AMS514 Project 1

Evaluation of Gustafsson's Improvement of the LSM

with Prof. Andrew Mullhaupt

Ziqi Zhou

(Dated: October 17, 2025)

I. GUSTAFSSON'S LSM ALGORITHM CHOICES

In this section, we briefly describe and comment on Gustafsson's LSM algorithm implementation and the choices made therein.

We first briefly review the original LSM (per Gustafsson [1] Algorithm 2).

The most natural method to compute American put option prices is to compute the value $V(t, s)$ of the option backwards in time, using Black-Scholes equation or simulation techniques equivalent to the stock price behaviors underlying Black Sholes. In the original LSM, this is done by the following steps. First, geometric Brownian paths of the underlying asset(s) are directly generated. Using the martingale property in the continuation region (non-exercising), we know that the value $V(t, s)$ is the discounted expectation of future prices. Therefore, the discounted values can be incrementally carried back along a generated path if it does not locally cross the exercise boundary. However, the exercise payoff should be used if the asset crosses the exercise boundary.

Naively comparing the option value directly discounted along the path to the exercise payoff can be dangerous, because the geometric Brownian path may bring path with vary difference terminal payoff close to exercise boundary, leading to deviations from correct exercise decisions. Therefore, the LSM employs regression to fit the continuation values to estimate exercising behaviors better. In practice, the original LSM usually regresses on in-the-money continuation values, since out-of-the-money prices are never exercised.

The original LSM has some inherent sources of errors (Gustafsson [1]):

- Monte Carlo error, vanishes as the number of paths $\to \infty$;
- finite time increments, vanishes as $\mathrm{d}t \to \infty$;
- finite number of regression bases, vanishes as the basis become complete; since only few bases are used, the choice of bases is important.

These sources of errors can be reduced as more computation power is allocated to address each of them.

The original LSM has some other weaknesses:

- near expiration, the exercise boundary is adjacent to the out-of-the-money region and has a steep slope, which may be dangerous; we need to plot the exercise boundary to observe if it behaves well near expiration;

- computation resources (namely the regression part) are distributed across geometric Brownian paths; we may focus the resources more on where the value function behaves more non-trivially; namely
  - as the algorithm progresses away from expiration time, the exercise boundary goes further into the money; we may save computation power by regressing only on data far in the money;
  - the Monte Carlo is effectively integrating the martingale measure backwards, we therefore do not necessarily have to follow along particular instances of generated stock price paths.

Gustafsson's implementation of the LSM saves memory by generating the stock paths as the algorithm evaluates backwards in time instead of initializing and storing them throughout the runtime. This is possible through conditioning the backwards paths on $S(t = 0) = S_0$; in other words, these paths are implemented as (geometric) Brownian bridges backwards in time.

It is clear that, other than the reduction in memory, Gustafsson's implementation of the LSM has the same weaknesses as the basic LSM.

II. IMPLEMENTATION AND ASSESSMENTS GUSTAFSSON'S LSM

Our testing uses $S_0 = 90$, $K = 100$, $T = 1.0$, $r = 0.03$, $\sigma = 0.15$, 100 time steps, and LSM regression with 4 bases unless otherwise noted.

A. Preliminary Assessments

We independently implemented Gustafsson's LSM. Here, we present the preliminary results.

In Fig. 1, we observe that Gustafsson's LSM uses memory independent of the number of time steps. In contrast, the original LSM algorithm uses memory that grows with the number of time steps, significantly more than Gustafsson's improvement. This is consistent with Gustafsson [1].

