1 Summary of Simulation Lessons

My two main MLE approaches to estimating parameter values on simulated so far have been:

1. Routines: using optimization routines to automatically find the maximizing parameter values of the Log-Likelihood function
2. Manual Plotting: plotting the Log-Likelihood function to manually find the maximum

For approach (1), I have started to document how well the optimization routines do on simulated data. The issue we were facing with the real estimation is one of the parameters being sent to zero to maximize the likelihood function. So I want to pay special attention to parameter values close to zero compared to the other parameter. In Section 2, I document the first through pass at examining how routines perform when estimating parameters from simulated data where we know the "true" parameter values. I realized that I need to know the actual likelihood-maximizing parameter values in addition to the "true" parameter values, which may be different due to sampling variation (see Section 2.1.3 for more info on this).

Approach (2) serves as a brute-force check on the automatic routines in approach (1). My goal is to understand and document how close the routines come to estimating the likelihood-maximizing parameter values and how close those are to the "true" underlying DGP parameters. I document this brute force approach in Section 3.

Documenting these issues over a range of different parameter values and estimation hyperparameters is becoming a larger computational project, which is why I am moving some work to the Savio computation cluster. The below sections

2 Approach 1: Optimization Routines and Examining Bias

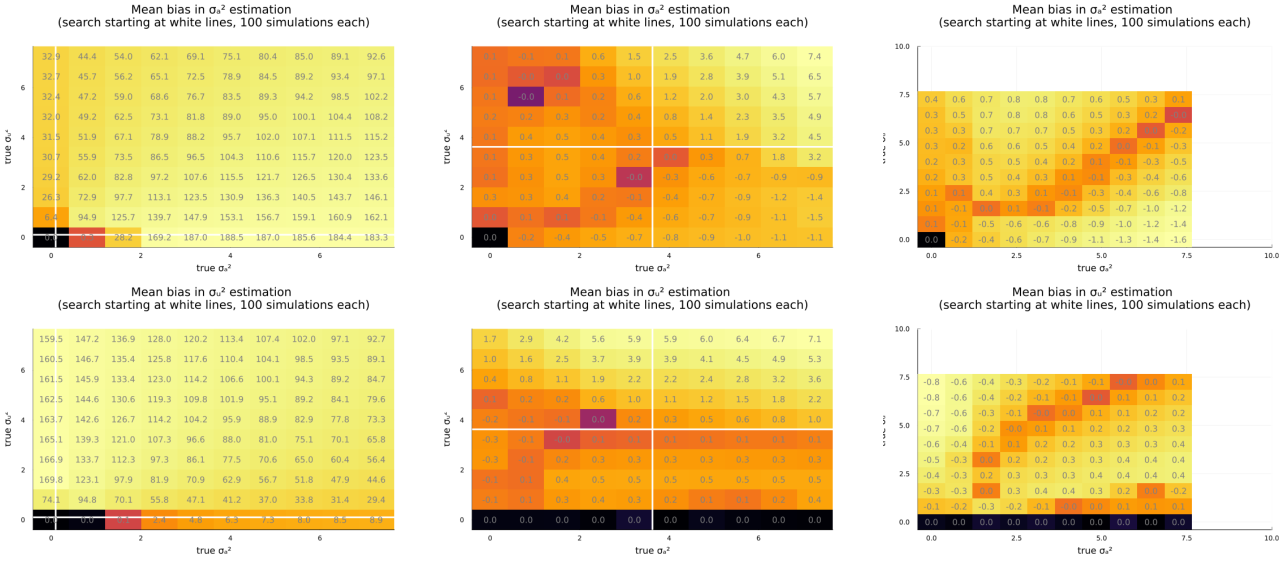
Note before getting too far: This section deals with results from using the default optimization routines. I have examined the loglikelihood function itself in the past to manually evaluate what I think is the maximizing parameter values, which sometimes differs from the estimates from the routines. See Section 3 for more explanation of the manual approach and lessons learned. The manual approach is also needed to further examine the sample variation issues I discuss in Section 2.1.3.

I generated simulations and MLE estimates with the following structure:

- setting all $\beta_k = 0$ and $b_{0i} = 0$ (to focus on the MLE of the error structure)
- an $N_\sigma \times N_\sigma$ grid of "true" $(\sigma_\alpha^2, \sigma_\mu^2)$ pairs ranging from 0.0001 to $2\sigma_{base}^2$
- N_{sim} simulated data sets for each $(\sigma_\alpha^2, \sigma_\mu^2)$ pair. Each dataset is the same size as our sample $(N = 4, T = 60)$ using the DGP in the paper.
- N_{search} different starting search values for the optimization routines (which means N_{search} different MLE's for each simulated dataset). The starting value is a location in $(\sigma_\alpha^2, \sigma_\mu^2)$ space to start the maximization search.
- ML estimates $(\hat{\sigma}_\alpha^2, \hat{\sigma}_\mu^2)$ for each simulated dataset and each starting value, based on the optimization routine (these may not be the true maximizing values).
- ML estimates' mean bias and mean absolute bias for each true $(\sigma_\alpha^2, \sigma_\mu^2)$ pair and each starting value. The means are calculated over the N_{sim} simulations.

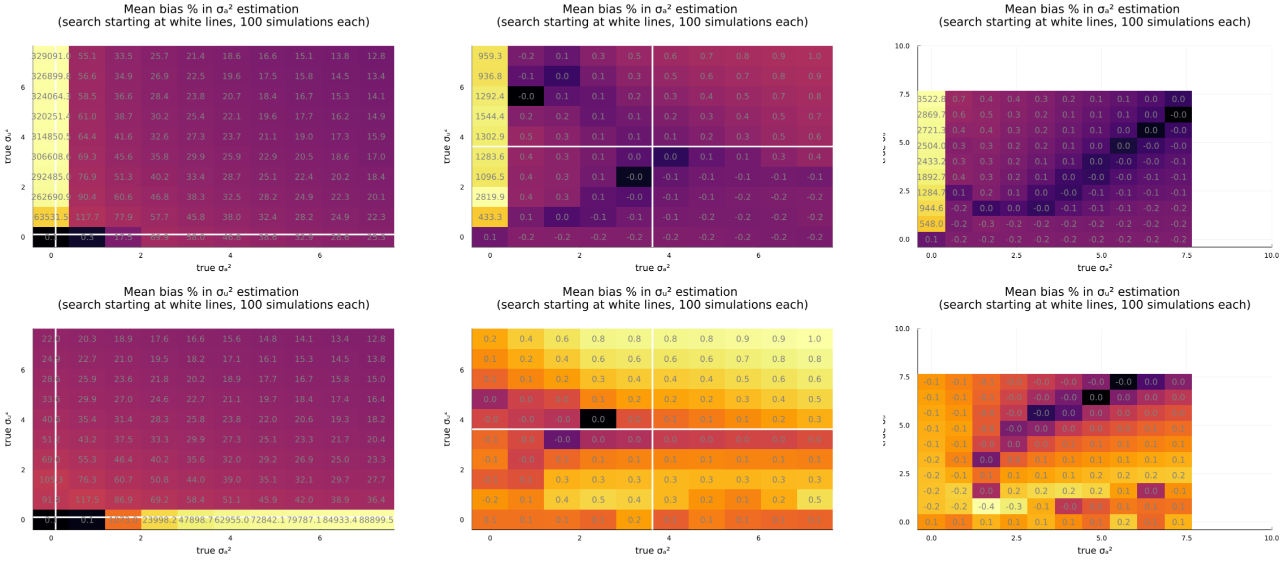
Because the sample is small, I wanted to generate enough simulated datasets for each "true" $(\sigma_\alpha^2, \sigma_\mu^2)$ pair to get a sense of the distribution of ML estimates. 100 simulations for each pair seems to be enough to start with. To examine the problem without taking too much computational time, I started with $N_\sigma = 10$, $N_{sim} = 100$, $N_{search} = 3$. I started with $\sigma_{base}^2 \approx 3.63$, which has been our best estimate of σ_μ^2 from the real data.

This produces $N_\sigma \times N_\sigma \times N_{sim} \times N_{search} = 30,000$ different MLE's. I took the MLE mean bias over the N_{sim} datasets, so have $N_\sigma \times N_\sigma \times N_{search} = 300$ mean bias heatmaps I want to show. Because I want to explore the bias in both σ_α^2 and σ_μ^2, each map figure contains two maps, one for each of the σ^2's. Let's review the figures and then discuss what that tells us we need to explore more.



