

# Simulation Experiment Results

Victor Tsang (z5209633)

2022-11-13

## Load in the results

```
library(knitr)
library(tidyverse)
library(latex2exp)

load("../data/synthetic-data.RData")
attach(synthetic.data.config)

estimates = readRDS("../data/sim_exp-estimate_extinction_results.RDS")
estimates = estimates %>% filter(!(method %in% c("SI-RM", "GB-RM")))
estimates = estimates %>% mutate(across(method, str_replace, 'SI-RM-corrected', 'SI-RM'))
estimates = estimates %>% mutate(method_cat = ifelse(method %in% c("SI-RM", "MINMI"), "Proposed", "Existing"))
estimates = estimates %>% filter(error_factor != 4)
head(estimates)
```

##	id	error_factor	method	lower	point	upper	point_runtime
## 1	1	0.0	MLE	NA	12660.896	NA	1.907349e-05
## 2	1	0.0	BA-MLE	NA	12293.940	NA	1.682997e-03
## 3	1	0.0	SI-UGM	11262.804	12422.265	12681.61	3.777524e+00
## 4	2	0.5	MLE	NA	9871.056	NA	1.279831e-03
## 5	2	0.5	BA-MLE	NA	9364.609	NA	4.145861e-03
## 6	2	0.5	SI-UGM	7789.998	9518.421	10035.51	2.695084e+00
##	conf_int_runtime		B.point	B.lower	B.upper	method_cat	
## 1			NA	NA	NA	Existing	
## 2			NA	NA	NA	Existing	
## 3		3.777524	NA	NA	NA	Existing	
## 4			NA	NA	NA	Existing	
## 5			NA	NA	NA	Existing	
## 6		2.695084	NA	NA	NA	Existing	

```
# Point estimates
performance.point_estimates = estimates %>%
  filter(!is.na(point)) %>%
  group_by(error_factor, method, method_cat) %>%
  summarise(MSE_000 = mean((point - theta.true)^2)/1000,
            bias = mean(point)-theta.true,
            variance_000 = var(point)/1000,
            avg_runtime = round(mean(point_runtime), 5))
```

```
## 'summarise()' has grouped output by 'error_factor', 'method'. You can override
## using the '.groups' argument.
```

```
kable(performance.point_estimates)
```