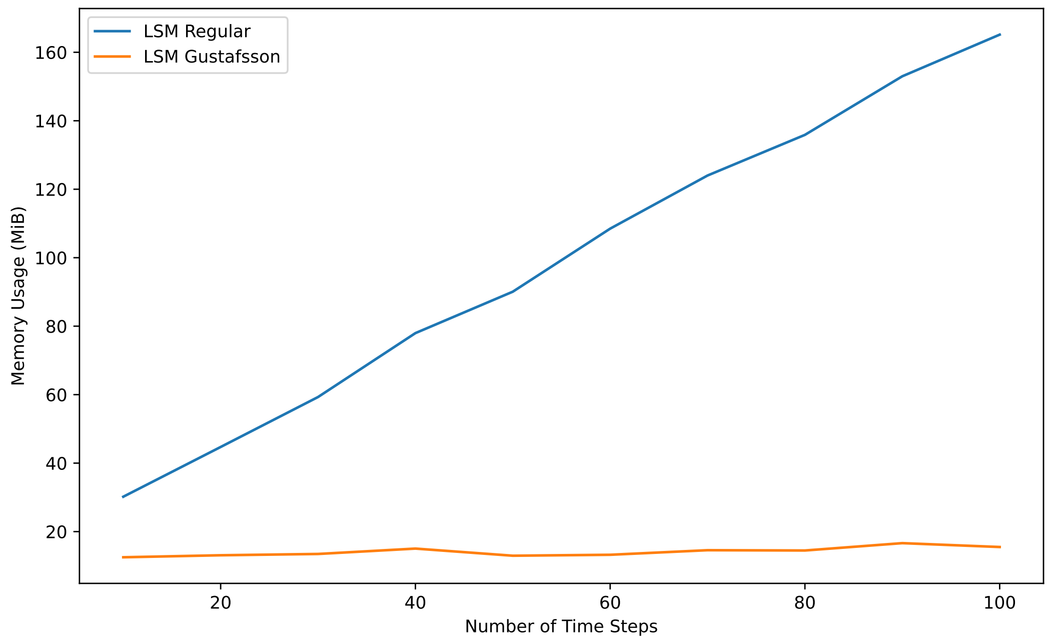


FIG. 1: Memory use of the original and Gustafsson's LSM

In Fig. 2, we compare Gustafsson's LSM versus the naive LSM (with pre-generated stock price paths and monomial basis). The values computed from the two algorithms and the quoted "true values" from Gustafsson's paper do not show significant differences. We therefore believe that our implementation of Gustafsson's LSM does not produce significant systematic errors other than those discussed in Section I.

Closely following Gustafsson's paper Section 3, we evaluate relative errors, run times, and simulated prices as we vary the number of time steps. Figs. 3-5 show similar behaviors as obtained by Gustafsson.

B. Exercise Boundary

In Fig. 6, we produced the exercise boundary for an example of Gustafsson's LSM run. In particular, we capture the least in-the-money stock price where the algorithm exercises at each time step.

There are several peculiarities regarding the simulated boundary.

Analytically, the exercise boundary for an American put option ends where the stock price