(a) Search start: $(\sigma_\alpha^2, \sigma_\mu^2) = (0.1, 0.1)$ (b) Search start: $(\sigma_\alpha^2, \sigma_\mu^2) = (3.63, 3.63)$ (c) Search start: $(\sigma_\alpha^2, \sigma_\mu^2) = (10, 10)$

Figure 1: (Large figures in appendix) Mean bias in the MLE estimates for each pair of "true" $(\sigma_\alpha^2, \sigma_\mu^2)$, mean calculated over 100 simulations. Each of the three figures are results from using three different starting locations for the optimization search, increasing from left to right. In each figure, the top heatmap is for bias in $\hat{\sigma}_\alpha^2$ and the bottom is for bias in $\hat{\sigma}_\mu^2$. Notice that the magnitudes of the bias are much larger in the left maps than the right maps, suggesting that the algorithms do better at finding the maximum of the loglikelihood function when starting from above, rather than below. But staring from 10 (right) is not universally better than starting from 3.63 (middle) for all true parameter values, showing that the algorithms still perform better when the starting value is closer to the maximizing value.



(a) Search start: $(\sigma_\alpha^2, \sigma_\mu^2) = (0.1, 0.1)$ (b) Search start: $(\sigma_\alpha^2, \sigma_\mu^2) = (3.63, 3.63)$ (c) Search start: $(\sigma_\alpha^2, \sigma_\mu^2) = (10, 10)$

Figure 2: (Large figures in appendix) Mean percentage bias in the MLE estimates for each pair of "true" $(\sigma_\alpha^2, \sigma_\mu^2)$, mean calculated over 100 simulations, as a percentage of the true parameter value. Here, percentage is displayed as a fraction so 80% $= 0.8$. This is the same data as in figure 1 but percentages. **If we look at the top middle plot, the top left grid square can be interpreted as: the routine $\hat{\sigma}_\alpha^2$ is on average 959 times the true σ_α^2 used to generate the data in the 100 simulations underlying that grid square.** The top left plot is even more dramatic: the top left grid square says: $\hat{\sigma}_\alpha^2 \approx 329,091 \times \sigma_\alpha^2$.

2.1 Discussion & Next Steps

The bias seems to depend a lot on the distance between the true sigmas that generated the data and the starting sigma values used for the maximization search. The starting search values are given by the white cross hairs in the heatmaps. The heatmap results are all from the default gradient decent optimization algorithm; however, the dependence on the starting value also led me to try to document how different optimization algorithms compare – some seem better than others at finding maximums far away from the starting value.

The bias maps helped me think about four important features of the estimation problem:

1. Where you start the search for the optimization routine matters
2. Which optimization routine we use matters (and may alleviate point (1))
3. The small size of the sample means sample variation will lead to the likelihood being maximized at a location different from the true DGP parameter values.
4. I need a way to tell if the routines are finding the true maximum of the loglikelihood function and how far that maximum is from the true DGP parameter values.

2.1.1 Multi-start optimization

To address (1), it's common to use "multi-start" algorithms that start from several different locations in the parameter space and generate several estimates. They generally report the final estimates that result in the largest objective function value.

2.1.2 Algorithm choice

Another important hyperparameter to address (1) is optimization algorithm choice. I have been testing all the available alogrithm, but need a more thorough testing procedure to say anything useful. I plan to use the list of algorithms that use the gradient as another layer of iteration in the simulations. I plan to do this on the Savio computing cluster.

One limiting factor of our algorithm choice is the Hessian. Some algorithms can use the curvature information in the Hessian to better optimize, but they require the analytically derived hessian. Perhaps that's an area for future work.

2.1.3 Sample variation and Small sample issues

The small size of the sample being considered (N=4, T=60) leads to questions regarding sample variation. When generating data for simulations, we know the "true" σ_α^2 and σ_μ^2 used to generate the data. However, because of sample variation, the small sample may be more consistent with other parameter values than the ones used to generate the data. This means that the likelihood function may have a maximum that is not at the underlying parameter values. I would like to know more about the difference between bias in the parameter estimates caused by sample variation vs bias caused by the estimation routine.

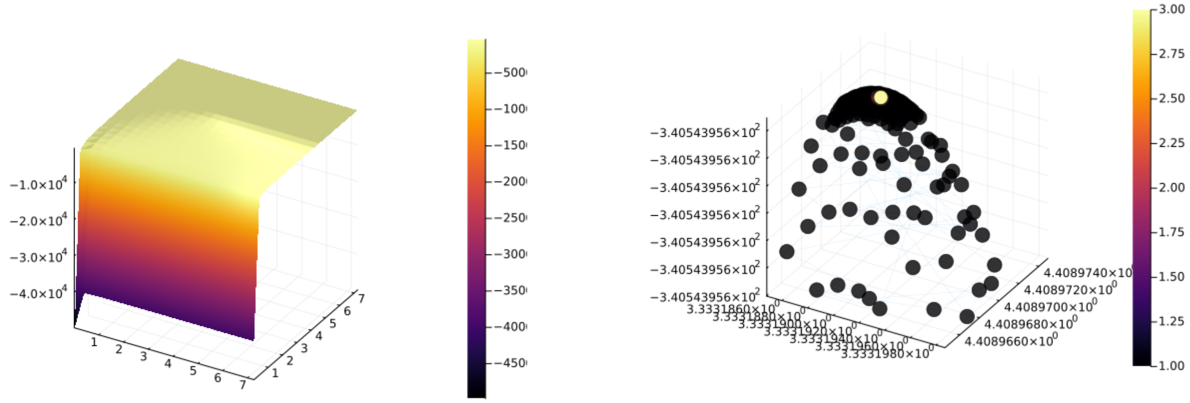
To separate these two forms of bias, I would like to have a way to know where the true maximum of the loglikelihood (LL) function is. The location of the true maximum as compared to the true DGP parameter values will tell me about bias due to sample variation. Then, knowing the true maximum of the LL function, I can see how close the optimization routines come. I have developed a more intensive and brute force search technique to find the maximum of the likelihood function. I wanted to be very sure that I knew the real maximum and was able to plot it.

3 Approach 2: Brute-force Evaluation of the Log-Likelihood Function

After examining some simulations, I realized that the optimization routines that come standard may not be stopping prematurely. However due to sampling variation and the small sample size we are dealing with, I was not sure how close the likelihood-maximizing parameter values were to the "true" DGP parameter values. I wanted to be sure about where the likelihood function was maximized, so I developed an algorithm that starts evaluating the LL function on a coarse grid of points, then iteratively zooms into the maximizing area. Because I am relatively confident in the LL function's smoothness in simulated data, I am not too worried about missing the maximum in the coarse grid.

I iteratively zoom in on the maximum until the grid passes some preset coarseness threshold, then return the maximal point as the ML estimate. It's hard to tell in the image below, but the right image has a very high density of evaluated points near the top of the likelihood function. It takes a relatively long time to evaluate the likelihood function, which means this brute-force search method takes much longer than the optimization routines. I plan to use the estimates from this brute force method to evaluate how the different routines and different starting values perform in finding the maximizing values. Then I will eventually choose one algorithm and starting value to estimate the on the real data before moving onto GMM estimation.

Note: These are placeholders for more useful images. Sorry for the poor quality of plots here, I just want to explain the process I'm using.



(a) Plotting the LL function over a wide domain

(b) Search algorithm: zooming into the maximum. The z-axis is plotted in log units to aid with visualization of the maximum.

Figure 3: (Large figures in appendix) The algorithm iteratively zooms in on the likelihood-maximizing point until it reaches the preset grid coarseness threshold.

4 Using Approach 2 to Evaluate Approach 1

I will use the brute-force estimates of the LL maximum to evaluate the canned optimization routines used in Approach 1.

[Add evaluation statistics/plots here]

