

Prissætning af amerikanske optioner med neurale netværk

Lukas Rønman Behr & Mads Emil Christensen

University of Copenhagen

Department of Science

Juni 2023

Abstract

This paper addresses the issues of pricing american style options within Black Scholes theory. We investigate whether the Least Squares Monte Carlo method (LSM) for option pricing can be improved by introducing machine learning. The binomial tree model will be used as reference for this comparison.

First we lay the theoretical foundations for understanding the financial markets, option type derivatives and MLP neural networks.

The three models LSM, LSM-MLP and B-MLP, which we use for option pricing are then introduced after which they are extensively investigated with various numerical tests.

Our numerical results show that the LSM-MLP model is both inaccurate and unstable, indicating that the LSM method can not be improved by machine learning. However, the B-MLP implementation of machine learning into option pricing showed promising results that suggest a model of this type could be a viable pricing method.

Indholdsfortegnelse

1	Indledning	3
2	Teori	3
2.1	De finansielle markeder	3
2.2	De matematiske rammer	4
2.3	Europæiske optioner - Black Scholes model	5
2.4	Amerikanske optioner	6
2.5	Binomial modellen	7
2.6	LSM modellen	10
3	Numeriske resultater	14
3.1	Korrekthed af implementeringen	14
3.2	Udvalg af parametre	16
3.3	LSM og binomial sammenligning	19
4	Neurale netværk	20
4.1	Multi Layer Perceptron	20
4.2	LSM-MLP	22
4.3	Binomial-MLP	24
4.4	Resultater for amerikansk put option	26
5	Optioner på to aktiver	27
5.1	Europæiske minimum optioner - Black Scholes model	27
5.2	Binomial modellen i to dimensioner	28
5.3	Korrekthed af implementeringen	32
5.4	LSM i to dimensioner	33
5.5	LSM-MLP i to dimensioner	34
5.6	Sammenligning af forskellige designs	35
5.7	B-MLP i to dimensioner	36
5.8	Resultater for amerikansk put minimum option	37
6	Delta hedging	40
6.1	Delta hedging metode	40
6.2	Hedging eksperiment	42
6.3	Hedging resultater	44
7	Diskussion	45
8	Konklusion	47

1 Indledning

Optioner er blandt de mest handlede derivater på de finansielle markeder. Prissætningen af disse er derfor et emne, der ofte er blevet undersøgt i litteraturen.

I dette projekt arbejder vi inden for Black Scholes teorien. Europæiske optioner har her en lukket løsning. Deres amerikanske modpart har ingen lukket løsning, hvorfor prissætningen af disse er mere kompliceret af natur.

Vi vil undersøge prissætningen af amerikanske optioner med LSM algoritmen, som er en metode, der numerisk approksimerer prisen på amerikanske optioner ved hjælp af Monte Carlo simulerede aktiekurser. Vi vil desuden integrere et neuralt netværk til algoritmen og undersøge, hvorvidt machine learning kan forbedre algoritmens performance.

Vi introducerer ydermere en prissætnings model, der bygger udelukkende på et neuralt netværk og store datasæt.

Modellerne vil blive evalueret på baggrund af deres nøjagtighed og computationstid, hvor vi benytter den amerikanske optionspris fra binomial modellen som reference.

Derudover vil modellerne blive testet på et hedging eksperiment for at undersøge, hvor godt de performer out of sample.

2 Teori

2.1 De finansielle markeder

De finansielle markeder er markeder, hvor køber og sælger mødes for at handle instrumenter såsom aktier, obligationer, valuta, råvarer og derivater. I dette projekt vil vi fokusere på aktier og derivater. En aktie er en ejerandel af en virksomhed. Derivater er finansielle instrumenter, hvis pris er afledt af et underliggende aktiv, i dette tilfælde en aktie. En type af derivater er optioner.

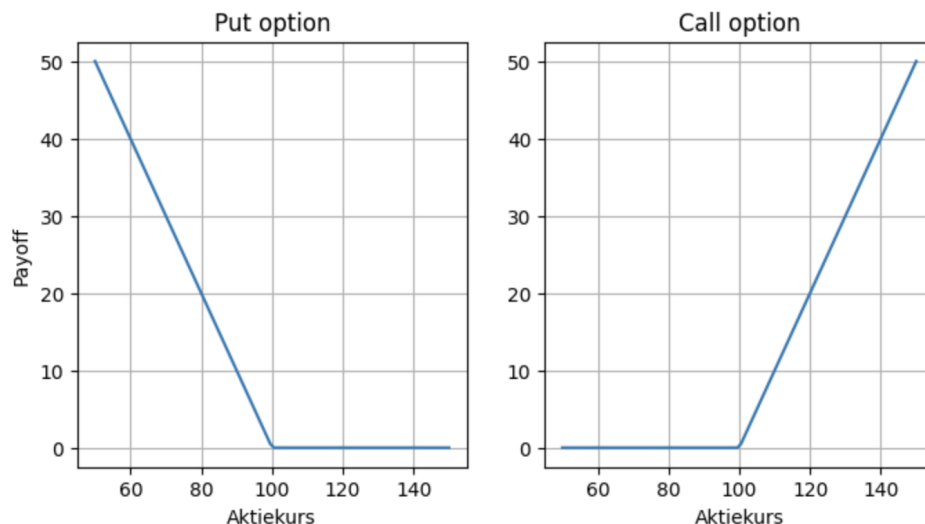
En option er en kontrakt, der giver indehaveren en uforpligtende rettighed til at købe (call option)/sælge (put option) det underliggende aktiv til en fastsat pris (strike). Når det underliggende aktiv bliver købt/solgt, bliver optionen indfriet.

Europæiske optioner er optioner, der kun kan indfries til tidspunkt T ved udløb af kontrakten. De har altså følgende payoff funktion Φ , der afhænger af aktiekursen ved udløb S_T og strike K :

$$\text{Europæisk put:} \quad \Phi(S_T) = (K - S_T)^+$$

$$\text{Europæisk call:} \quad \Phi(S_T) = (S_T - K)^+$$

Payoff for hhv. en europæisk put og call option til tid T med strike $K = 100$ er illustreret i figur 1.



Figur 1: Payoff funktion for europæisk put option og call option med strike $K = 100$.

En anden type option er den amerikanske option. Amerikanske optioner har ligeledes en strike K og et udløbstidspunkt T . Forskellen ligger i, at den amerikanske option kan indfries til ethvert tidspunkt $t \in [0, T]$.

Udover disse to typer optioner eksisterer der mere eksotiske typer af optioner, der f.eks. kan afhænge af flere underliggende aktiver eller gennemsnitskurser henover en løbetid. En af disse bliver introduceret senere.