error_factor	method	method_cat	MSE_000	bias	variance_000	avg_runtime
0.0	BA-MLE	Existing	234.5607	-0.9379571	234.7946	0.00002
0.0	GRIWM	Existing	1183.5359	-949.8850000	281.5360	2.33548
0.0	GRIWM-corrected	Existing	246.1900	133.7070000	228.5410	13.89881
0.0	MINMI	Proposed	247.4576	139.4092643	228.2509	0.00001
0.0	MLE	Existing	438.6601	475.2971837	212.9656	0.00002
0.0	SI-RM	Proposed	438.6601	475.2971837	212.9656	0.05641
0.0	SI-UGM	Existing	248.9648	150.6138962	226.5068	4.70657
0.0	STRAUSS	Existing	234.8453	-0.7152842	235.0799	0.00002
0.5	BA-MLE	Existing	244.0353	-22.0990141	243.7908	0.00003
0.5	GRIWM	Existing	1275.8894	-992.6550000	290.8162	2.35887
0.5	GRIWM-corrected	Existing	244.7798	95.1730000	235.9578	13.88254
0.5	MINMI	Proposed	253.0606	118.8454210	239.1755	0.00047
0.5	MLE	Existing	428.0602	455.1437961	221.1254	0.00002
0.5	SI-RM	Proposed	428.0602	455.1437961	221.1254	0.06072
0.5	SI-UGM	Existing	250.7756	117.2579537	237.2634	2.33059
0.5	STRAUSS	Existing	245.5802	-22.8493286	245.3034	0.00002
1.0	BA-MLE	Existing	365.6242	-45.7554617	363.8946	0.00002
1.0	GRIWM	Existing	1547.7470	-1060.0020000	424.5673	18.10725
1.0	GRIWM-corrected	Existing	345.2420	34.4120000	344.4022	13.90095
1.0	MINMI	Proposed	373.3208	103.7670782	362.9161	0.00057
1.0	MLE	Existing	516.8878	432.6138460	330.0631	0.00002
1.0	SI-RM	Proposed	516.8878	432.6138460	330.0631	0.05992
1.0	SI-UGM	Existing	371.3002	115.2129004	358.3845	1.93741
1.0	STRAUSS	Existing	366.4727	-50.5383309	364.2829	0.00002
2.0	BA-MLE	Existing	542.9867	-233.5215118	488.9434	0.00002
2.0	GRIWM	Existing	2506.9717	-1407.5250000	526.3714	2.36410
2.0	GRIWM-corrected	Existing	504.9335	-278.4774775	427.8120	13.94220
2.0	MINMI	Proposed	491.9000	27.3303294	491.6447	0.00071
2.0	MLE	Existing	507.4514	253.7890364	443.4860	0.00002
2.0	SI-RM	Proposed	507.4514	253.7890364	443.4860	0.05993
2.0	SI-UGM	Existing	501.5358	65.3780494	497.7593	1.68008
2.0	STRAUSS	Existing	553.8156	-247.9251968	492.8416	0.00002

```
# Confidence Intervals
```

```
performance.conf_int_estimates = estimates %>%
  filter(!is.na(conf_int_runtime)) %>%
  mutate(width = upper - lower,
         contains_theta = ifelse(theta.true > lower & theta.true < upper, 1, 0)) %>%
  group_by(error_factor, method, method_cat) %>%
  summarise(Coverage = round(mean(contains_theta) * 100, 1),
            `Average Width` = round(mean(width), 2),
            `Average Runtime` = round(mean(conf_int_runtime), 4)) %>%
```

```
ungroup() %>%
  arrange(desc(Coverage), `Average Width`, `Average Runtime`)
```

## 'summarise()' has grouped output by 'error\_factor', 'method'. You can override  
## using the '.groups' argument.

```
kable(performance.conf_int_estimates)
```

error_factor	method	method_cat	Coverage	Average Width	Average Runtime
0.0	SI-RM	Proposed	97.4	2005.35	0.0564
0.0	SI-UGM	Existing	97.3	1961.14	4.7066
0.5	SI-RM	Proposed	96.3	2097.43	0.0607
0.5	SI-UGM	Existing	96.2	2091.09	2.3306
0.5	MINMI	Proposed	95.2	2066.21	0.0013
2.0	SI-RM	Proposed	95.1	2960.26	0.0599
2.0	MINMI	Proposed	95.0	2945.03	0.0021
2.0	SI-UGM	Existing	94.9	2964.30	1.6801
0.0	MINMI	Proposed	94.5	1917.16	0.0000
1.0	SI-RM	Proposed	93.8	2346.11	0.0599
1.0	SI-UGM	Existing	93.7	2351.53	1.9374
1.0	MINMI	Proposed	93.0	2325.84	0.0014
2.0	GRIWM-corrected	Existing	80.6	1949.67	13.9422
1.0	GRIWM-corrected	Existing	69.5	1047.99	13.9010
0.5	GRIWM-corrected	Existing	49.9	548.08	13.8825
2.0	GRIWM	Existing	22.6	2163.50	2.3641
1.0	GRIWM	Existing	13.9	1162.65	18.1072
0.5	GRIWM	Existing	7.0	608.33	2.3589
0.0	GRIWM	Existing	0.0	0.00	2.3355
0.0	GRIWM-corrected	Existing	0.0	0.00	13.8988

## Point Estimates

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.2.2
```

```
##
```

```
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      group_rows
```

```
for (err in error_factors) {
  experiment.results.kbl = performance.point_estimates %>%
    filter(error_factor == err) %>%
    ungroup() %>%
    mutate(across(!c(method, avg_runtime, method_cat), round)) %>%
```