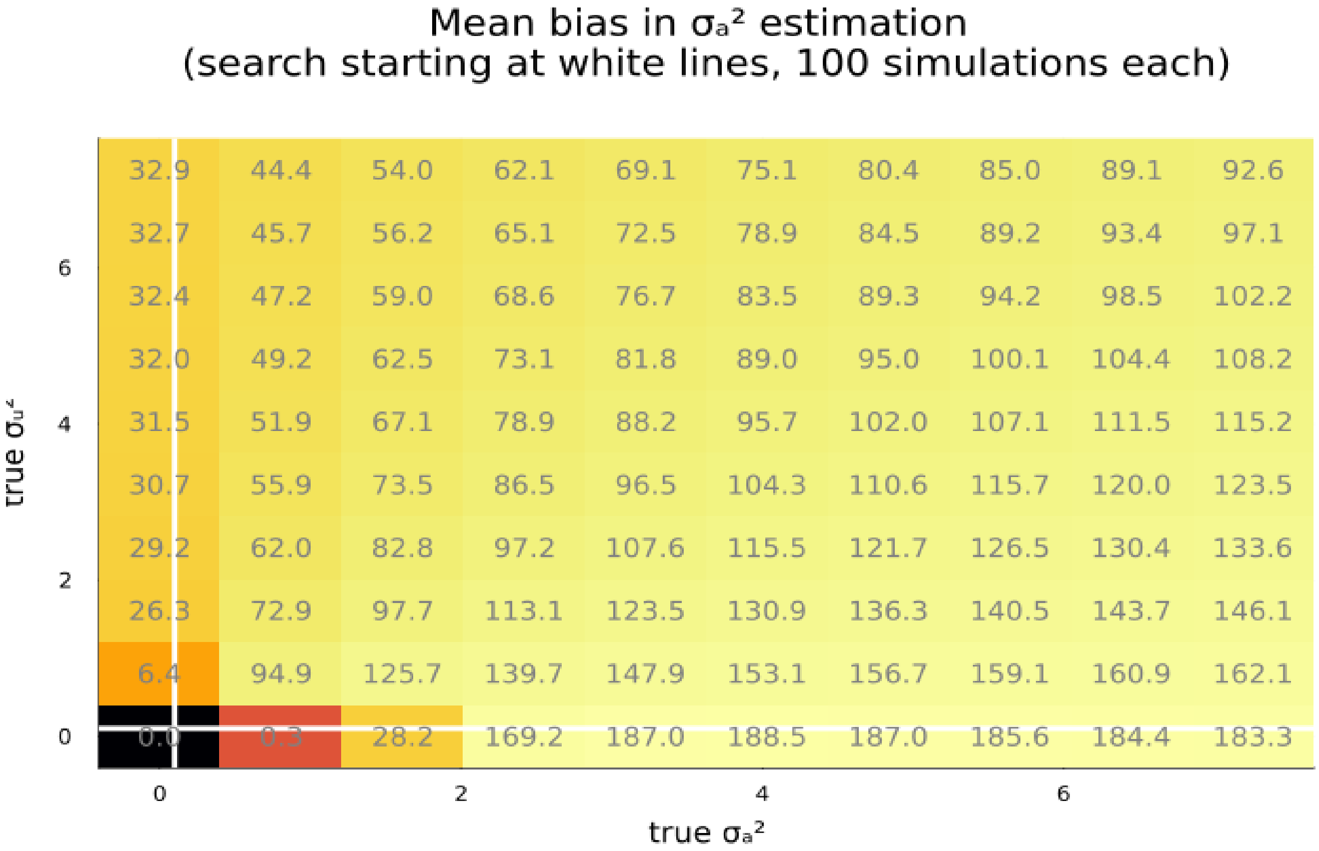
5 Modified Approach 1

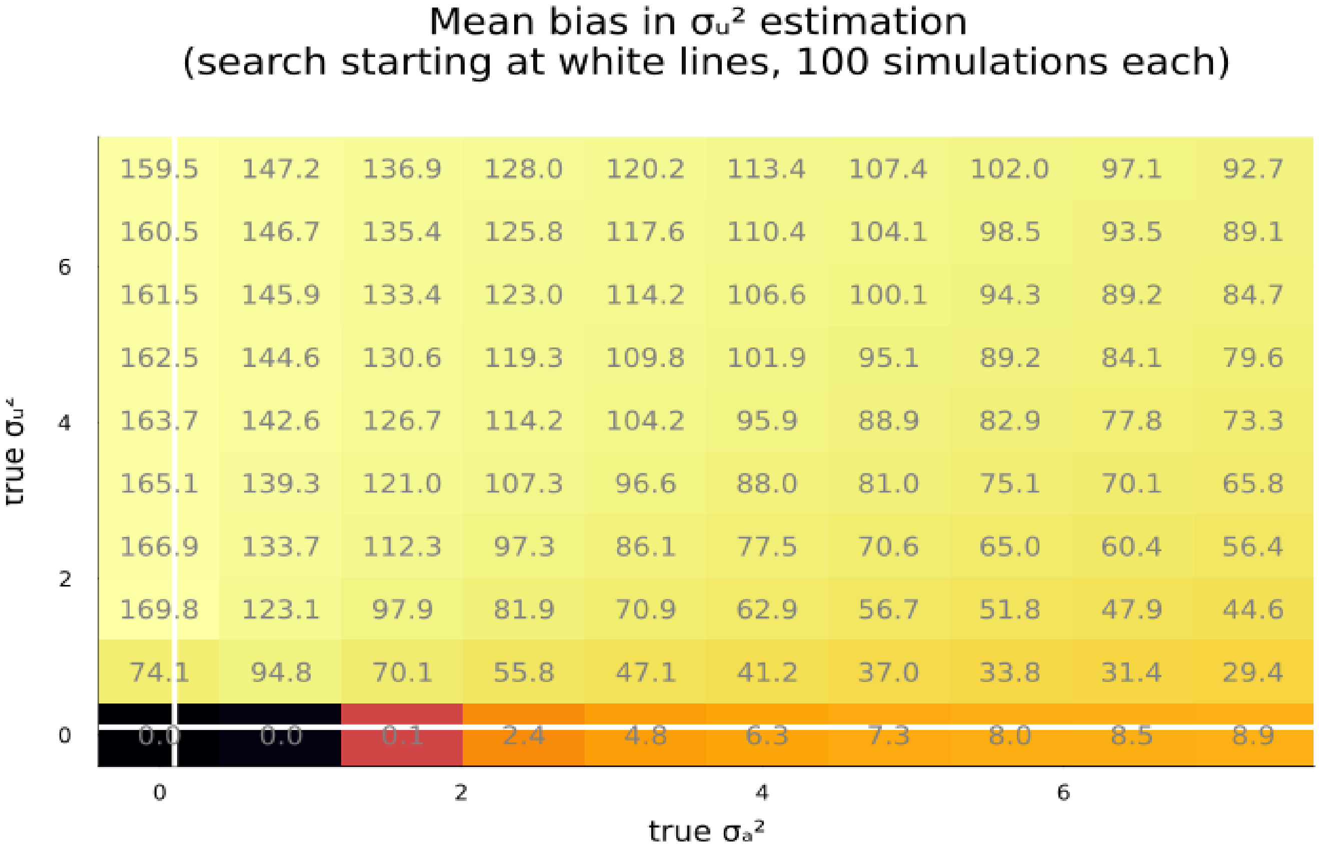
I will use the "best" routine from the last section to estimate the parameters on the real data.