I de næste afsnit vil vi fremlægge prissætningen af både de europæiske og amerikanske optioner.

2.2 De matematiske rammer

Inden optionerne kan prissættes, skal vi opsætte de matematiske rammer for de finansielle markeder. Vi tager det underliggende sandsynlighedsrum (Ω, \mathcal{F}, P) og endelige tidshorisont $t \in [0, T]$, hvor Ω er alle mulige stokastiske udfald i $[0, T]$, for givet. \mathcal{F} er vores σ -algebra af alle hændelser til tid T . Vi definerer filtreringen $F = \{\mathcal{F}_t \mid t \in [0, T]\}$ af prisudviklingerne i aktiverne på det finansielle marked. 6, p. 120

Om markedet antages det, at der eksisterer mindst et risikofyldt aktiv samt et risikofrit aktiv i form af f.eks. en bankbog. Om markedet antages der yderligere:

- Short-positioner¹ er tilladt, og instrumenter kan handles i brøkdele
- Der er intet "bid-ask spread", dvs. salgsprisen er lig med købsprisen
- Der er ingen transaktionsomkostninger

¹En short-position består i at låne et aktiv og herefter sælge det på det åbne marked. En short-position er profitabel, hvis man kan købe og aflevere aktivet billigere tilbage på et fremtidigt tidspunkt.

- Markedet er fuldstændig likvidt, dvs. det er muligt at købe/sælge ubegrænsede mængder på markedet. Der kan lånes et ubegrænset beløb i det risikofrie aktiv
- Der er ingen arbitrage muligheder i markedet

1, p. 8

Om aktiverne antages følgende:

- Den risikofrie rente r og volatiliteten σ er konstante
- Aktien følger en geometrisk brownian motion
- Aktien er ikke udbytte-betalende

2, p. 640

Dynamikken i de to aktiver er givet ved følgende differentialligninger:

$$\begin{aligned}dB_t &= rB_t dt \\ dS_t &= rS_t dt + \sigma S_t dW_t\end{aligned}$$

hvor W_t er en Brownian motion. En Brownian motion er defineret som:

- $W_0 = 0$
- W har uafhængige tilvækster
- $W_{t+\Delta t} - W_t$ er normalfordelt med middelværdi 0 og varians Δt
- W_t er kontinuert i t

1, p. 43

I overensstemmelse med antagelsen om ingen arbitrage eksisterer der et risikoneutralt mål Q .

Med disse rammer kan vi nu prissætte optioner.

2.3 Europæiske optioner - Black Scholes model

Den universelle anerkendte prissætningsmodel til europæiske optioner er Black Scholes modellen opstillet af Myron Scholes og Fisher Black i 1973 [2]. Modellen bruger alle rammerne fra afsnit 2.2, og ydermere gælder det, at optionen kun kan blive indfriet til udløb, altså at den er europæisk. Black Scholes modellen siger, at prisen på en europæisk option er givet ved:

$$F(t, S_t) = e^{-r(T-t)} \mathbb{E}^Q[\Phi(S_T)] \quad (2.1)$$

Løsningen og dermed prisen af en europæisk call option er da givet ved Black Scholes formelen.

Proposition 1 (Black Scholes formel:). *Prisen på en europæisk call option med strike K og tid til udløb T er givet ved formelen $c(t, S_t)$, hvor*

$$c(t, S_t) = S_t N[d_1(t, S_t)] - e^{-r(T-t)} K N[d_2(t, S_t)]$$

Her angiver N fordelingsfunktionen for standard normalfordelingen $\mathcal{N}(0,1)$ og

$$d_1(t, S_t) = \frac{1}{\sigma\sqrt{T-t}} \left(\ln \left(\frac{S_t}{K} \right) + \left(r + \frac{1}{2}\sigma^2 \right) (T-t) \right),$$

$$d_2(t, S_t) = d_1(t, S_t) - \sigma\sqrt{T-t}$$

1, p. 109

Dette bestemmer kun prisen for en europæisk call option. Prisen på en europæisk put option kan bestemmes med "Put-call pariteten".

Proposition 2 (Put-call paritet): Givet en europæisk call og put option med samme strike K og tid til udløb T med prisfunktioner givet ved hhv. $c(t, S_t)$ og $p(t, S_t)$, gælder følgende relation:

$$p(t, S_t) = Ke^{-r(T-t)} + c(t, S_t) - S_t$$

1, p. 139

Altså kan man givet prisen på den europæiske call option også udlede prisen på den europæiske put option med samme strike og udløb.

Black scholes formelen bruges til at bestemme en reference pris på europæiske optioner, som vil blive benyttet til at sikre korrektheden af implementering af binomial modellen og Monte Carlo simuleringen, hvilke vil blive introduceret i et senere afsnit.

2.4 Amerikanske optioner

Vi ønsker at bestemme prisen på amerikanske optioner, som ikke har en lukket løsning. I modsætning til den europæiske modpart er prisen på en amerikansk option udledt af den optimale indfrielsesstrategi, givet filtreringen F .

Til udløb T indfries optionen, hvis den er in the money, altså hvis indfrielsesværdien er større end 0, ellers lader man optionen udløbe værdiløs. Til tidspunkt t inden udløb skal optionsholderen vurdere, om han skal indfri eller holde optionen ud fra den umiddelbare indfrielsesværdi og den forventede fortsættelsesværdi. Man skal altså løse et optimal indfrielsesproblem. Vi definerer

$$\mathcal{T}_t = \{\tau \mid t \leq \tau \leq T\} \quad (2.2)$$

som er mængden af alle indfrielsestidspunkter τ henover perioden mellem beslutningstidspunktet t og optionens udløbstidspunkt T . Det optimale indfrielsesproblem, der skal løses er da:

$$\sup_{\tau \in \mathcal{T}_t} \left(\mathbb{E}^Q [e^{-r(\tau-t)} \Phi(S_\tau) | \mathcal{F}_t] \right) \quad (2.3)$$

I praksis lader man ikke t være en kontinuert variabel, men diskret hvor mængden af alle diskreterings-tidspunkter er givet ved $\mathbb{T} = \{t_0 = 0, t_1, \dots, t_k, \dots, t_K = T\}$ for $\{0, 1, \dots, K\} \subseteq \mathbb{N}_0$. Det gælder her, at $\mathcal{T}_t \subseteq \mathbb{T}$.