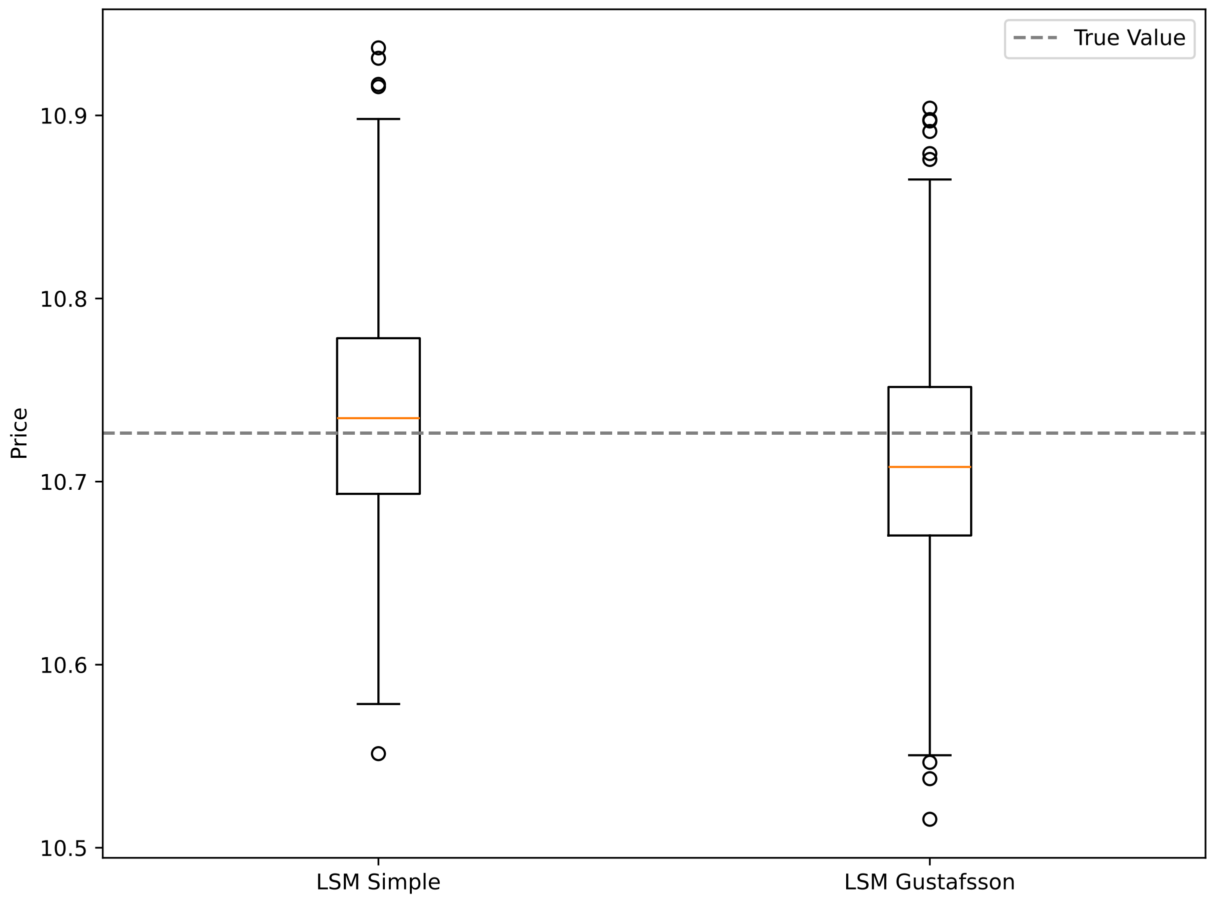
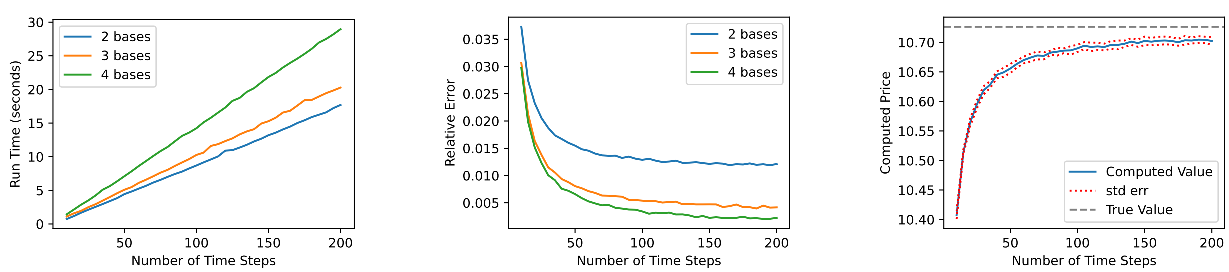


FIG. 2: Comparison of simple vs. Gustafsson's LSM with 10,000 paths sampling 1,000 runs