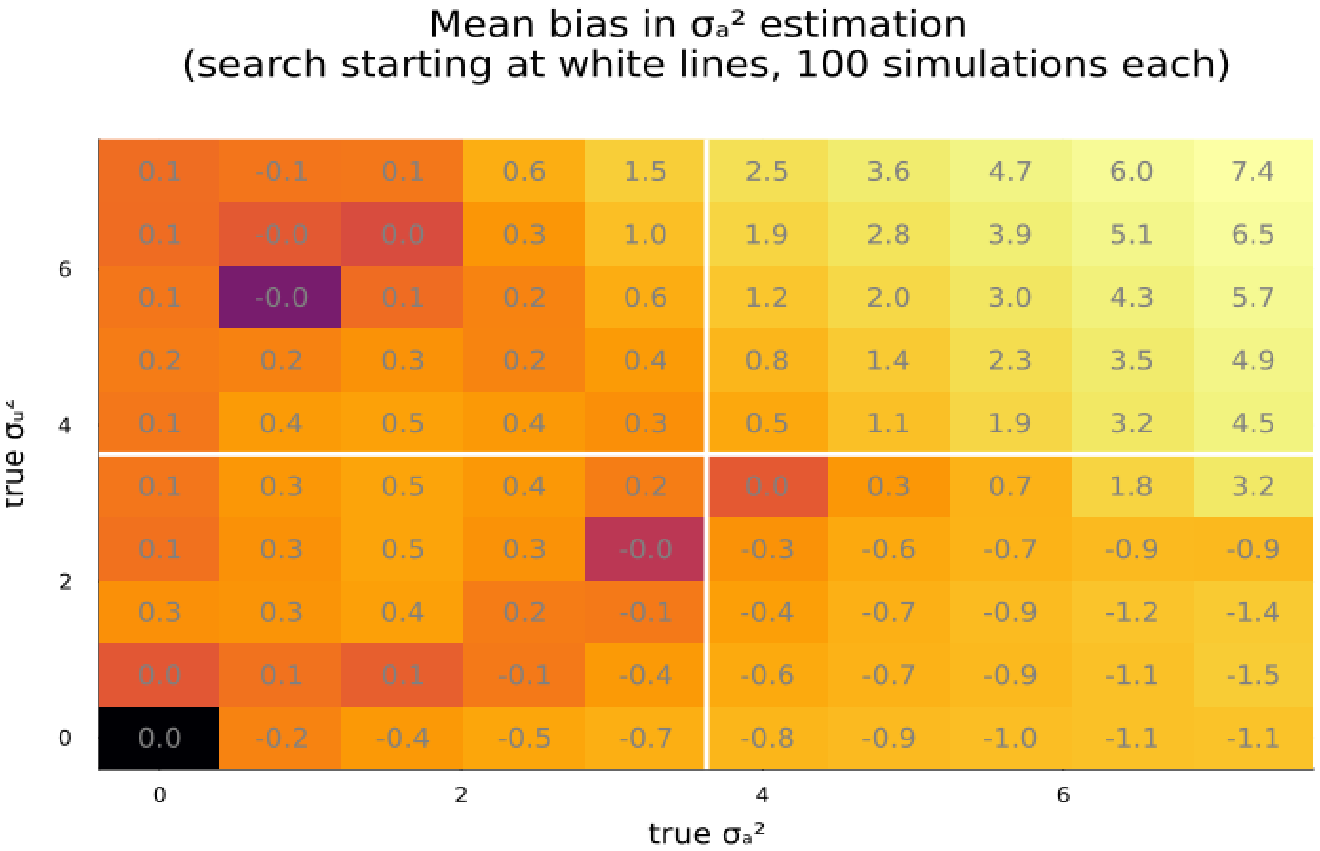
6 GMM

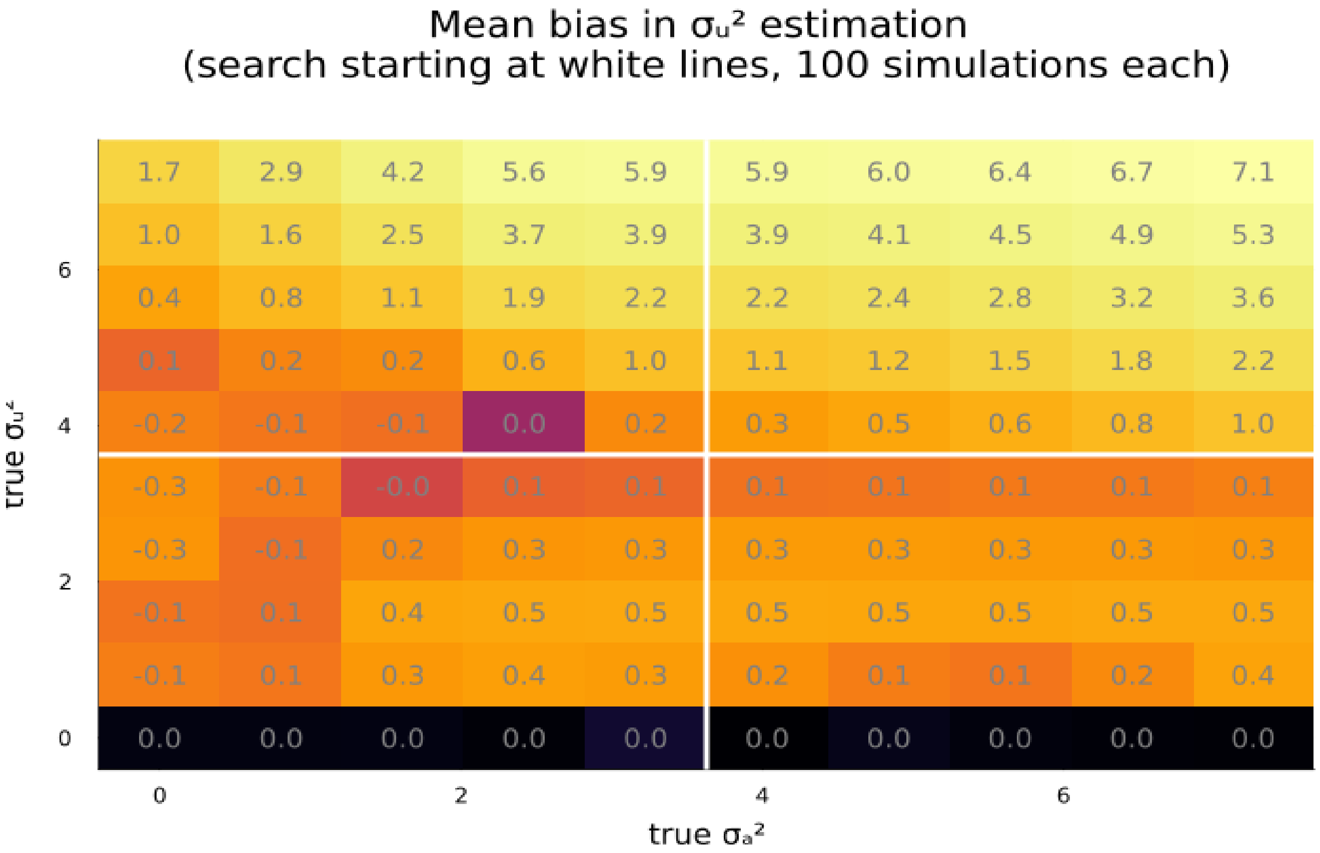
I will start programming the GMM estimation after finishing the simulation / MLE documentation.