Robert Merton viste i 1973, at fortsættelsesværdien af en amerikansk call option altid vil være større eller lig med indfrielsesværdien, hvis det underliggende aktiv ikke betaler udbytte. En amerikansk

call option på en ikke udbytte-betalende aktie vil da altid holdes til udløb, og derfor har den samme pris som en europæisk call.[7] Grundet dette er fokus i vores projekt at prissætte amerikanske put optioner.

I de kommende afsnit introducerer vi to forskellige modeller til at prissætte amerikanske optioner med.

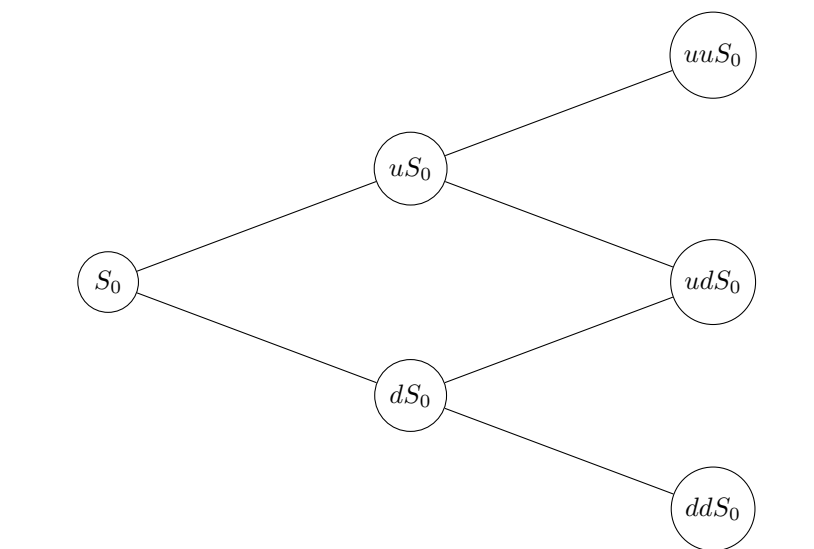
2.5 Binomial modellen

Binomial modellen blev formaliseret af John Cox, Stephen Ross og Mark Rubenstein i 1979 [4].

I deres model er kursudviklingen af et aktiv opbygget som et træ. Antagelsen er, at kursen S på et aktiv til næste periode er givet på følgende måde:

$$S_{t+1} = \begin{cases} uS_t & \text{med sandsynlighed } q \\ dS_t & \text{med sandsynlighed } 1 - q \end{cases}$$

Altså udvikler kursen sig enten med en faktor u eller d . Yderligere gælder det, at $u \cdot d \cdot S = d \cdot u \cdot S$, hvilket skaber et kompakt træ under udviklingen af aktiekurserne. Et eksempel på et træ med to perioder kan ses herunder.



Figur 2: To-periode binomial model med et risikofyldt aktiv

Denne opbygning gør det også muligt at finde kursen på aktien til et hvilket som helst tidspunkt givet antallet af op-og nedryk. Aktiens kurs kan dermed beskrives ved $S = S_0 u^k d^j$, hvor k beskriver antal opryk, og j beskriver antal nedryk. Derudover er u og d defineret således:

$$u = e^{\sigma\sqrt{\Delta t}} \tag{2.4}$$

$$d = e^{-\sigma\sqrt{\Delta t}} \tag{2.5}$$

Altså kan kursen til et givent tidspunkt beskrives ved følgende.

$$S = S_0 u^{k-j} \quad (2.6)$$

Dette gør det til en meget effektiv metode at bruge, når det kommer til beregningstid og hukommelse, da hele træet ikke behøver at blive genereret for at kende de sidste knuder i træet.

Sandsynlighedsmålet i modellen q er defineret ved:

$$q = \frac{R - d}{u - d} \quad (2.7)$$

$$1 - q = \frac{u - R}{u - d} \quad (2.8)$$

4, p. 234

Dette er det risikoneutrale sandsynlighedsmål, da det opfylder følgende lighed.

$$qu + (1 - q)d = \frac{R - d}{u - d}u + \frac{u - R}{u - d}d = R$$

Her er det givet, at $R = e^{r \cdot \Delta t}$, hvor r beskriver renten på det risikofrie aktiv i modellen. $\Delta t = \frac{T}{n}$ beskriver størrelsen af afstanden mellem to perioder, hvor n er størrelsen på træet og T er antal år, som træet strækker sig over.

For hver knude i træet må nutidsværdien af den forventede værdi til næste periode være givet ved

$$\frac{\mathbb{E}^q[S_{t+\Delta t} \mid S_t]}{R} = (quS_t + (1 - q)dS_t) \cdot \frac{1}{R} \quad (2.9)$$

Med alt dette er det muligt at beregne prisen for en option indenfor modellens rammer. Ligningerne (2.4), (2.5) og (2.6) bruges til at regne aktiekursen til periode T . Payoff regnes så for optionen med $\Phi(S_T)$. Herefter trævles baglæns, hvor der for hver knude igen regnes payoff og den forventede værdi til næste periode. Hvis det er en europæisk option, bliver knudens værdi altid den forventede fortsættelsesværdi, da optionen ikke kan indfries før tid. Dette kan ses i ligning (2.1). Hvis derimod det er en amerikansk option, skal der ved hver knude vurderes om det nuværende payoff er større eller mindre end den forventede fortsættelsesværdi. Lad $V_{k,t}$ beskrive værdien af en knude med k opryk til tidspunkt t . Altså bliver hver knudes værdi følgende:

$$\text{Europæisk option: } V_{k,t} = F(t, S_t) = \frac{\mathbb{E}^q[\Phi(S_T)]}{R^{T-t}} \quad (2.10)$$

$$\text{Amerikansk option: } V_{k,t} = \max(\Phi(S_t), \frac{\mathbb{E}^q[V_{t+\Delta t} \mid V_{k,t}]}{R}) \quad (2.11)$$

hvor

$$\mathbb{E}^q[V_{t+\Delta t} \mid V_{k,t}] = qV_{k+1,t+\Delta t} + (1 - q)V_{k,t+\Delta t} \quad (2.12)$$

Ved denne algoritmes afslutning opnåes det så, at t_0 prisen for optionen er givet ved den første knude af træet.