```

mutate(avg_runtime=round(avg_runtime, digits=5)) %>%
  arrange(MSE_000) %>%
  select(-c(error_factor, method_cat)) %>%
  kable(booktabs=T, col.names = c("", "(000's years)", "(years)", "(000's years)", "(seconds)"), form
  add_header_above(c("Method" = 1, "MSE" = 1, "Bias"=1, "Variance"=1, "Average Runtime"=1), line=F, a
writeLines(experiment.results.kbl, paste0("../figures/table-sim-exp-point-error", err, ".tex"))
print(experiment.results.kbl)
}

```

Method	MSE (000's years)	Bias (years)	Variance (000's years)	Average Runtime (seconds)
BA-MLE	235	-1	235	0.00002
STRAUSS	235	-1	235	0.00002
GRIWM-corrected	246	134	229	13.89881
MINMI	247	139	228	0.00001
SI-UGM	249	151	227	4.70657
MLE	439	475	213	0.00002
SI-RM	439	475	213	0.05641
GRIWM	1184	-950	282	2.33548

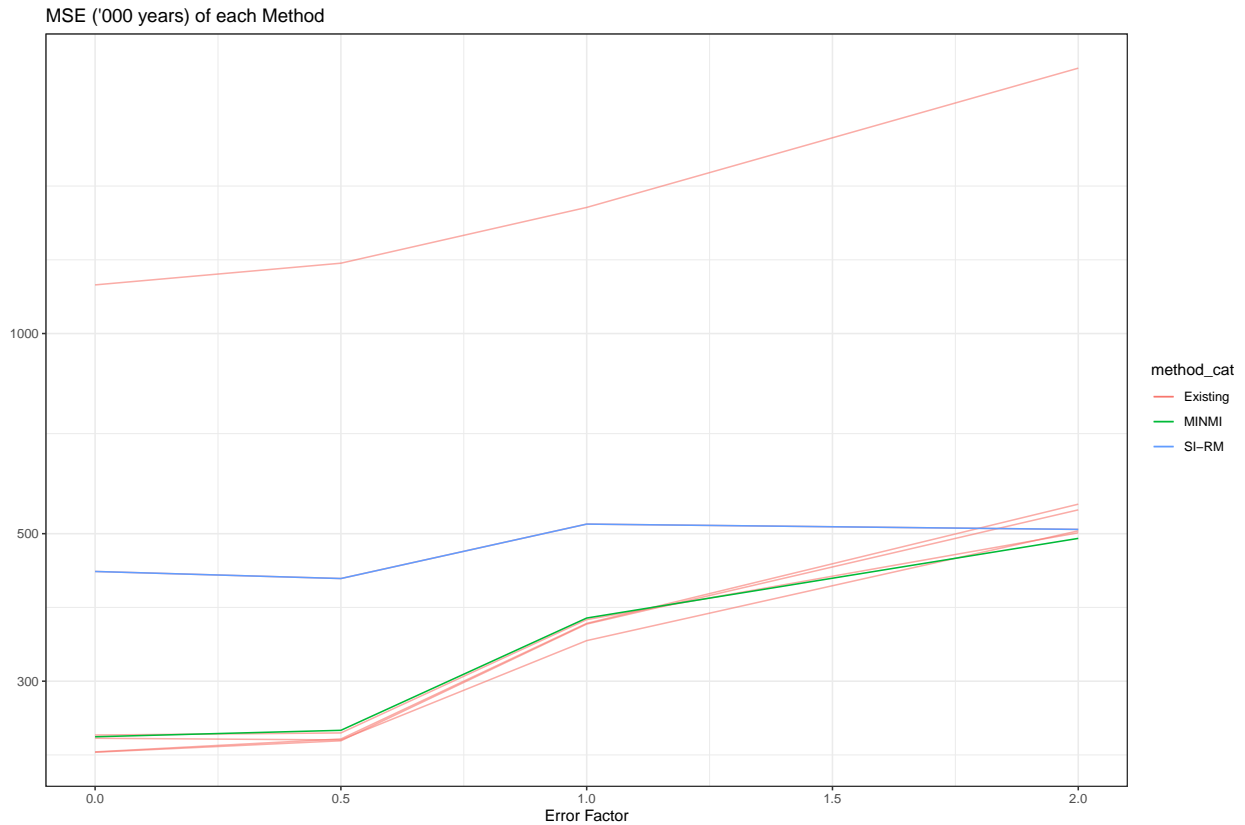
Method	MSE (000's years)	Bias (years)	Variance (000's years)	Average Runtime (seconds)
BA-MLE	244	-22	244	0.00003
GRIWM-corrected	245	95	236	13.88254
STRAUSS	246	-23	245	0.00002
SI-UGM	251	117	237	2.33059
MINMI	253	119	239	0.00047
MLE	428	455	221	0.00002
SI-RM	428	455	221	0.06072
GRIWM	1276	-993	291	2.35887

Method	MSE (000's years)	Bias (years)	Variance (000's years)	Average Runtime (seconds)
GRIWM-corrected	345	34	344	13.90095
BA-MLE	366	-46	364	0.00002
STRAUSS	366	-51	364	0.00002
SI-UGM	371	115	358	1.93741
MINMI	373	104	363	0.00057
MLE	517	433	330	0.00002
SI-RM	517	433	330	0.05992
GRIWM	1548	-1060	425	18.10725

Method	MSE (000's years)	Bias (years)	Variance (000's years)	Average Runtime (seconds)
MINMI	492	27	492	0.00071
SI-UGM	502	65	498	1.68008
GRIWM-corrected	505	-278	428	13.94220
MLE	507	254	443	0.00002
SI-RM	507	254	443	0.05993
BA-MLE	543	-234	489	0.00002
STRAUSS	554	-248	493	0.00002
GRIWM	2507	-1408	526	2.36410

Method	MSE (000's years)	Bias (years)	Variance (000's years)	Average Runtime (seconds)
--------	----------------------	-----------------	---------------------------	------------------------------