7 Appendix: Figures

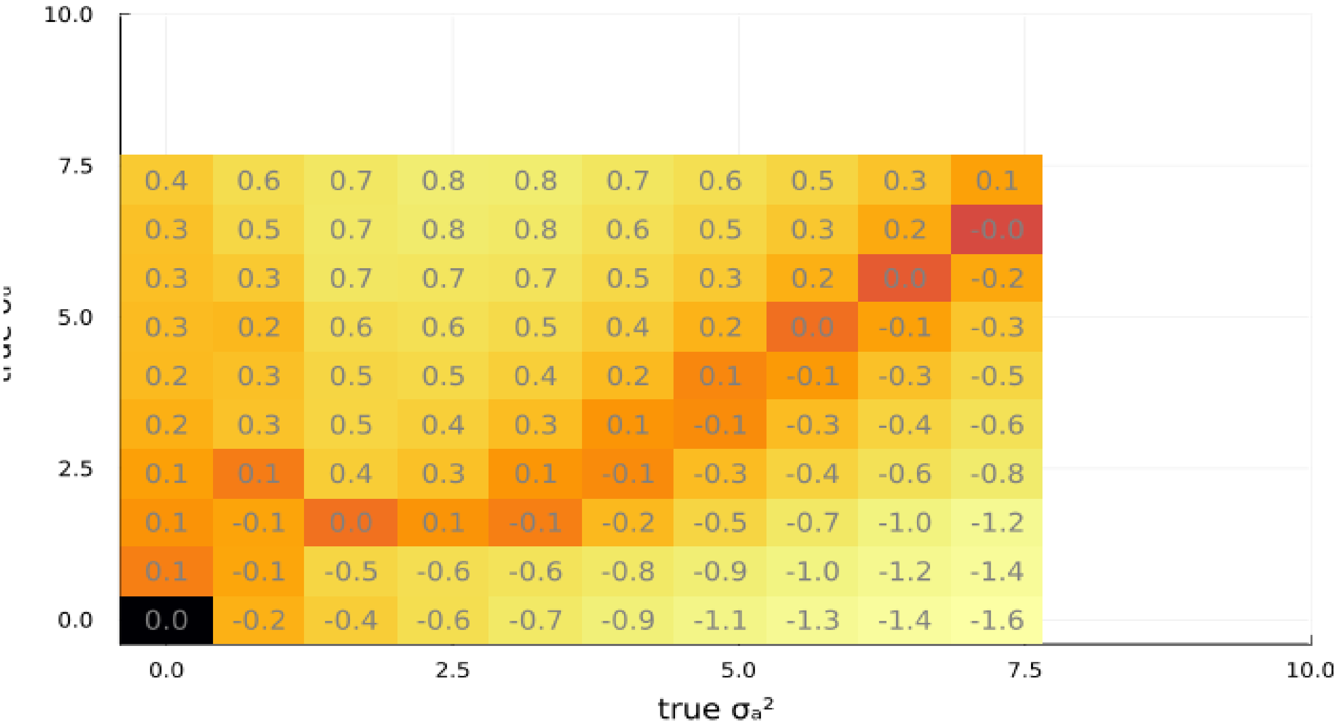




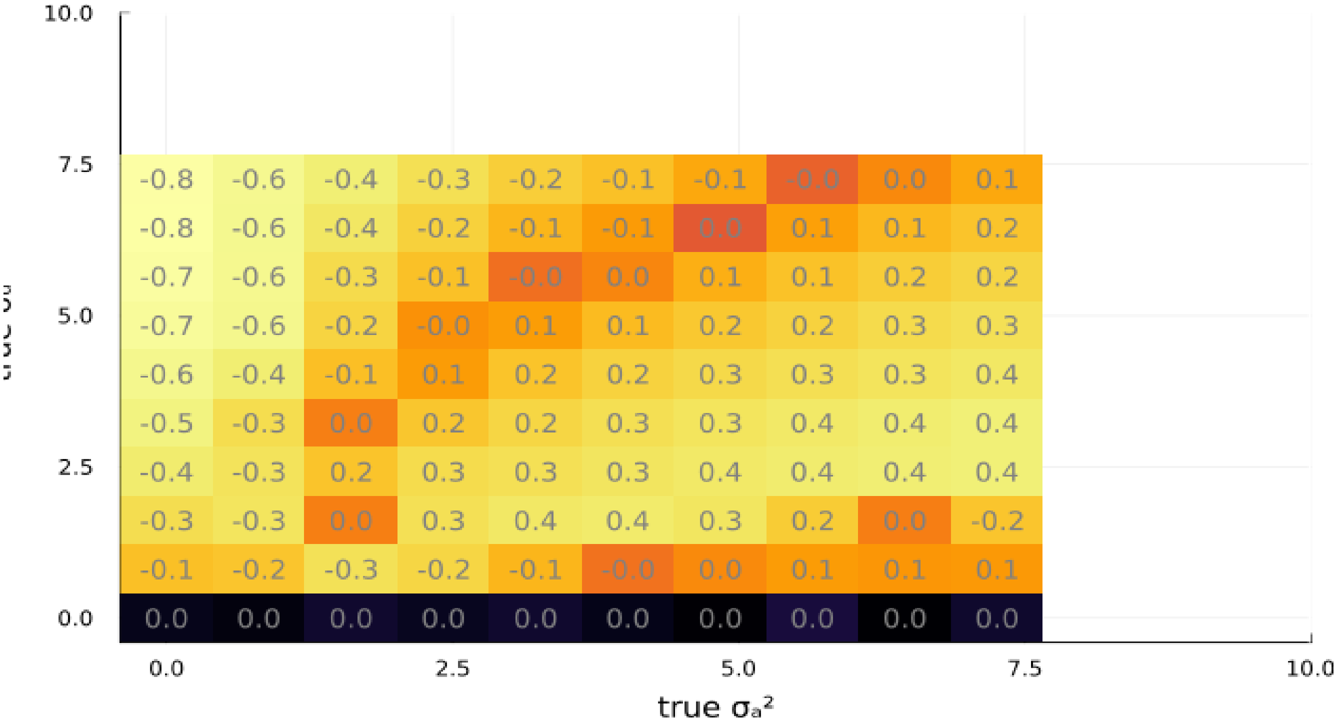

Mean bias in σₐ² estimation
(search starting at white lines, 100 simulations each)
0.1 -0.1 0.1 0.6 1.5 2.5 3.6 4.7 6.0 7.4
0.1 -0.0 0.0 0.3 1.0 1.9 2.8 3.9 5.1 6.5
0.1 -0.0 0.1 0.2 0.6 1.2 2.0 3.0 4.3 5.7
0.2 0.2 0.3 0.2 0.4 0.8 1.4 2.3 3.5 4.9
0.1 0.4 0.5 0.4 0.3 0.5 1.1 1.9 3.2 4.5
0.1 0.3 0.5 0.4 0.2 0.0 0.3 0.7 1.8 3.2
0.1 0.3 0.5 0.3 -0.0 -0.3 -0.6 -0.7 -0.9 -0.9
0.3 0.3 0.4 0.2 -0.1 -0.4 -0.7 -0.9 -1.2 -1.4
0.0 0.1 0.1 -0.1 -0.4 -0.6 -0.7 -0.9 -1.1 -1.5
0.0 -0.2 -0.4 -0.5 -0.7 -0.8 -0.9 -1.0 -1.1 -1.1
true σᵤ²
6
4
2
0
0
2
4
6
true σₐ²


Mean bias in σᵤ² estimation
(search starting at white lines, 100 simulations each)
1.7 2.9 4.2 5.6 5.9 5.9 6.0 6.4 6.7 7.1
1.0 1.6 2.5 3.7 3.9 3.9 4.1 4.5 4.9 5.3
0.4 0.8 1.1 1.9 2.2 2.2 2.4 2.8 3.2 3.6
0.1 0.2 0.2 0.6 1.0 1.1 1.2 1.5 1.8 2.2
-0.2 -0.1 -0.1 0.0 0.2 0.3 0.5 0.6 0.8 1.0
-0.3 -0.1 -0.0 0.1 0.1 0.1 0.1 0.1 0.1 0.1
-0.3 -0.1 0.2 0.3 0.3 0.3 0.3 0.3 0.3 0.3
-0.1 0.1 0.4 0.5 0.5 0.5 0.5 0.5 0.5 0.5
-0.1 0.1 0.3 0.4 0.3 0.2 0.1 0.1 0.2 0.4
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
true σᵤ²
6
4
2
0
0
2
4
6
true σₐ²

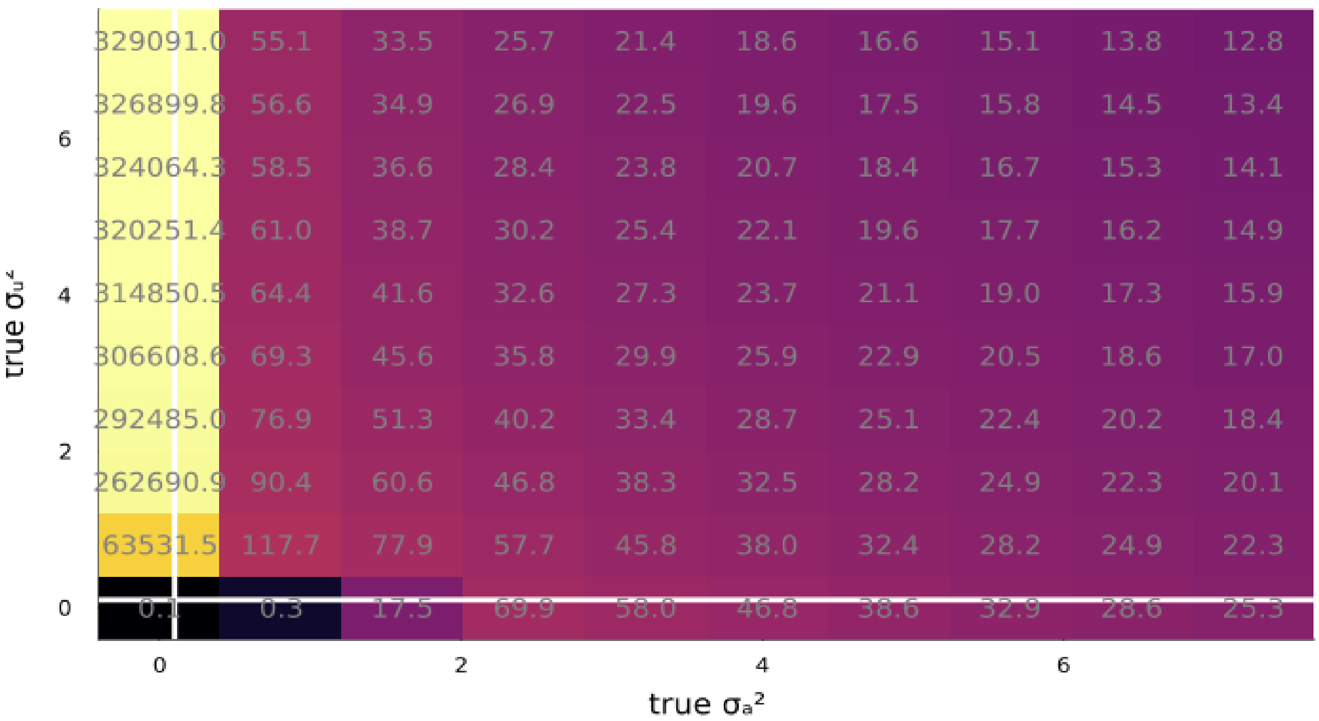
Mean bias in σ_a^2 estimation
(search starting at white lines, 100 simulations each)



Mean bias in σ_u^2 estimation
(search starting at white lines, 100 simulations each)