I dette projekt bruges binomial modellens pris på den amerikanske put som den vejledende pris på optionen. Argumentet for denne tilgang kommer af, at binomial modellen konvergerer mod Black

Scholes modellens aktieudvikling, når $\Delta t \rightarrow 0$. Altså gælder det, at den amerikanske optionspris beregnet med modellen konvergerer mod den sande pris for $n \rightarrow \infty$.^[4]

I praksis implementeres algoritmen på følgende vis:

Data:

S_0 : Start pris, r : Risikofri rente, σ : Volatilitet, n : Antal perioder, T : År til udløb,

K : Strike, EU/AM : Optionens type

Result: $P[0]$: Optionens pris

$P \leftarrow \underbrace{(0,0,\dots,0)}_{n+1};$

$\Delta t \leftarrow \frac{T}{n};$

$R \leftarrow e^{r\Delta t};$

$u \leftarrow e^{\sigma\sqrt{\Delta t}};$

$d \leftarrow \frac{1}{u};$

$q \leftarrow \frac{R-d}{u-d};$

for $i \leftarrow 0$ **to** n **do**

$P[i] \leftarrow \max(K - S_0 u^{2i-n}, 0)$

end

for $j \leftarrow n-1$ **to** 0 **do**

for $i \leftarrow 0$ **to** j **do**

$P_{exercise} \leftarrow \max(K - S_0 u^{2i-j}, 0);$

$P_{fortsæt} \leftarrow q \cdot P[i+1] \cdot \frac{1}{R} + (1-q) \cdot P[i] \cdot \frac{1}{R};$

if EU **then**

$P[i] \leftarrow P_{fortsæt};$

else

$P[i] \leftarrow \max(P_{fortsæt}, P_{exercise});$

end

end

end

return $P[0]$

Algoritme 1: Binomial modellen for put option

2.6 LSM modellen

En måde at estimere amerikanske optionspriser på er ved at simulere mange forskellige stier af udviklingen i det underliggende aktiv og anvende en algoritme ved navn LSM². Modellen blev introduceret i 2001 af Francis Longstaff og Eduardo Schwartz[6]. Dette afsnit vil introducere modellen.

2.6.1 Monte Carlo simulering

Vi har tidligere antaget, at dynamikken i risikofyldte aktiver er givet ved den stokastiske differential ligning:

$$dS_t = rS_t dt + \sigma S_t dW_t$$

Hvor løsningen til ligningen er givet ved:

$$S_t = S_0 e^{(r - \frac{1}{2}\sigma^2)t + \sigma W_t}$$

Her gælder det, at $W_t \sim \mathcal{N}(0, t)$. Dette kan omskrives til:

$$S_{t+\Delta t} = S_t e^{(r - \frac{1}{2}\sigma^2)\Delta t + \sigma W_{\Delta t}} \quad (2.13)$$

Med denne løsning kan vi nu simulere udviklingen i aktiekursen for vilkårligt mange diskreteringspunkter og antal stier.

Da hver sti har en stokastisk variabel, medfører dette en usikkerhed i Monte Carlo simuleringen. For at mindske denne er det vigtigt, at man simulerer mange stier. Hvor mange stier, der skal simuleres, for at sikre en stabil og nøjagtig pris vil vi belyse senere. I det næste afsnit kigger vi på variansen af simuleringen og en mulig måde at gøre den mindre på.

2.6.2 Antitetisk varians reduktion

Variansen for gennemsnittet mellem to uafhængigt simulerede stier er givet ved:

$$Var\left(\frac{1}{2}(S_t + \bar{S}_t)\right) = \frac{Var(S_t) + Var(\bar{S}_t) + 2Cov(S_t, \bar{S}_t)}{4} \quad (2.14)$$

\Updownarrow

$$Var\left(\frac{1}{2}(S_t + \bar{S}_t)\right) = \frac{Var(S_t) + Var(\bar{S}_t)}{4} \quad (2.15)$$

Da de to stier er uafhængige, vil kovariansen mellem de to være 0.

For at mindske variansen introduceres antitetisk varians reduktion til simuleringen. Dette er et redskab, som kan bruges til at reducere variansen for simuleringen og formindske beregningstiden for algoritmen. Der tages udgangspunkt i ligning (2.13). For hver sti, der simuleres, trækkes $W_{\Delta t} \sim \mathcal{N}(0, \Delta t)$. Eftersom normalfordelingen er symmetrisk omkring 0, gælder det, at $W_{\Delta t} \sim \bar{W}_{\Delta t}$ for $W_{\Delta t} = -\bar{W}_{\Delta t}$. Med antitetisk varians reduktion udnyttes dette ved at skabe en ”spejl” sti for hver

²Least Squares Monte Carlo

sti ved at bruge $\bar{W}_{\Delta t}$ til at simulere aktieprisen med.

Vi bestemmer kovariansen mellem en sti og sin antitetiske sti ligesom i ligning (2.14)

$$\begin{aligned} Cov(S_t, \bar{S}_t) &= \mathbb{E}(S_t \cdot \bar{S}_t) - \mathbb{E}(S_t)\mathbb{E}(\bar{S}_t) \\ &\Downarrow \\ Cov(S_t, \bar{S}_t) &= \mathbb{E}(S_0^2 e^{2(r-\frac{\sigma^2}{2})t + \sigma W_t + \sigma \bar{W}_t}) - S_0^2 e^{2rt} [8] \end{aligned}$$

Siden $W_t = -\bar{W}_t$, må følgende gælde

$$\begin{aligned} Cov(S_t, \bar{S}_t) &= \mathbb{E}(S_0^2 e^{2(r-\frac{\sigma^2}{2})t}) - S_0^2 e^{2rt} \\ &\Downarrow \\ Cov(S_t, \bar{S}_t) &= S_0^2 e^{2rt - \sigma^2 t} - S_0^2 e^{2rt} < 0 \end{aligned}$$

Altså må det gælde, at variansen af gennemsnittet mellem to antitetiske kurser er mindre end for to uafhængigt simulerede kurser. Dette er altså en fordelagtig metode for at øge præcisionen af simuleringen og derved præcisionen af optionsprisen. Derudover mindsker denne teknik også beregningstiden, da der kun skal trækkes halvt så mange værdier fra normalfordelingen for at opnå det samme antal stier.

