

# Semester Project - Weeks 9-10 Report

Alexia Paratte

May 2, 2024

## Goals

After the midterm presentation which happened on the 15th of April, here are the remarks made during the presentation:

- Report 3 significant digits when reporting numbers
- Turn off second derivative in Biogeme: we wanted to try going from 100% to 0%, however Bierlaire suggested to just turn it off to 0. This should considerably optimize the estimation time
- Do not change the default optimization algorithm: turning off the second derivative should be enough

For the remaining weeks, here is a list of tasks we want to further explore:

- Document comments about the [Biogeme Documentation Website](#)
- Look at the features of Apollo that are not available in Biogeme: To do so, should compare the documentation
- Keep looking at Models using the Monte-Carlos and Halton Draws: can see the `triangular_generator_function` (or define it as we wish) to ensure you use the same draws with Apollo and Biogeme
- Look at post estimation analysis (Chapter 9 in Apollo Documentation) (and look at examples on the Biogeme website to directly see what we can do post est.)

The imminent goal for this week is to try a Model from Biogeme and translate it to Apollo, to avoid mistakes in translation. After translation, we want to import the draws generated in Apollo to Biogeme, to make sure these are the same. We also want to look at Biogeme website along the way, and notify anything that could be better. As for the documentation, we want to make a list of what Biogeme and Apollo can do post-estimation.

# Model exploration

## 1. Construction of the script in Biogeme and translation to Apollo

The Model choosen in Biogeme is a simple MMNL, whose dataset is explained [here](#). We will call it **swissmetro**. For now, we want to only have one random component, which is generated with Halton Draws. We will try different number of draws, as to see how the optimisation times in Biogeme and Apollo change.

The original script can be found [here](#). The **swissmetro** Model utilities are given as follows:

$$\begin{aligned} V_{\text{train}} &= ASC_{\text{train}} + \beta_{\text{time\_rnd}} \cdot \text{train\_tt\_scaled} + \beta_{\text{cost}} \cdot \text{train\_cost\_scaled} \\ V_{\text{sm}} &= ASC_{\text{sm}} + \beta_{\text{time\_rnd}} \cdot \text{sm\_tt\_scaled} + \beta_{\text{cost}} \cdot \text{sm\_cost\_scaled} \\ V_{\text{car}} &= ASC_{\text{car}} + \beta_{\text{time\_rnd}} \cdot \text{car\_tt\_scaled} + \beta_{\text{cost}} \cdot \text{car\_cost\_scaled} \end{aligned}$$

The original dataset is modified, to have scaled variables. Here are the model parameters:

Parameter	Starting value	Estimation needed
$ASC_{\text{car}}$	0	✓
$ASC_{\text{train}}$	0	✓
$ASC_{\text{sm}}$	0	×
$\beta_{\text{cost}}$	0	✓
$\beta_{\text{time}}$	0	✓
$\beta_{\text{time\_s}}$	1	✓

Table 1: **Swissmetro** model parameters

The randomness is brought using normally distributed Haltons draws (with base 3):

$$\beta_{\text{time\_rnd}} = \beta_{\text{time}} + \beta_{\text{time\_s}} \cdot \text{bioDraws}(\text{'B\_TIME\_RND'}, \text{'NORMAL\_HALTON3'})$$

The script being done in Biogeme, it is now time to translate it to Apollo. We need to be careful about the following, to avoid any mistake in translation:

- Defining the new variables the same way as in Biogeme
- Defining the parameters and utility functions correctly
- Make sure the availabilites are taken into account in both cases
- Using the same draws and same number of draws

When translation to Apollo, it has appears to me that Apollo does not seem to be able to work **without panel data**. In the original Biogeme script, the dataset is not modified into panel data.

Therefore, I have modified the code in Biogeme to match the panel data used in Apollo. (Read documentation p.22: *"panelData: A boolean variable indicating whether the data is to be treated as panel data. This is set automatically to TRUE if multiple observations are present per individual, and FALSE otherwise. If a user sets this to FALSE in the presence of multiple observations per individual, the data will be treated as cross-sectional."*) Also note that the second derivative in Biogeme was turned to 0, as suggested by M.Bierlaire.

## 2. Model Estimation

The model estimation is done using simple bounds algorithm in Biogeme, BGW in Apollo. At first, we draw 100 Haltons. The estimations results as well as the model statistics can be seen in Tables 2 and 3 respectively. We observe that for a 100 draws, the estimated parameters are almost equal. The statistics show that the models in both Apollo and Biogeme are equivalent.

Parameter	Value in Apollo	Value in Biogeme
$ASC_{car}$	0.2807	0.284
$ASC_{train}$	-0.572	-0.565
$ASC_{sm}$	0	0
$\beta_{cost}$	-1.650	-1.648
$\beta_{time}$	-3.165	-3.194
$\beta_{time\_s}$	3.718	3.706

Table 2: Final **Swissmetro** model estimation results for 100 draws

Statistic	Value in Apollo	Value in Biogeme
Number of estimated parameters	5	5
Number of threads	8	8
Sample size	752	752
Observation	6768	6768
Initial log likelihood	-5782.84	-4363.15
Final log likelihood	-4363.05	-4362.97
AIC	8736.11	8735.94
BIC	8770.21	8759.06
Time of estimation (hh:mm:ss)	00:00:43	0:01:51