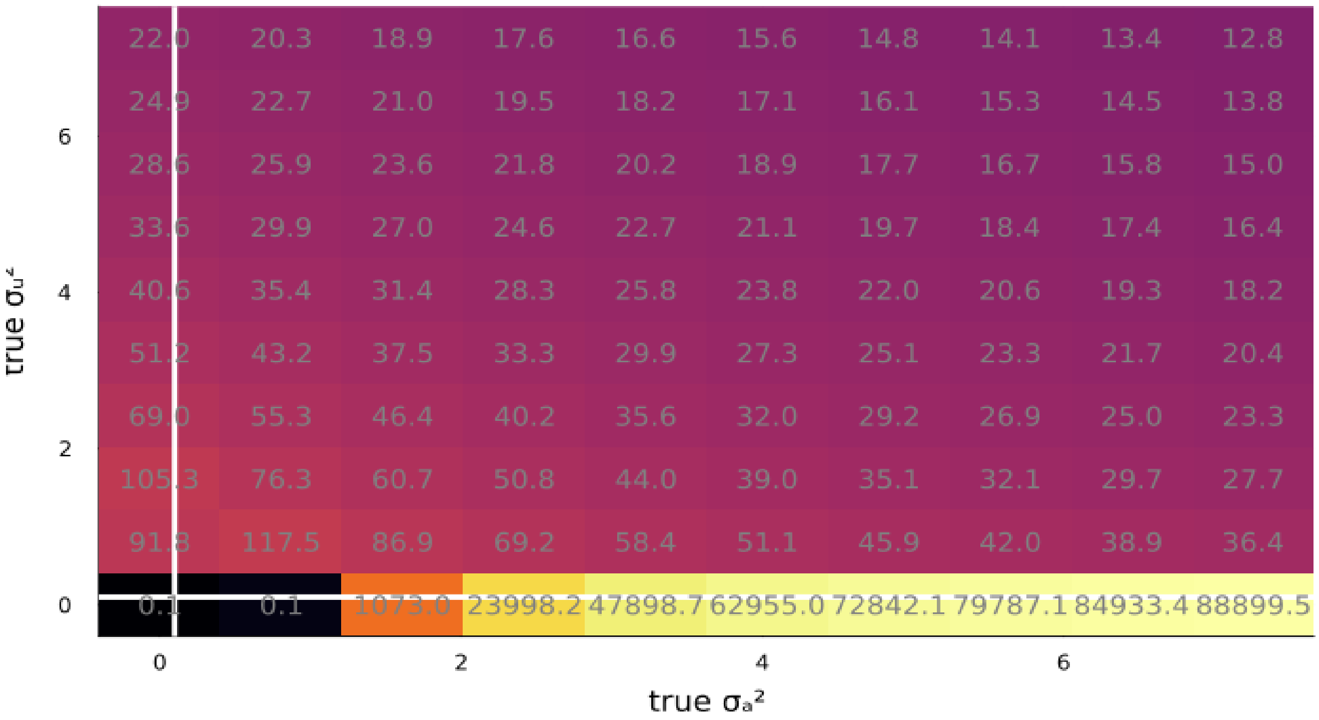


Mean bias % in σ_a^2 estimation
(search starting at white lines, 100 simulations each)

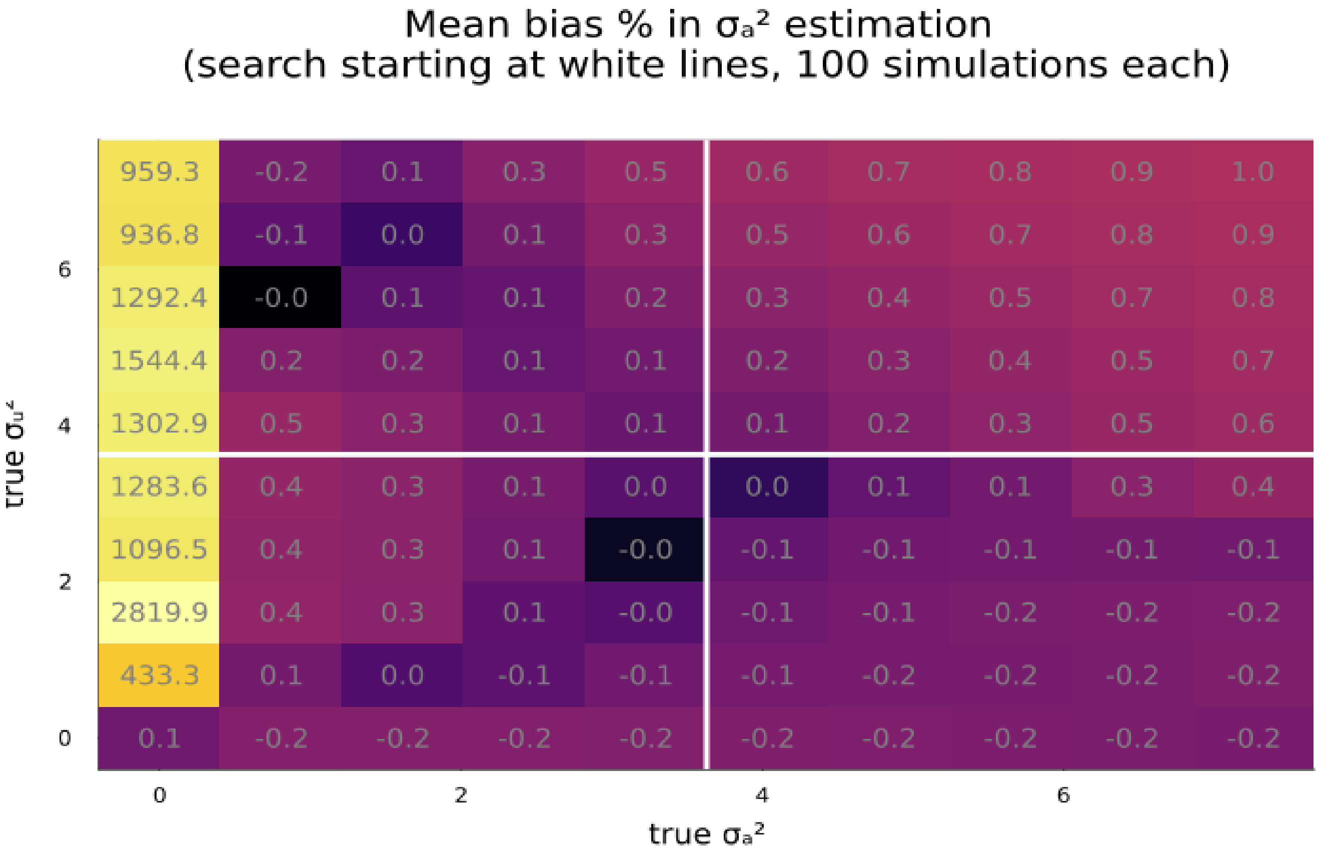


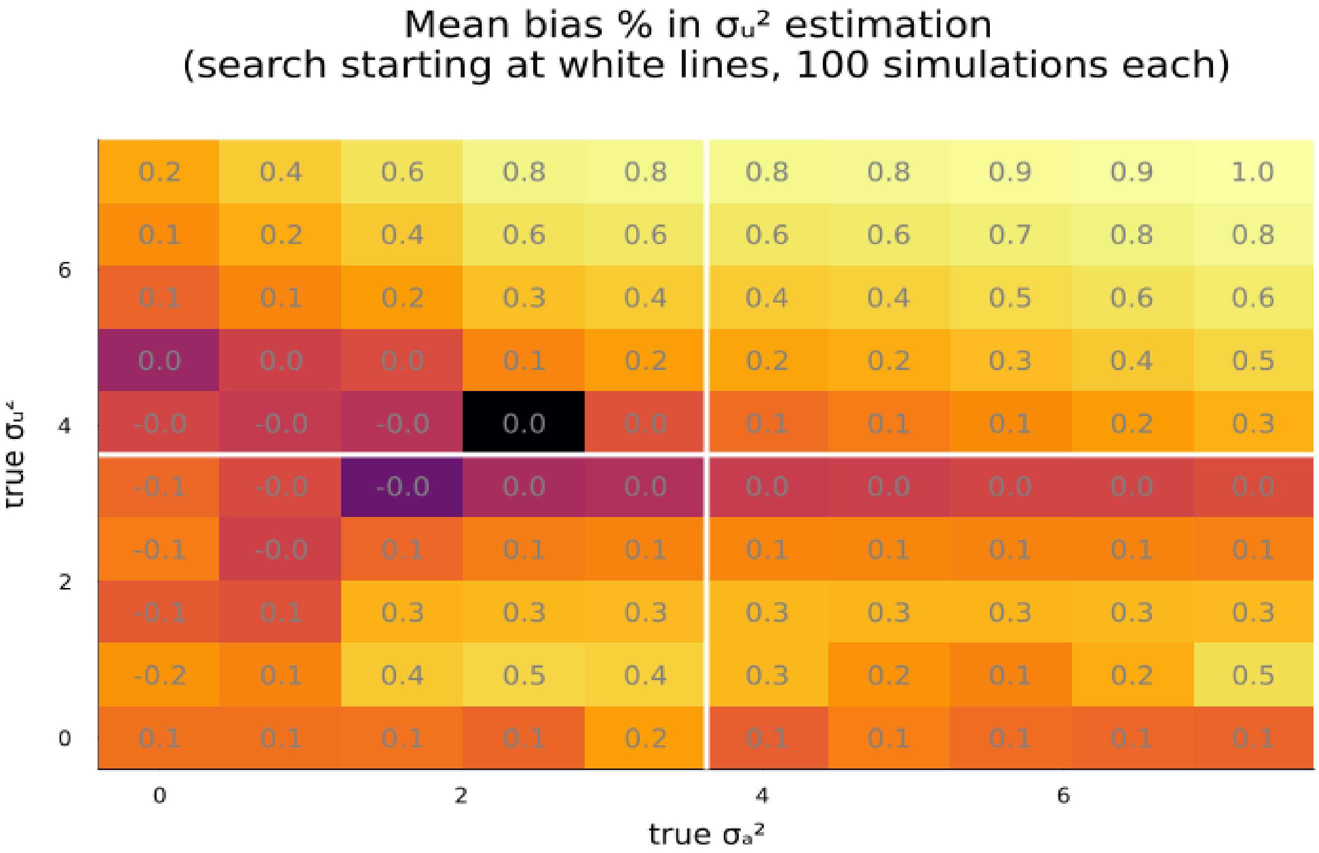
329091.0	55.1	33.5	25.7	21.4	18.6	16.6	15.1	13.8	12.8
326899.8	56.6	34.9	26.9	22.5	19.6	17.5	15.8	14.5	13.4
324064.3	58.5	36.6	28.4	23.8	20.7	18.4	16.7	15.3	14.1
320251.4	61.0	38.7	30.2	25.4	22.1	19.6	17.7	16.2	14.9
314850.5	64.4	41.6	32.6	27.3	23.7	21.1	19.0	17.3	15.9
306608.6	69.3	45.6	35.8	29.9	25.9	22.9	20.5	18.6	17.0
292485.0	76.9	51.3	40.2	33.4	28.7	25.1	22.4	20.2	18.4
262690.9	90.4	60.6	46.8	38.3	32.5	28.2	24.9	22.3	20.1
63531.5	117.7	77.9	57.7	45.8	38.0	32.4	28.2	24.9	22.3
0.1	0.3	17.5	69.9	58.0	46.8	38.6	32.9	28.6	25.3