I praksis simuleres aktiestierne på følgende vis:

Data:

S_0 : Start pris, r : Risikofri rente, σ : Volatilitet, n : Antal stier, T : År til udløb,

Δt : Afstanden $t_{k+1} - t_k$ i år ($\frac{1}{50}$)

Result: Simulerede stier

perioder $\leftarrow \frac{T}{\Delta t} + 1$;

$$\text{stier} \leftarrow n \left\{ \underbrace{\begin{pmatrix} S_0 & 0 & \dots & 0 \\ S_0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ S_0 & 0 & \dots & 0 \end{pmatrix}}_{\text{perioder}} \right\};$$

for $i \leftarrow 0$ **to** $\frac{n}{2} - 1$ **do**

for $j \leftarrow 1$ **to** perioder $- 1$ **do**

$W_{\Delta t} \leftarrow \mathcal{N}(0, \Delta t)$;

$\bar{W}_{\Delta t} \leftarrow -W_{\Delta t}$;

$\text{stier}[i, j] \leftarrow \text{stier}[i, j-1] e^{(r-\frac{1}{2}\sigma^2)\Delta t + \sigma W_{\Delta t}}$;

$\text{stier}[i + \frac{n}{2}, j] \leftarrow \text{stier}[i + \frac{n}{2}, j-1] e^{(r-\frac{1}{2}\sigma^2)\Delta t + \sigma \bar{W}_{\Delta t}}$

end

end

return stier

Algoritme 2: Monte carlo simulering med antitetisk varians reduktion

2.6.3 LSM algoritmen

Ud fra de simulerede stier forsøger LSM algoritmen at løse indfrielsesproblemet:

$$\sup_{\tau \in \mathcal{T}_t} \left(\mathbb{E}^Q [e^{-r(\tau-t)} \Phi(S_\tau) | \mathcal{F}_t] \right)$$

Dette gøres ved at til hvert indfrielsestidspunkt τ_k bestemme den forventede værdi opnået ved at holde optionen frem til tidspunkt τ_{k+1} . Denne fortsættelsesværdi bestemmes med en least squares estimator.

Lad $C(\omega, \tau_k, T)$ betegne cashflow stien genereret af optionen for hver simuleret sti ω til indfrielsestidspunkt τ_k , betinget af optionen ikke er indfriet inden τ_k , og at optionsholderen følger den optimale indfrielsesstrategi frem til tidspunkt T .

Startende rekursivt i τ_{K-1} , hvor $K-1$ betegner det sidste indfrielsestidspunkt inden udløb $\tau_K = T$, fitter man nutidsværdien af $C(\bar{\omega}, \tau_K, T)$ på $S_{\bar{\omega}, \tau_{K-1}}$. Her beskriver $\bar{\omega}$ de stier, hvor optionen er in the money til tidspunkt τ_{K-1} . På denne vis trævler man tilbage til tid 0 og finder den forventede fortsættelsesværdi i alle indfrielsestidspunkter τ givet ved:

$$\mathbb{E}^Q [e^{-r(\tau_{k+1}^* - \tau_k)} C(\bar{\omega}, \tau_{k+1}, T) | S_{\bar{\omega}, \tau_k}] \approx \beta_0 + \beta_1 \cdot S_{\bar{\omega}, \tau_k} + \beta_2 \cdot S_{\bar{\omega}, \tau_k}^2 \quad (2.16)$$

hvor τ_{k+1}^* beskriver det optimale indfrielsestidspunkt i mellem τ_{k+1} og T .

Indfrielsesstrategien for hver sti findes da ved, at i hvert indfrielsestidspunkt evaluere indfrielsesværdien og den tilbagediskonterede fortsættelsesværdi:

$$\begin{cases} \tau_K^* = T \\ \tau_k^* = \tau_k \cdot \mathbf{1}_{\{\Phi(S_{\bar{\omega}, \tau_k}) \geq \mathbb{E}^Q [e^{-r(\tau_{k+1}^* - \tau_k)} C(\bar{\omega}, \tau_{k+1}, T) | S_{\bar{\omega}, \tau_k}]\}} + \tau_{k+1}^* \cdot \mathbf{1}_{\{\mathbb{E}^Q [e^{-r(\tau_{k+1}^* - \tau_k)} C(\bar{\omega}, \tau_{k+1}, T) | S_{\bar{\omega}, \tau_k}] > \Phi(S_{\bar{\omega}, \tau_k})\}} \end{cases} \quad (2.17)$$

Optionens pris er da givet ved at tilbagediskontere $C(\omega, \tau_0, T)$ for alle stier ω og tage gennemsnittet af dette:

$$\frac{1}{N} \sum_{i=0}^N C(\omega_i, \tau_0, T) \cdot e^{-r\tau_{i,0}^*}$$

[6]

Proposition 3. Lad $LSM(\omega_i, M, K)$ betegne prisen for den i 'te sti på en amerikansk type option, med K indfrielsestidspunkter bestemt med LSM algoritmen med M basisfunktioner, da gælder følgende:

$$V(x) \geq \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N LSM(\omega_i, M, K)$$

hvor $V(x)$ betegner den sande pris på den amerikanske option.

Rationalet bag denne proposition er, at indfrielsesstrategien bag algoritmen er suboptimal. Dette kan også vises for optioner, som afhænger af flere aktiver. For bevis se 6, p. 124.

I praksis implementeres algoritmen på følgende vis:

Data:

stier: Simulerede stier, r : Risikofri rente, K : Strike, T : År til udløb

Result: Optionens pris

n , perioder $\leftarrow \text{shape}(\text{stier})$;

$\Delta t \leftarrow \frac{T}{\text{perioder}-1}$;

$R \leftarrow e^{r\Delta t}$;

cash_flow $\leftarrow \underbrace{(0,0,\dots,0)}_n$;

for $i \leftarrow 0$ **to** $n-1$ **do**

 | cash_flow[i] $\leftarrow \max(K - \text{stier}[i, \text{perioder}-1], 0) \cdot \frac{1}{R}$;

end

for $j \leftarrow \text{perioder}-2$ **to** 0 **do**

 temp_cash_flow $\leftarrow \underbrace{(0,0,\dots,0)}_n$;

$X \leftarrow ()$;

$Y \leftarrow ()$;

for $i \leftarrow 0$ **to** $n-1$ **do**

 temp_cash_flow[i] $\leftarrow \max(K - \text{stier}[i,j], 0)$;

if temp_cash_flow[i] > 0 **then**

 | X **append** stier[i,j];

 | Y **append** cash_flow[i];

else

 | **continue**

end

end

$f(x) \leftarrow \text{fit}(Y = \beta_1 X^2 + \beta_2 X + \beta_0)$;

for $i \leftarrow 0$ **to** $n-1$ **do**

$y \leftarrow f(\text{stier}[i,j])$;

if temp_cash_flow[i] > 0 **and** temp_cash_flow[i] $> y$ **then**

 | cash_flow[i] $\leftarrow \text{temp_cash_flow}[i] \cdot \frac{1}{R}$;

else

 | cash_flow[i] $\leftarrow \text{cash_flow}[i] \cdot \frac{1}{R}$;

 | **continue**

end

end

end

return mean(cash_flow)

Algorithm 3: LSM algoritmen for amerikansk put option

3 Numeriske resultater

3.1 Korrekthed af implementeringen

Et af problemerne ved at arbejde med numerisk approksimering er, at det er svært at vurdere præcisionen af approksimeringen. Heldigvis findes der indenfor Black-Scholes modellen en lukket løsning for prisen på europæiske optioner. Denne er givet i proposition 1 og 2 i afsnit 2.3. Givet denne pris er det muligt for os at tjekke vores implementering af binomial modellen og Monte Carlo simuleringen. Dette gøres ved at undersøge en europæisk put option med forskellige år til udløb (T) og volatilitet (σ). Optionen har fast en strike på $K = 100$, og det underliggende aktiv har en fast start pris på $S_0 = 100$. Binomial modellen er kørt med størrelsen $n = 10\,000$. Monte Carlo simuleringen er kørt med $n = 100\,000$, hvor af halvdelen er antitetiske stier.