```
ggplot() +
  geom_line(data = filter(performance.point_estimates, method_cat == "Existing"),
            aes(x=error_factor, group=method, y=MSE_000, colour=method_cat), alpha=0.6) +
  geom_line(data = filter(performance.point_estimates, method_cat == "Proposed"),
            aes(x=error_factor, y=MSE_000, colour=method)) +
  theme_bw() +
  labs(x="Error Factor", y=NULL, title="MSE ('000 years) of each Method") +
  scale_y_continuous(trans='log10')
```



## Confidence Intervals

```
options(scipen=9)
for (metric in c("Coverage", "Average Width", "Average Runtime")) {
  experiment.results.kbl = performance.conf_int_estimates %>%
    select(c(method, error_factor, one_of(metric))) %>%
    pivot_wider(id_cols = method,
                 names_from=error_factor,
                 values_from=one_of(metric),
                 names_prefix=paste(metric, "| error = sigma*")) %>%
    arrange(!syms(paste(metric, "| error = sigma*0"))) %>%
    kable(col.names = c("", paste0(c(0,0.5,1,2), r"{$\sigma$}")), booktabs=T, format="latex", escape =
    add_header_above(unlist(lst("Method" = 1, !!metric := 4)), line=F)

  print(experiment.results.kbl)
  writeLines(experiment.results.kbl, paste0("../figures/table-sim-exp-conf-int-", str_replace(tolower(m