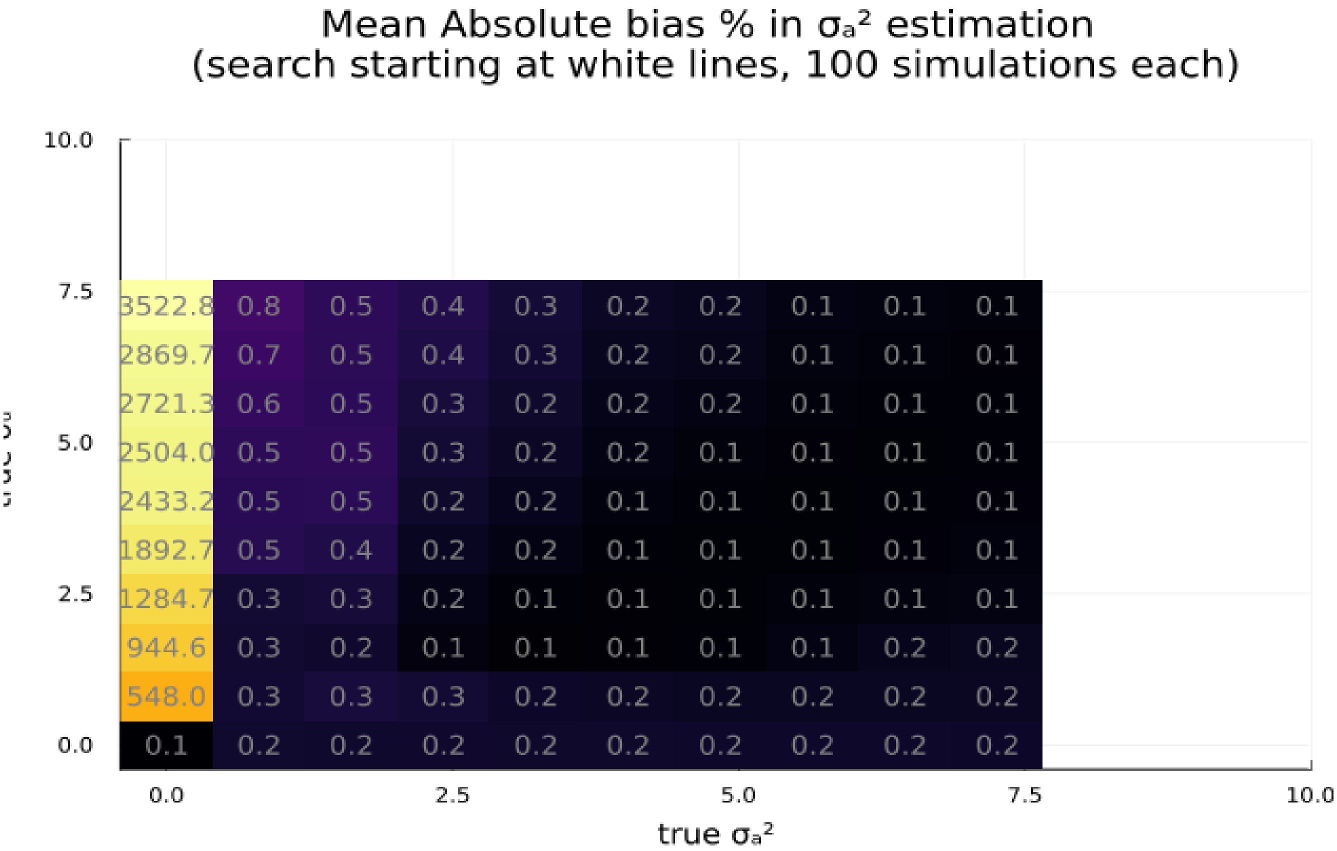
Mean bias % in σ_u^2 estimation
(search starting at white lines, 100 simulations each)