Prisen p på en europæisk put option kan man estimere med Monte Carlo simulering ved at tage payoff på tidspunkt T , tilbagediskontere denne værdi og herefter tage gennemsnittet over alle stier.

$$p(0, S_0) \approx \frac{1}{n} \sum_{i=1}^n e^{-rT} (K - S_{\omega_i, T})^+$$

Tabel 1 viser resultaterne, hvor priserne under Black Scholes beskriver den sande pris for den europæiske put indenfor Black Scholes modellen.

Udover optionspriserne kan afvigelser og standard error også ses i tabellen. Afvigelsen beskriver afstanden mellem Monte Carlo og den sande pris, og binomial modellen og den sande pris. Standard error i tabellen er for aktiepriserne simuleret med Monte Carlo modellen, hvilket grundet antitetisk varians reduktion er regnet på følgende måde

$$\sigma_{\bar{S}_T} = \frac{\sigma_{S_T}}{\sqrt{n}} \quad (3.1)$$

Ud fra de simulerede stier kan standard afvigelsen af aktiepriserne estimeres.

$$\sigma_{S_T} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{1}{2} S_{\omega_i, T} + \frac{1}{2} \bar{S}_{\omega_i, T} - \hat{\mu} \right)^2} \quad (3.2)$$

$$\text{hvor } \hat{\mu} = \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^n S_{\omega_i, T} + \frac{1}{n} \sum_{i=1}^n \bar{S}_{\omega_i, T} \right) \quad (3.3)$$

Hvor $\bar{S}_{i, T}$ er den antitetiske sti hørende til $S_{i, T}$.

I disse udregninger gør det sig gældende, at $n = 50\,000$, da dette er antallet af henholdsvis de oprindelige stier og antitetiske stier.

T	σ	Monte Carlo	Binomial	Black Scholes	Afvigelse (MC)	Afvigelse (BM)	Standard error (MC)
0.1	0.1	0.97991	0.98118	0.98121	0.00130	0.00003	0.00031
0.1	0.3	3.47386	3.48030	3.48040	0.00654	0.00010	0.00283
0.1	0.5	5.96063	5.98767	5.98783	0.02720	0.00016	0.00796
0.5	0.1	1.54332	1.54716	1.54723	0.00391	0.00007	0.00163
0.5	0.3	6.94348	6.92701	6.92722	-0.01626	0.00021	0.01469
0.5	0.5	12.41049	12.39542	12.39577	-0.01472	0.00035	0.04092
1.0	0.1	1.64804	1.63567	1.63578	-0.01226	0.00011	0.00345
1.0	0.3	8.86081	8.89323	8.89353	0.03272	0.00030	0.02993
1.0	0.5	16.44042	16.38912	16.38961	-0.05081	0.00049	0.08452
3.0	0.1	1.23110	1.22011	1.22030	-0.01080	0.00019	0.01140
3.0	0.3	11.61860	11.66230	11.66280	0.04420	0.00050	0.10195
3.0	0.5	23.18294	23.12219	23.12297	-0.05997	0.00078	0.29870
5.0	0.1	0.78219	0.79776	0.79797	0.01578	0.00021	0.02112
5.0	0.3	12.10050	12.05049	12.05110	-0.04940	0.00061	0.19061
5.0	0.5	25.18603	25.10956	25.11049	-0.07554	0.00093	0.56199

Tabel 1: Priser på europæiske put optioner med fast $S_0 = K = 100$ bestemt med hhv. Black Scholes, binomial modellen og Monte Carlo simulering.

Ydermere beregnes mean absolute error (MAE), hvilket regnes som:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (3.4)$$

Her beskriver x_i de sande priser og y_i er modellernes priser. MAE for binomial modellen er 0.00034, og for Monte Carlo er det 0.02809. For binomial modellen gør denne meget lave MAE os meget sikre i, at den er implementeret korrekt. Til trods for, at det kun er for en europæisk option, viser det stadig, at vores implementering af henholdsvis op-og nedfaktorene er korrekte og ligeledes q -sandsynligheden, hvilke udgør de største usikkerheder, når det kommer til implementeringen.

MAE for Monte Carlo priserne er en del højere end for binomial priserne, men dette er, hvad der kan forventes. Grundet støj i simuleringsmodellen vil vi opleve, at Monte Carlo prisen generelt er mere upræcis. Med dette i mente, er vi dog rigtig godt tilfredse med MAE.

Vi bemærker, at afvigelsen og standard error stiger, jo længere til udløb og jo højere volatilitet. Dette er præcist, som vi forventede, når vi beregner priser med samme n for alle løbetider og volatiliteter. Generelt ligger begge dog lavt relativt til optionens pris.

Dette er med til at bekræfte, at vi har implementeret metoderne korrekt.

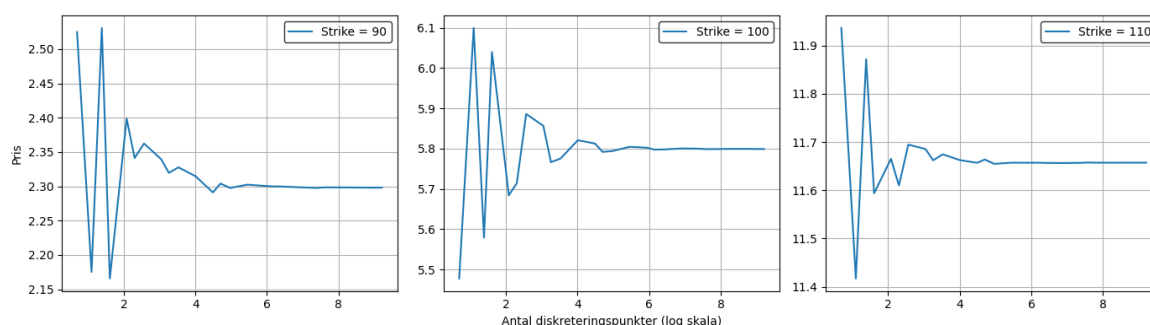
3.2 Udvalg af parametre

For at modellerne giver gode og nøjagtige resultater, er det vigtigt, at parametre såsom antal diskreteringspunkter i binomial modellen, antallet af simulerede stier i LSM og hvor mange indfrielsestidspunkter, der tillades, er nøje udvalgt. Da vi i praksis har diskrete antal indfrielsestidspunkter, bestemmer vi faktisk prisen på en såkaldt bermuda option. Dette er en option med et fast antal indfrielsestidspunkter i løbetiden. Lader man afstanden Δt mellem indfrielsestidspunkterne gå mod 0, da approksimerer bermuda optionen den amerikanske godt.