}

```

Method	Coverage			
	$0*\sigma$	$0.5*\sigma$	$1*\sigma$	$2*\sigma$
SI-RM	97.4	96.3	95.1	93.8
SI-UGM	97.3	96.2	94.9	93.7
MINMI	94.5	95.2	95.0	93.0
GRIWM-corrected	0.0	49.9	80.6	69.5
GRIWM	0.0	7.0	22.6	13.9

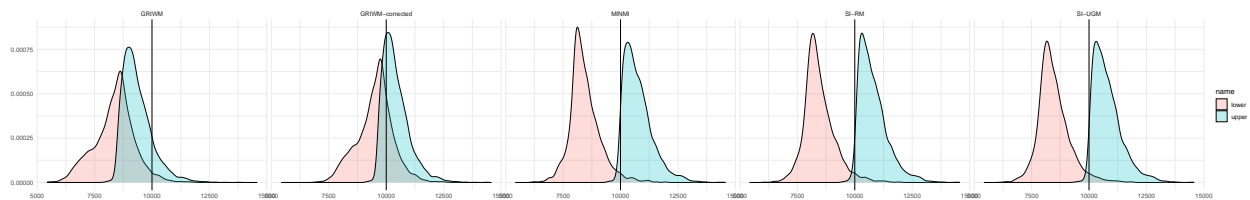
Method	Average Width			
	$0*\sigma$	$0.5*\sigma$	$1*\sigma$	$2*\sigma$
SI-RM	2005.35	2097.43	2960.26	2346.11
SI-UGM	1961.14	2091.09	2964.30	2351.53
MINMI	1917.16	2066.21	2945.03	2325.84
GRIWM-corrected	0.00	548.08	1949.67	1047.99
GRIWM	0.00	608.33	2163.50	1162.65

Method	Average Runtime			
	$0*\sigma$	$0.5*\sigma$	$1*\sigma$	$2*\sigma$
SI-RM	0.0564	0.0607	0.0599	0.0599
SI-UGM	4.7066	2.3306	1.6801	1.9374
MINMI	0.0000	0.0013	0.0021	0.0014
GRIWM-corrected	13.8988	13.8825	13.9422	13.9010
GRIWM	2.3355	2.3589	2.3641	18.1072

```
estimates %>%
  filter(!is.na(lower)) %>%
  select(method, lower, upper) %>%
  pivot_longer(cols=c(lower, upper)) %>%
  filter(!is.na(value)) %>%
  ggplot(aes(x=value, fill=name)) +
  geom_density(alpha=0.25) +
  geom_vline(aes(xintercept=theta.true)) +
  facet_wrap(method ~ ., nrow=1) +

```

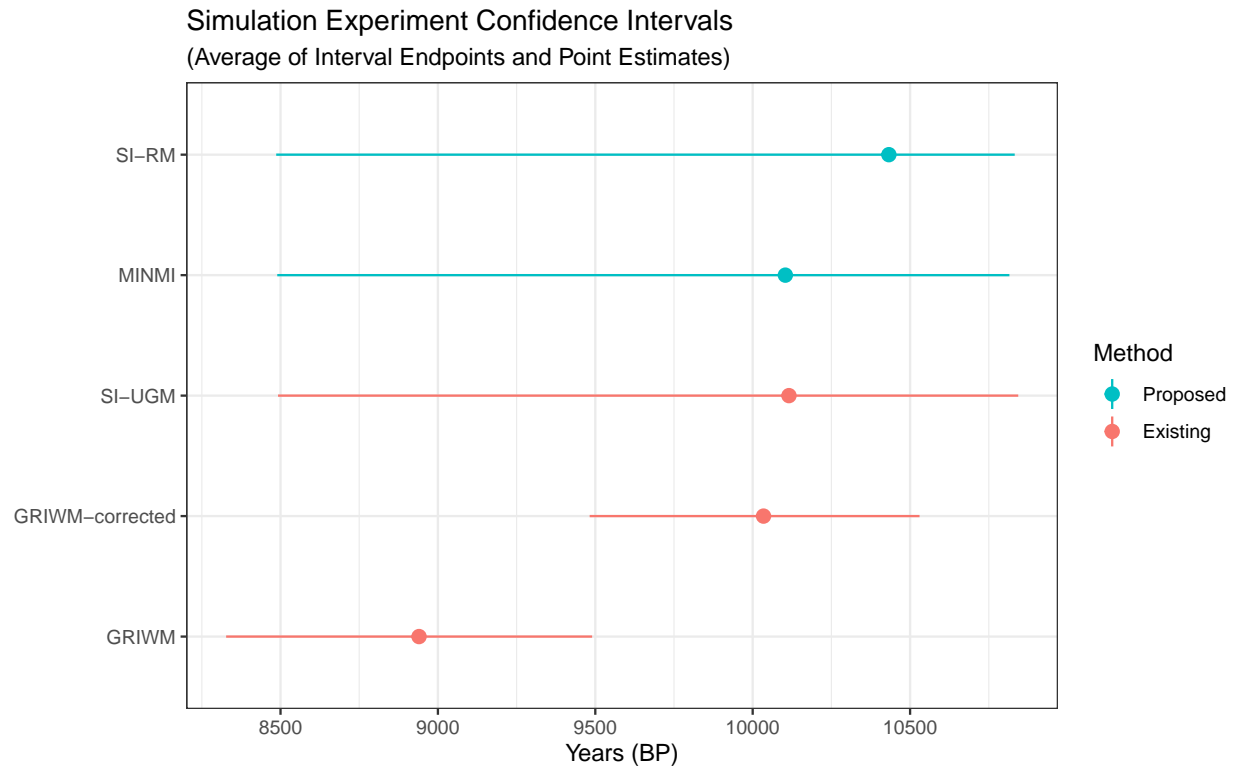
```
theme_minimal() +
labs(x=NULL, y=NULL)
```