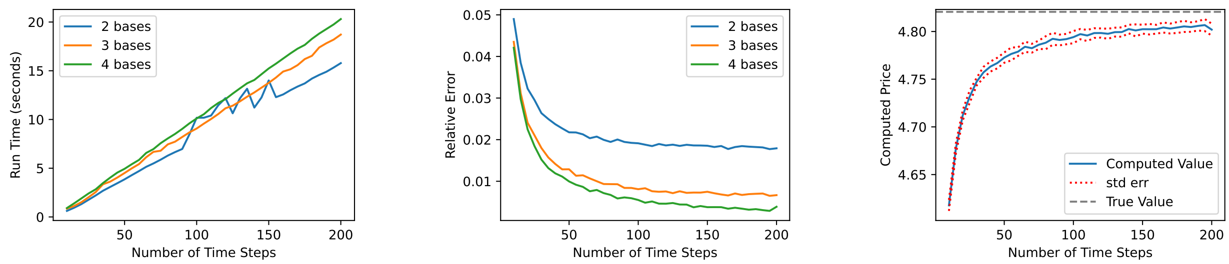
(a) run time

(b) relative error

(c) computed price

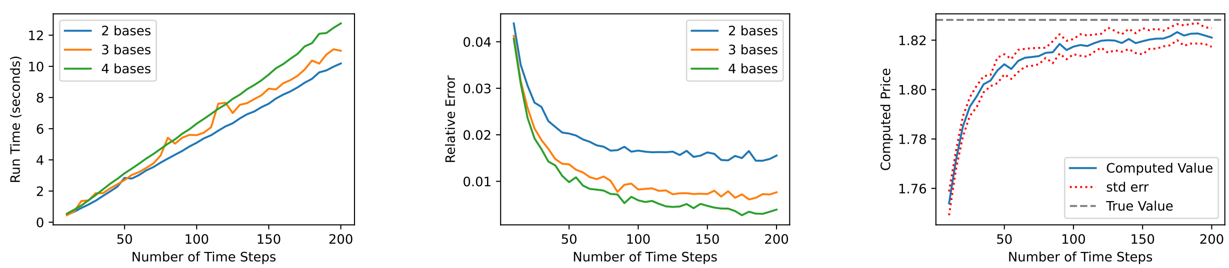
FIG. 3: Gustafsson's LSM at $S_0 = 90$ with 10^6 paths, 20 runs each, 10 to 200 time steps

equals the strike price $S = K$ at expiration. Close to expiration, the boundary price rises steeply until the option expires. We observe something slightly different in the simulation. The closest-to-expiry point captured in Fig. 6 is subtly distanced from $S = K$. We are not



(a) run time (b) relative error (c) computed price

FIG. 4: Gustafsson's LSM at $S_0 = 100$ with 10^6 paths, 20 runs each, 10 to 200 time steps



(a) run time (b) relative error (c) computed price

FIG. 5: Gustafsson's LSM at $S_0 = 110$ with 10^6 paths, 20 runs each, 10 to 200 time steps

sure whether this part of simulation correctly reflects the theoretical boundary at $t = T - \Delta t$, or it is an artifact of our simulation algorithm.

The simulation depiction of the exercise boundary also suffers close to $t = 0$. This is because our starting stock price is almost always away from the exercise boundary, and not enough simulated paths reach as far in-the-money as the exercise boundary. Therefore, the simulated boundary is not very reliable for $t \to 0$, especially when the number of paths is small, time t is too close to zero, or S_0 is far from the actual exercise boundary. The algorithm's pricing accuracy does not suffer from this effect, since the more relevant paths included in the simulation do not necessarily have to come close to the exercise boundary for $t \to 0$.

In addition to the discussion above, we may also take a closer look at an actual snapshot of how the LSM is making exercise decisions. Fig. 7a shows the continuation value data at $t = T/2$ for an example run. We observe that the fitted continuation value curve crosses the exercise value curve multiple times. In fact, the far in-the-money decisions are highly