Følgende afsnit vil gennemgå "indstillingerne" i modellerne.

3.2.1 Binomial modellen

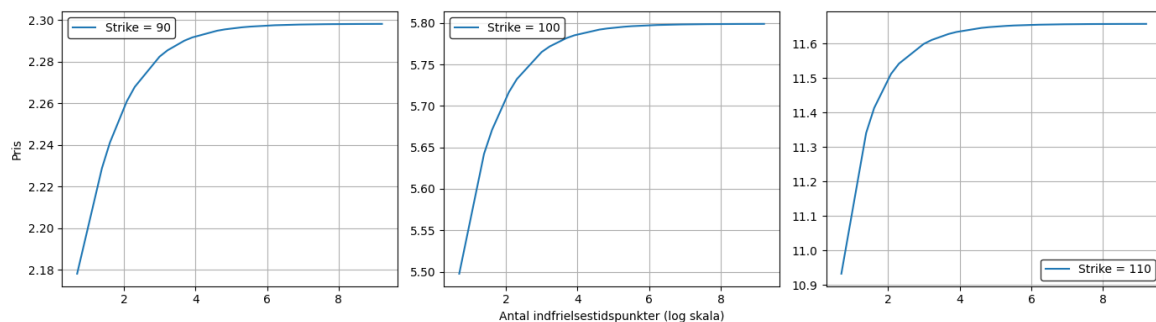
I binomial modellen er det vigtigt for prisen på optionen, at antallet af knuder i binomial træet er tilstrækkeligt højt. Gennem numeriske test, hvor prisen på en amerikansk put option er bestemt med binomial modellen ud fra et stigende antal perioder med indfrielse i dem alle, har vi etableret, at $e^6 \approx 400$ knuder er tilstrækkeligt til at stabilisere prisen. Dette resultat er vist i Figur 3 med tre optioner, hvor en er out of the money (otm), en at the money (atm) og en in the money (itm).



Figur 3: Pris på amerikansk put med startpris 100, løbetid 1 år, volatilitet 0.2 og risikofri rente 0.06 beregnet med binomial modellen vs antal perioder i binomial modellen

Antallet af indfrielsestidspunkter er ligeledes vigtigt for prisen på optionen.

For at bestemme et passende antal indfrielsestidspunkter opstiller vi et forsøg, hvor prisen på en amerikansk put bestemmes med binomial modellen med 10 000 diskreteringspunkter, og antallet af indfrielsestidspunkter gradvist øges mod 10 000.



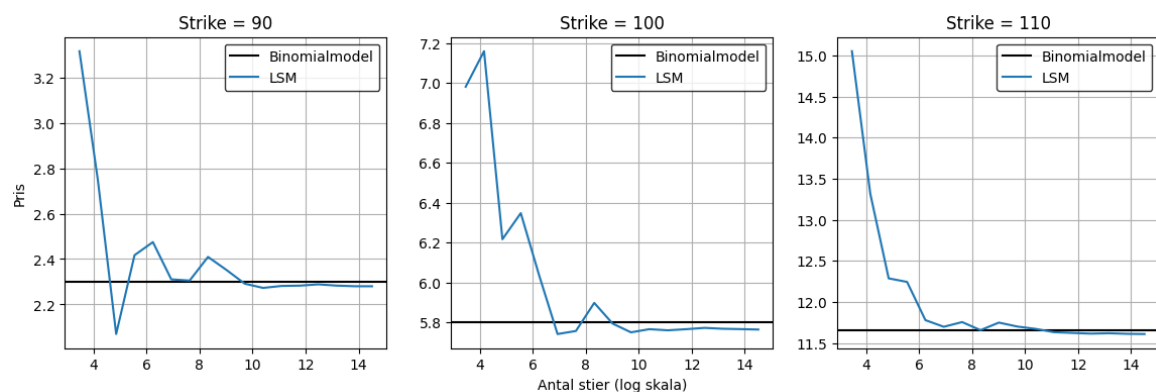
Figur 4: Pris på amerikansk put med startpris 100, løbetid 1 år, volatilitet 0.2 og risikofri rente 0.06 beregnet med binomial modellen vs antal årlige indfrielsestidspunkter i optionens løbetid

Vi ser, at prisen stabiliseres omkring $e^6 \approx 400$ indfrielsestidspunkter. Eftersom vi bruger binomial prisen som reference pris til at teste nøjagtigheden af vores andre modeller på amerikanske optioner, vælger vi at lade antallet af perioder $n = 10\,000$ med mulighed for indfrielse i dem alle.

3.2.2 LSM modellen

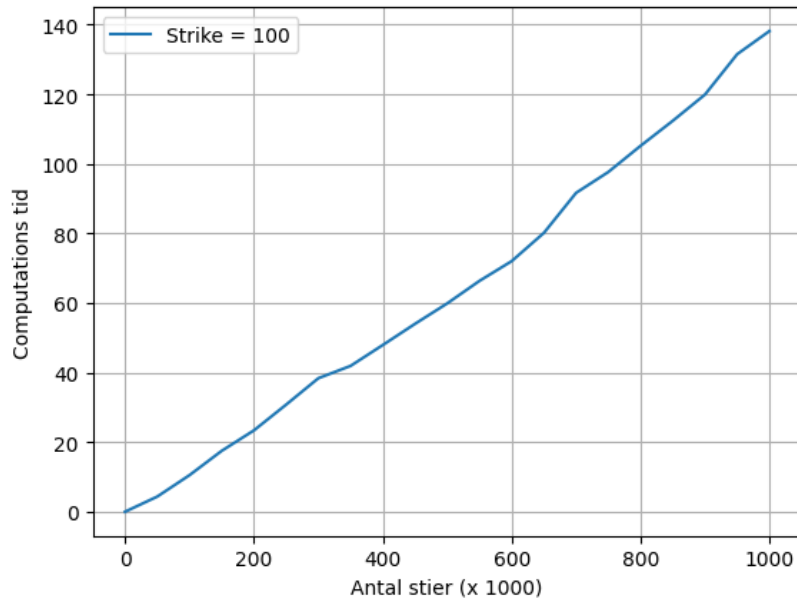
I LSM er det først og fremmest vigtigt at mindske Monte Carlo usikkerheden, der opstår under simuleringen. Antallet af stier, der er nødvendige for at få Monte Carlo usikkerheden til at blive negligerbar undersøger vi ved at prissætte amerikanske put optioner ud fra et stigende antal simulerede stier.