```
p = estimates %>%
  filter(!is.na(lower) & error_factor == 1) %>%
  group_by(method_cat, method) %>%
  summarise(lower = mean(lower), upper=mean(upper), point=mean(point)) %>%
  mutate(width=upper-lower) %>%
  mutate(method_cat_int = ifelse(method_cat == "Existing", 0, 1)) %>%
  ggplot(aes(colour=method_cat)) +
  geom_pointrange(aes(xmin=lower, xmax=upper, x=point, y=reorder(method, method_cat_int))) +
  guides(colour = guide_legend(reverse=TRUE)) +
  labs(y=NULL, x="Years (BP)",
       title="Simulation Experiment Confidence Intervals",
       colour="Method",
       subtitle="(Average of Interval Endpoints and Point Estimates)") +
  scale_x_continuous(breaks = seq(from=8000, to=11500, by=500)) +
  theme_bw()
```

## 'summarise()' has grouped output by 'method\_cat'. You can override using the  
## '.groups' argument.

p



```
ggsave(filename="../figures/sim-exp-intervals.svg", plot=p, height=5, width=8)
```

## Monte Carlo Samples for MINMI

```
B.minmi = readRDS("../data/sim_exp-B-minmi.RDS")

B.minmi.kbl = B.minmi %>%
  kable(col.names = c("", "q = 0.025", "q = 0.5", "q = 0.975"),
        booktabs=T, format="latex", escape = FALSE) %>%
  add_header_above(c(`Measurement Error Variation`=1, B=3), line=F)

print(B.minmi.kbl)

##
## \begin{tabular}{rrrr}
## \toprule
## \multicolumn{1}{c}{Measurement Error Variation} & \multicolumn{3}{c}{B} \\
## & q = 0.025 & q = 0.5 & q = 0.975 \\
## \midrule
## 0.5 & 5 & 6 & 3 \\
## 1.0 & 14 & 24 & 9 \\
## 2.0 & 42 & 91 & 33 \\
## 4.0 & 136 & 243 & 119 \\
## 0.0 & 2 & 2 & 2 \\
## \bottomrule
## \end{tabular}
```



```
writeLines(B.minmi.kbl,  
           "../figures/table-sim-exp-minmi-Bs.tex")
```