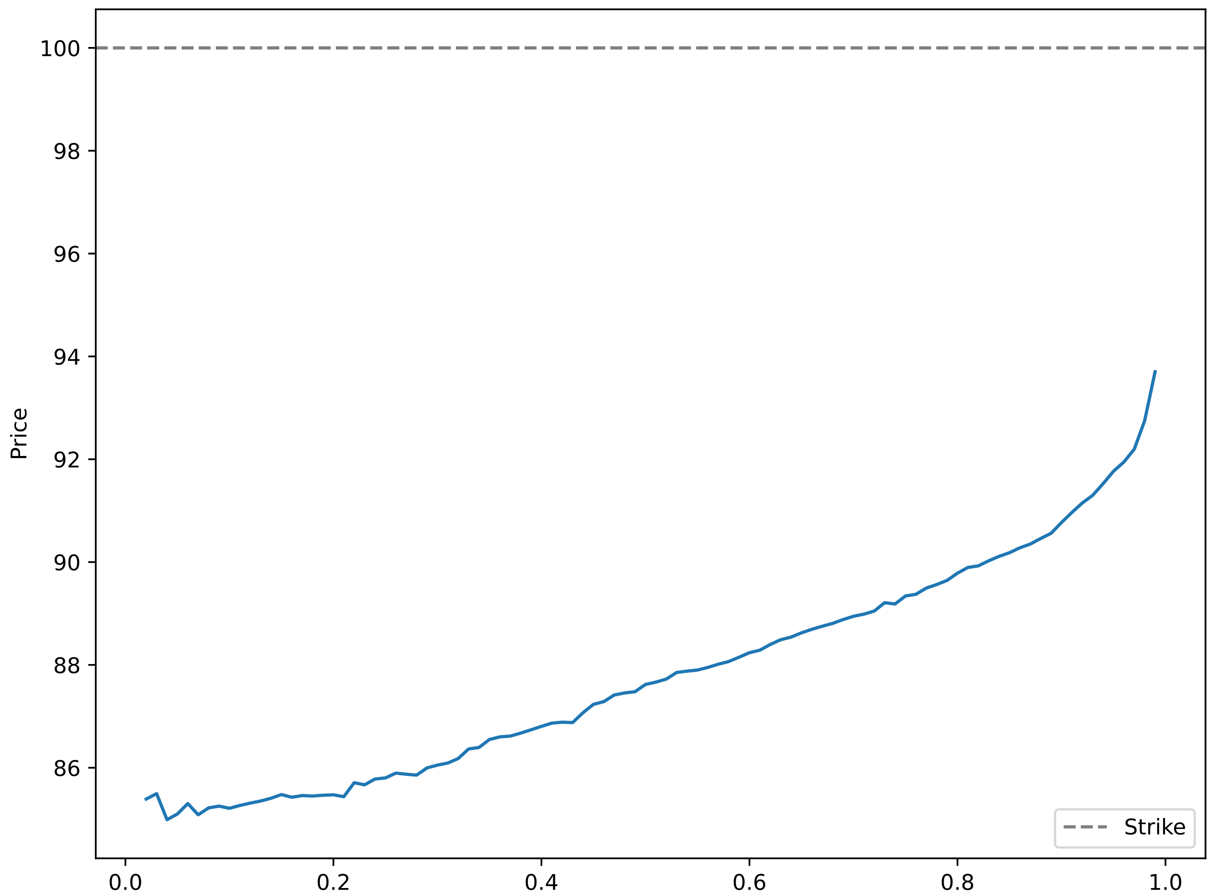
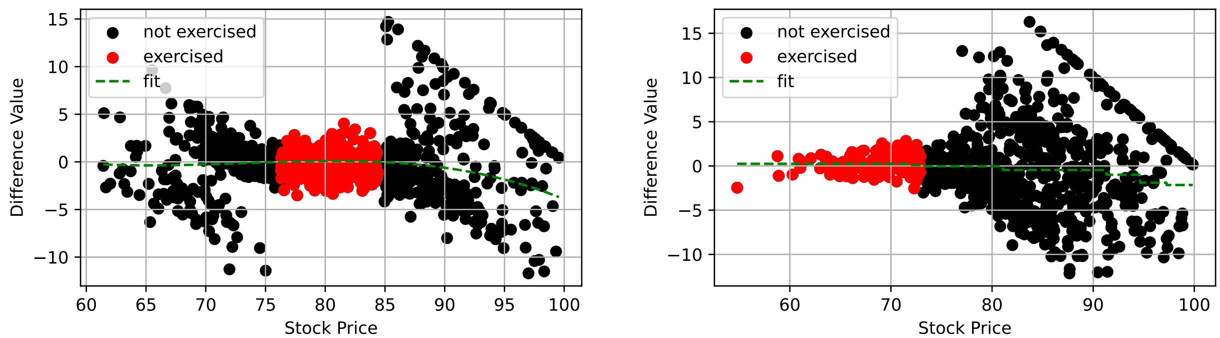


FIG. 6: LSM exercise boundary vs. time; plotted from a single run of 10^6 paths with 100 time steps

volatile on a run-to-run basis, often making inconsistent decisions. This means that the algorithm does not behave according to a practically coherent exercise price, but chooses to exercise or not as the difference of the exercise values and the fitted continuation values wiggle back and forth.