Table 3: Final **Swissmetro** statistics for 100 draws

## 3. Number of draws

Now that we know that the models are equivalent, we want to try different number of draws. The goal of this experiment is to see how the estimation time gap between Biogeme and Apollo changes as the number of draws increases. At first, when changing the number of draws in Apollo, a problem occurs. Indeed, when going from 100 draws to 500, the model output for diagnostics

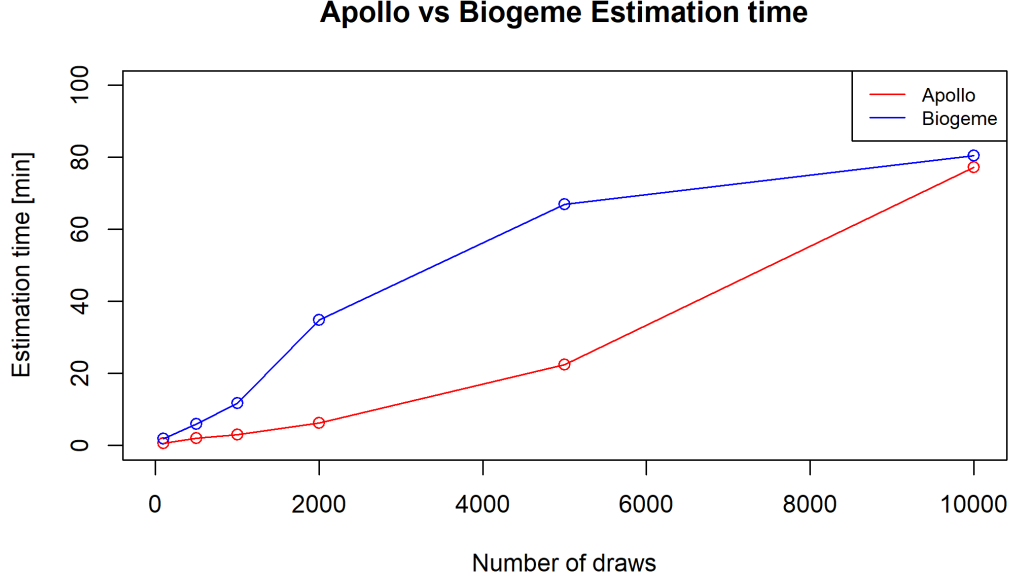


Figure 1: Estimation time plot for 100, 500, 1'000, 2'000, 5'000 and 10'000 draws, over one run

is "False convergence". This seems a bit odd, we thus decide to go from the BGW optimisation algorithm (which is the default one in Apollo) to the BFGS optimisation algorithm. We therefore get two versions of the model in Apollo, `swissmetro` and `swissmetro.bfgs`. This being done, it is time to experiment with the number of draws.

We report in Table 4 the exact estimation times in both Apollo and Biogeme, for 100, 500, 1'000, 2'000, 5'000 and 10'000 draws. This is also reported in Figure 1 for better visualisation.

Number of draws	Estimation Time Apollo	Estimation Time Biogeme
100	00:00:36	00:01:52
500	00:02:00	00:06:04
1'000	00:03:01	00:11:44
2'000	00:06:16	00:34:53
5'000	00:22:24	01:06:58
10'000	01:17:16	01:20:22

Table 4: Estimation time for different number of draws, over one run

At first, the estimation time gap between Apollo and Biogeme seems quite small. However, with the number of draws growing, the gaps widens up to 44 minutes in the 5'000 draws case. However, the gap seems to tighten up again at 10'000. To further explore this, it would be wise to run the scripts multiple times and do a median of the estimation time, and try draws between 5'000 and 10'000 (to see where it the gap narrows down) and after 10'000 (to see if it keeps narrowing).

## 4. Exportation/Importation of the Random Coefficient

Because the models in both Apollo and Biogeme were equivalent, and for a question of time, the exportation/importation of the random coefficient was let down for now.

## 5. Summary and further exploring

The summary of these two weeks exploration is as follows:

- We scripted a MMNL Model called **swissmetro** in Biogeme, then translated it to Apollo. The default optimisation algorithm are simple bounds in Biogeme and BGW in Apollo.
- This model has one random coefficient, which is generated using Haltons draws (in base 3) and Monte-Carlo estimation.
- Because of an error encountered in Apollo, the default algorithm was changed to a BFGS algorithm (only in Apollo).
- The final scripts in Biogeme and Apollo are respectively [swissmetro](#) and [swissmetro\\_bfgs](#).
- We did different number of draws (100, 500, 1'000, 2'000, 5'000, 10'000) and observed that the estimation time gap between Apollo and Biogeme first widens, then narrows down.
- The importation/exporation of random coefficients from Apollo to Biogeme was let down for now.

These are the points I would like to further explore, to deepen the research:

- To see why the gap widens then narrows down, we could: try other numbers of draws (between 5'000 and 10'000 and after 10'000), and do a median over multiple runs (as to deal with outliers).
- Analyze how the final loglikelihood changes as the number of draw increases
- Try to see how Biogeme and Apollo deals with a new random coefficient (to have 2 instead of only 1).
- The Pre/Post Estimation Features needs to be explored. It was supposed to be done this week, but because of time requirement for the run, I did not have the time

## Questions/Remarks

- Setting the seed in Apollo does not change the random coefficient: seem like it is always the same coefficients that are generated
- Website: [here](#), see a long list of things we can do with the biogeme expression. Would be nice to have a summary at the beginning of the page (values, draws, integration, etc)
- Want to understand why BGW has false convergence error