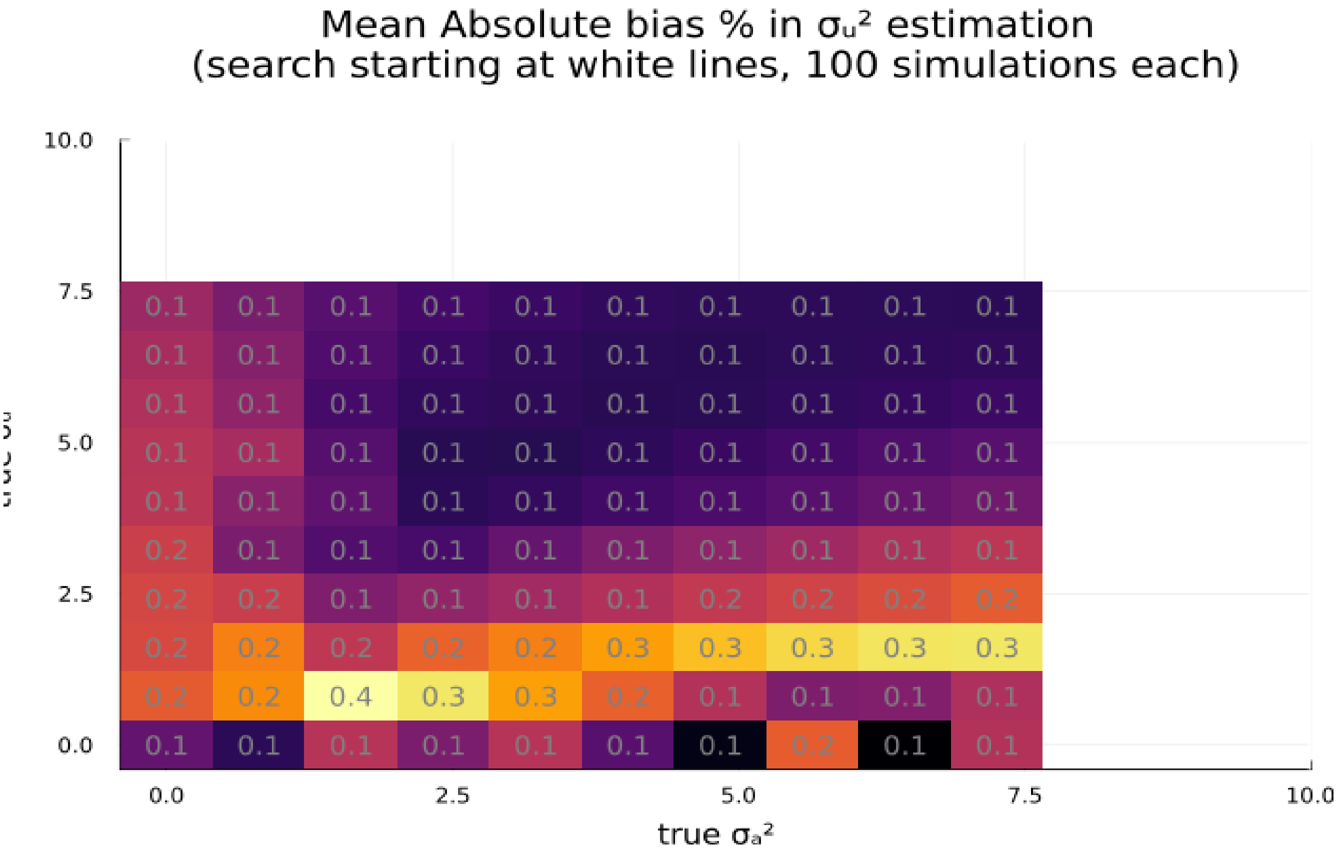


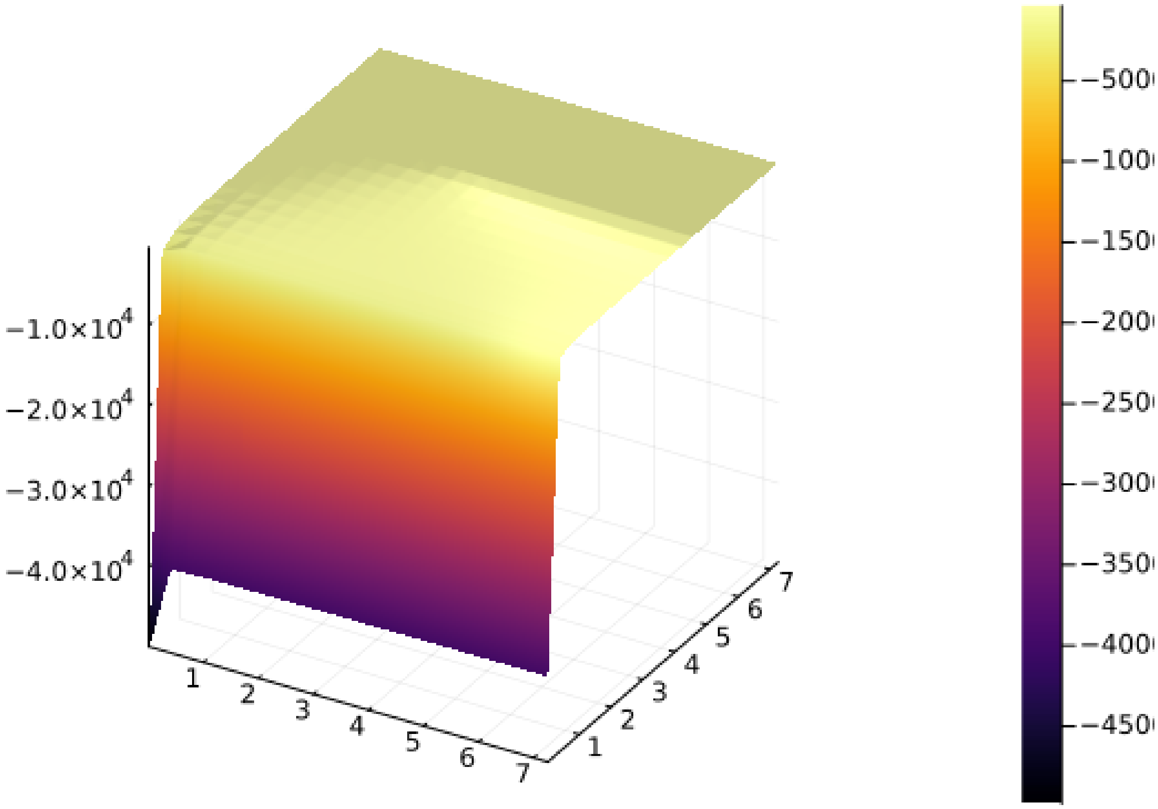
22.0	20.3	18.9	17.6	16.6	15.6	14.8	14.1	13.4	12.8
24.9	22.7	21.0	19.5	18.2	17.1	16.1	15.3	14.5	13.8
28.6	25.9	23.6	21.8	20.2	18.9	17.7	16.7	15.8	15.0
33.6	29.9	27.0	24.6	22.7	21.1	19.7	18.4	17.4	16.4
40.6	35.4	31.4	28.3	25.8	23.8	22.0	20.6	19.3	18.2
51.2	43.2	37.5	33.3	29.9	27.3	25.1	23.3	21.7	20.4
69.0	55.3	46.4	40.2	35.6	32.0	29.2	26.9	25.0	23.3
105.3	76.3	60.7	50.8	44.0	39.0	35.1	32.1	29.7	27.7
91.8	117.5	86.9	69.2	58.4	51.1	45.9	42.0	38.9	36.4
0.1	0.1	1073.0	23998.2	47898.7	62955.0	72842.1	79787.1	84933.4	88899.5

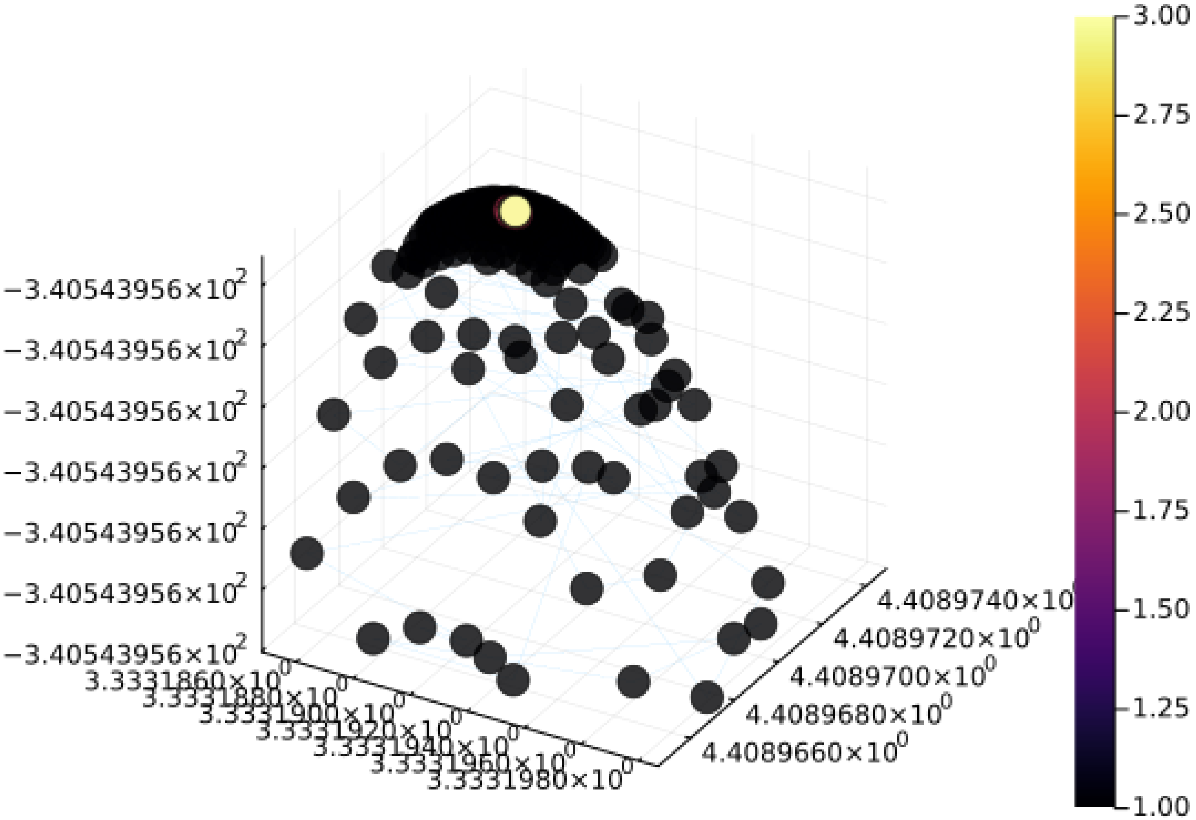

Mean bias % in σₐ² estimation
(search starting at white lines, 100 simulations each)


Mean bias % in σᵤ² estimation
(search starting at white lines, 100 simulations each)


Mean Absolute bias % in σₐ² estimation
(search starting at white lines, 100 simulations each)
10.0
7.5
5.0
2.5
0.0
3522.8 0.8 0.5 0.4 0.3 0.2 0.2 0.1 0.1 0.1
2869.7 0.7 0.5 0.4 0.3 0.2 0.2 0.1 0.1 0.1
2721.3 0.6 0.5 0.3 0.2 0.2 0.2 0.1 0.1 0.1
2504.0 0.5 0.5 0.3 0.2 0.2 0.1 0.1 0.1 0.1
2433.2 0.5 0.5 0.2 0.2 0.1 0.1 0.1 0.1 0.1
1892.7 0.5 0.4 0.2 0.2 0.1 0.1 0.1 0.1 0.1
1284.7 0.3 0.3 0.2 0.1 0.1 0.1 0.1 0.1 0.1
944.6 0.3 0.2 0.1 0.1 0.1 0.1 0.1 0.2 0.2
548.0 0.3 0.3 0.3 0.2 0.2 0.2 0.2 0.2 0.2
0.1 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.0
2.5
5.0
7.5
10.0
true σₐ²


Mean Absolute bias % in σᵤ² estimation
(search starting at white lines, 100 simulations each)
10.0
7.5
5.0
2.5
0.0
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.2 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.2 0.2 0.1 0.1 0.1 0.1 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.3 0.3 0.3 0.3 0.3
0.2 0.2 0.4 0.3 0.3 0.2 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.2 0.1 0.1
0.0
2.5
5.0
7.5
10.0
true σₐ²


−1.0×10^4
−2.0×10^4
−3.0×10^4
−4.0×10^4
1
2
3
4
5
6
7
1
2
3
4
5
6
7
−500
−100
−150
−200
−250
−300
−350
−400
−450


−3.40543956×10^2
−3.40543956×10^2
−3.40543956×10^2
−3.40543956×10^2
−3.40543956×10^2
−3.40543956×10^2
−3.40543956×10^2
3.3331860×10^0
3.3331880×10^0
3.3331900×10^0
3.3331920×10^0
3.3331940×10^0
3.3331960×10^0
3.3331980×10^0
4.4089740×10
4.4089720×10^0
4.4089700×10^0
4.4089680×10^0
4.4089660×10^0
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00