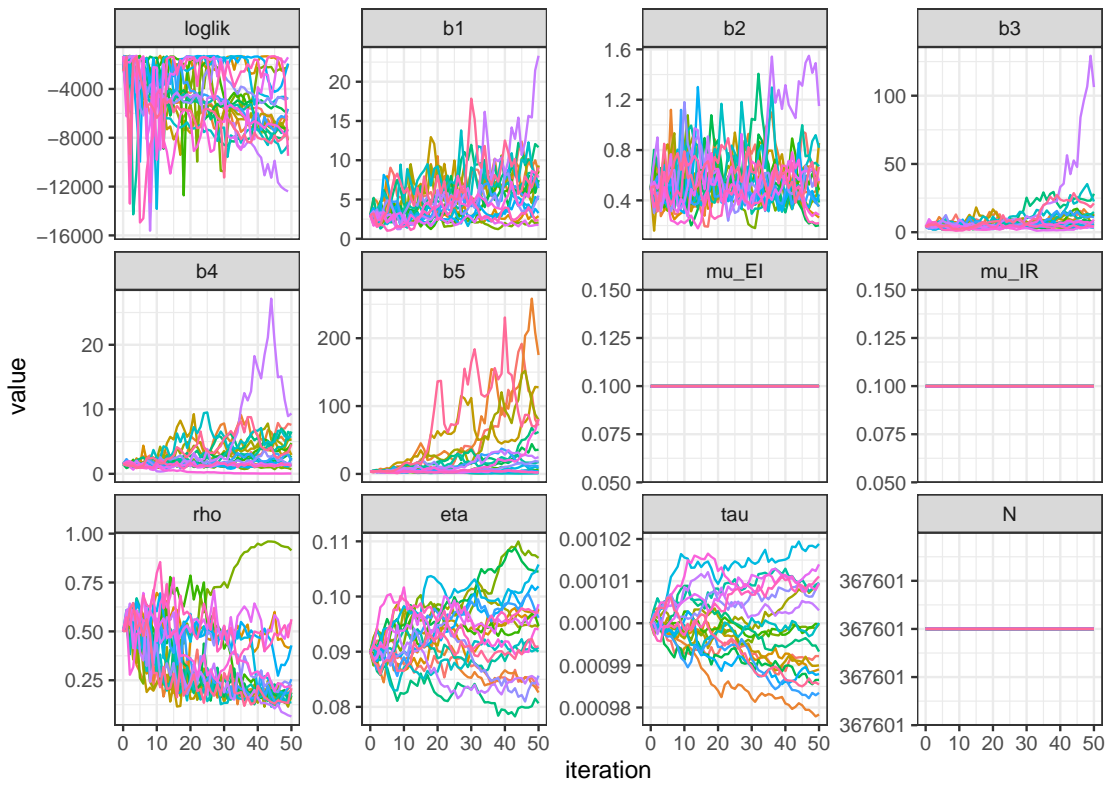


Figure 2: Iterated filtering convergence diagnostic plot for [Question 14](#)

- (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](#), [C](#) 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.

**Question 15.** In the following call to `mif2`, which of the statements below are true? You may assume that `obj` is a pomp object with parameters `alpha`, `Beta`, `gamma`, and `delta`.

```
obj %>%
  mif2(
    Nmif=100,
    partrans=parameter_trans(log=c("Beta","alpha","delta")),
    paramnames=c("Beta","alpha","delta"),
    rw.sd=rw.sd(Beta=0.05,alpha=ivp(0.02),gamma=0.05),
    cooling.fraction.50=0.1
  ) -> obj
```

- (A) 50 IF2 iterations will be performed.
- (B) `Beta` and `alpha` are estimated on the log scale.
- (C) `gamma` is not estimated.
- (D) `delta` is not estimated.
- (E) The magnitude of the perturbation on `Beta` at the end of the run will be  $0.05 \times 0.1^{100} = 5 \times 10^{-102}$ .
- (F) The magnitude of the perturbation on `gamma` at the end of the run will be  $0.05 \times 0.1^{100/50} = 5 \times 10^{-4}$ .
- (G) `alpha` is an initial-value parameter; it will be perturbed only at the beginning of the time series.
- (H) After the call, `obj` is an object of class 'mif2d.pomp'.

**Solution.** A is false; 100 iterations will be performed. B is true. C is false. Since a random-walk sd is provided for `gamma`, it will be estimated. It will be estimated on the natural scale, since no transformation is given. D is true. Although it is (unnecessarily) transformed, `delta` will receive no perturbations and will thus remain fixed at whatever value it has to begin with. E is false. F is true. G is true. H is true.

**Question 16.** Assume that `obj` is the result of the call in [Question 15](#), we consider the result of the following call.

```
obj %>%
  mif2(
    rw.sd=rw.sd(Beta=0.05,alpha=ivp(0.02)),
    cooling.fraction.50=0.2
  )
```

Explain whether each of the following are true or false.

- (A) 100 more IF2 iterations will be performed.
- (B) The settings of the previous calculation are re-used, with the exception of `rw.sd` and `cooling.fraction.50`.
- (C) The starting point of the new calculation is the end point of the old one.
- (D) `Beta` and `alpha` are estimated on the log scale.
- (E) `gamma` is not estimated.
- (F) `delta` is not estimated.
- (G) The cooling occurs more quickly than in the previous call.

**Solution.** A is true. B is true. C is true. D is true. The parameter transformations supplied in the earlier call are preserved. E is true. F is true. G is false. After 100 iterations, the perturbations are smaller than they were at the outset, by a factor of  $0.2^{100/50} = 0.04$ .