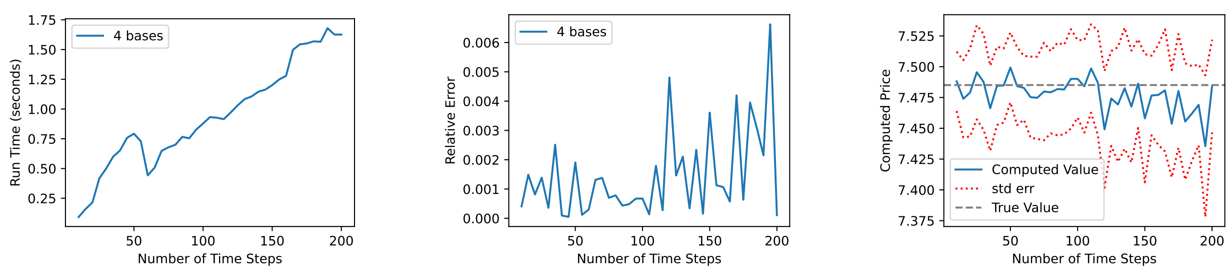
As Prof. Mullhaupt pointed out in the lectures, this unphysical behavior inspires a simple alternative to the naive LSM by using isotonic regression. Fig. 7b shows a snapshot from an LSM using centered isotonic regression, which clearly produces a unique exercise boundary price.



(a) lin. reg. with Laguerre basis

(b) centered isotonic regression

FIG. 7: LSM exercise decision snapshots; run for 1,000 paths and 100 time steps; the snapshots are taken at $t = 0.5$; the y-axis is the exercise value minus the continuation value



(a) run time

(b) relative error

(c) computed price

FIG. 8: Gustafsson's LSM at $S_0 = 100$ with 10^5 paths, 20 runs each, 10 to 200 time steps

C. American Call Option

Fig. 8 Shows a simulation of an American call option price using Gustafsson's LSM, compared to the analytic result.. We observe that Gustafsson's LSM produces call option values that agree with the analytical value. The run time data may be contaminated by the running environment.

Fig. 9 plots the exercise boundary for an example run of Gustafsson's LSM over an American call option. We observe an unphysical exercise simulated boundary far in-the-money. This can be explained since the chosen regression algorithm produces such systematic errors.

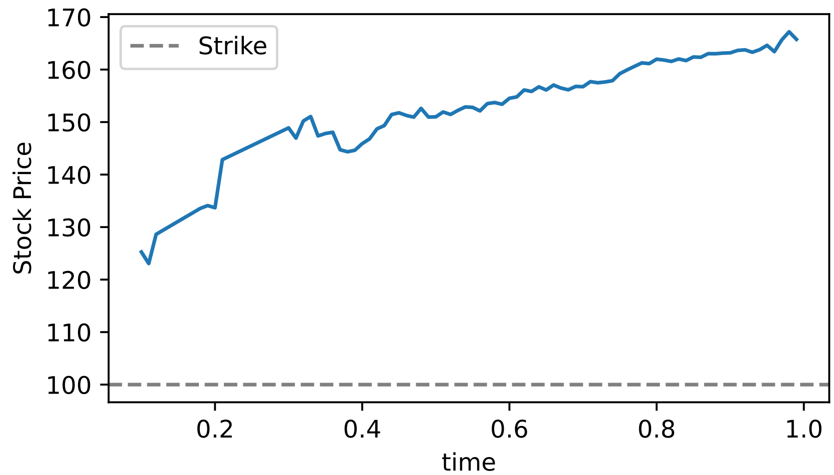


FIG. 9: LSM exercise boundary for American call ($S_0 = 100$); plotted from a single run of 10^6 paths with 100 time steps

D. Further Improvements

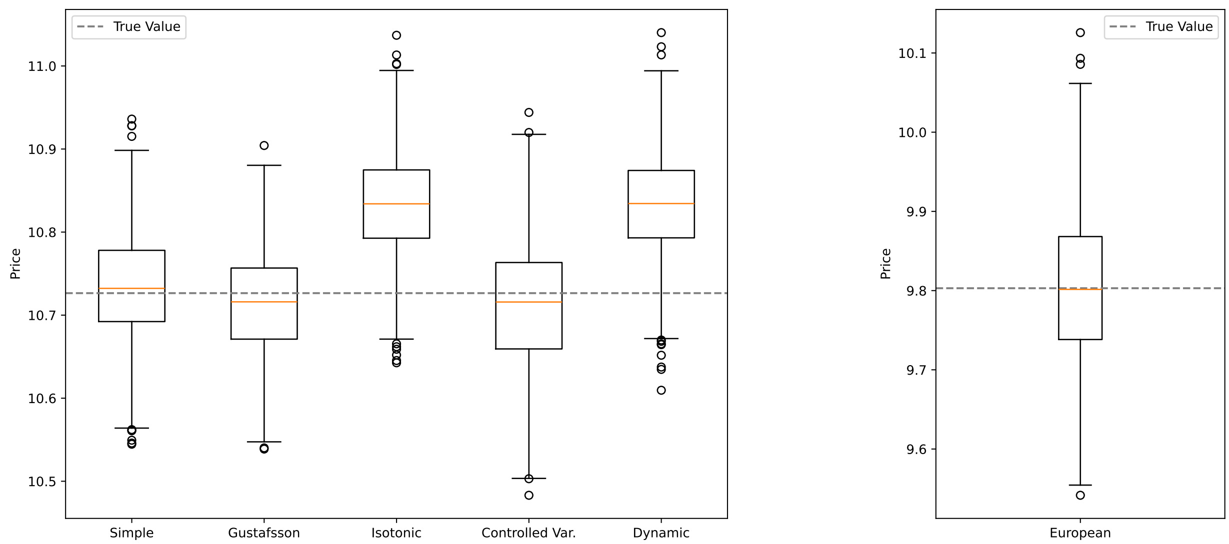
Since the regression component of the LSM algorithm is a source of systematic error, we may use regression algorithms that have more desirable behaviors in the regimes relevant for the LSM. Results are shown in Fig. 10a.

Comparing Fig. 10a with Fig. 10b, we observe that the sample variations of the American LSMs are not significantly greater than the European LSM. We conclude that the observed variations of the American put LSMs are dominated by the randomness in the generated paths. Improved estimation of exercise behaviors will **not** improve the **precision** of the algorithm.

We may also use controlled variate ideas to improve the algorithm. For instance, the American put and the European put are positively correlated, where the European put has an analytic solution. Therefore, we may use the European put as a controlled variate for the American put.

We have also experimented with replacing the regression with centered isotonic regression. The results indicate a bias, without significant improving the accuracy.

The exercise boundary may also be estimated given continuation values in each time step



(a) LSM American put prices

(b) LSM European put

FIG. 10: Comparison of LSM implementations with $10,000$ paths sampling $1,000$ runs; shown implementations: naive LSM, Gustafsson's LSM, LSM with centered isotonic regression, LSM using European option as controlled variate, LSM with "dynamically" determined exercise boundary; LSM on European included for accuracy comparison

by maximazing the sampled values with respect to the exercise boundary.

$$S_{\text{exercise estimate}} = \left\{ S_{\text{exercise}} \text{ that maximizes } \sum_{S_i \geq S_{\text{exercise}}} V_i + \sum_{S_i < S_{\text{exercise}}} (K - S_i) \right\}. \tag{1}$$

The LSM that makes exercise decisions this way (labeled "dynamic" in Fig. 10a) behaves similarly to the LSM centered isotonic regression. We suspect that they might effectively be the same algorithm.

III. CONCLUDING REMARKS

Our evaluation of Gustafsson's version of the LSM confirms the overall reliability of the LSM and Gustafsson's improvement on memory efficiency.

Further investigation reveals an inherent unphysical behavior of the regression component of the LSM. The exercise simulated boundary is affected by the unphysical multiple crossing of the fitted continuation values and the exercise values. This issue, however, does not introduce significant bias, because the fitted continuation values and the exercise values are extremely close in the regime where the exercise boundary price is biased. This issue may be resolved if a smarter basis is used for the regression or a better regression algorithm is employed. The variation in the estimated prices is dominated by the randomness in the generated stock price paths.

Our alternative attempts with isototic regression or controlled variate method do not significantly improve the performance of the LSM.

[1] W. Gustafsson, Evaluating the longstaff-schwartz method for pricing of american options (2015).