Dette er illustreret i Figur 5 for tre put optioner med strikes $K_1 = 90$, $K_2 = 100$ og $K_3 = 110$ på en underliggende aktie med startpris $S_0 = 100$. Volatiliteten er sat til $\sigma = 0.2$ og den risikofrie rente $r = 0.06$. Som reference benytter vi prisen på selvsamme option beregnet med binomial modellen.



Figur 5: Pris på amerikanske puts med startpris $S_0 = 100$, løbetid $T = 1$ år, volatilitet $\sigma = 0.2$ og risikofri rente $r = 0.06$ med strike på hhv. $K_1 = 90$, $K_2 = 100$ og $K_3 = 110$ og 50 årlige indfrielsestidspunkter.

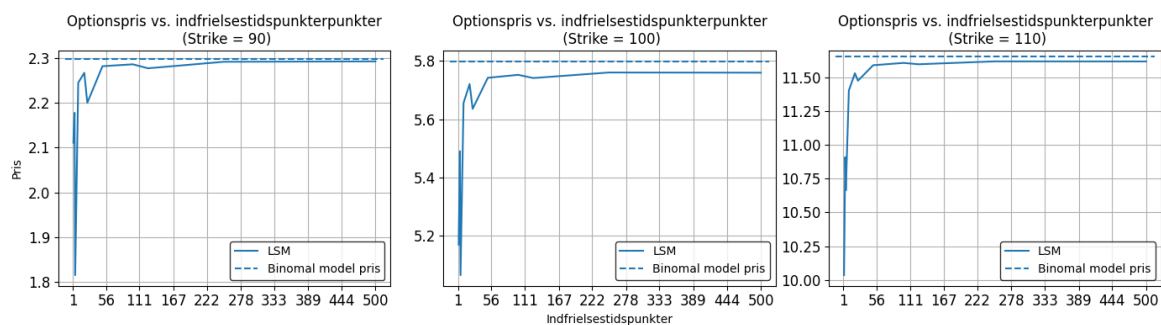
Vi observerer, at prisen på optionen stabiliseres, når antallet af simulerede stier øges. Fra $e^{12} \approx 162\,754$ stier og op ad er prisen stabil. Vi undersøger ligedes computationstiden for modellen baseret på antallet af stier. Dette er illustreret i Figur 6, som afbilder computationstid i sekunder overfor antallet af stier benyttet til prissætningen.



Figur 6: Computationstid i sekunder af LSM modellen baseret på et stigende antal stier

Vi bemærker, at computationstiden er lineært voksende i antal stier og ser altså ingen grund til at være hverken nærrige eller gavmilde med antallet af simulerede stier. Vi benytter derfor 200 000 simulerede stier, hvoraf halvdelen er antitetisk, til at prissætte amerikanske optioner med LSM algoritmen.

LSM algoritmen har diskrete antal indfrielsestidspunkter, altså bestemmer algoritmen, ligesom binomial modellen, prisen på en bermuda option. Figur 7 afbilder tre amerikanske put optioner på en underliggende aktie med start pris $S_0 = 100$ og løbetid på $T = 1$ år. Volatiliten er $\sigma = 0.2$ og den risikofrie rente $r = 0.06$.



Figur 7: Pris på amerikanske puts vs antal indfrielsestidspunkter

Vi ser, at prisen stabiliseres og approksimerer prisen på en amerikansk put godt allerede ved omkring 50 indfrielsestidspunkter årligt. Dette kan også ses på figur 4 for binomial modellen, hvor 50 indfrielsestidspunkter kun giver en lidt lavere pris end for 10 000. Til trods for at LSM prisen kan forbedres med antal indfrielsestidspunkter på 200 eller mere, vælger vi 50 årlige indfrielsestidspunkter. Dette er for at have et bedre sammenligningsgrundlag med andre modeller, som vi vil introducere senere. Disse modeller har en langt højere computationstid og er derfor ikke realistisk at køre med mange indfrielsestidspunkter.

3.3 LSM og binomial sammenligning

Vi har nu opstillet rammerne for både binomial modellen og LSM modellen. I dette afsnit vil vi holde de to modeller op i mod hinanden og vise, at LSM er en stabil prissætnings algoritme, der kan benyttes som et alternativ til binomial modellen. Dette vises ved at prissætte put optioner med varierende start priser og løbetider på underliggende aktier med varierende volatilitet. Renten og strike på optionerne holdes konstant på hhv. $r = 0.06$ og $K = 100$. Computationstiden findes ligeledes for begge modeller. Resultaterne er vist i tabel 2.

S_0	σ	T	Binomial	LSM	LSM afvigelse	Binomial tid	LSM tid
95	0.2	1	8.143015	8.090477	0.052538	1.507926	10.965328
95	0.2	3	10.100636	10.006492	0.094144	1.405017	33.951780
95	0.4	1	15.386793	15.357206	0.029587	1.568802	10.314332
95	0.4	3	21.602032	21.535008	0.067024	1.686351	35.833584
100	0.2	1	5.798864	5.769702	0.029162	1.472429	8.961140
100	0.2	3	8.047987	7.985215	0.062772	1.451413	27.505257
100	0.4	1	13.295550	13.273386	0.022164	1.655349	8.803188
100	0.4	3	19.853866	19.780108	0.073758	1.580408	28.634120
105	0.2	1	4.052937	4.032580	0.020357	1.405419	7.504762
105	0.2	3	6.414068	6.374677	0.039391	1.425279	25.210169
105	0.4	1	11.470386	11.411665	0.058720	1.584615	8.258441
105	0.4	3	18.272736	18.246679	0.026057	1.568317	30.004888

Tabel 2: Priser på amerikanske optioner bestemt med hhv. binomial modellen og LSM modellen. Computationstiden i sekunder for begge er angivet.

Vi beregner desuden standard error for 100 simulerede priser med LSM på optionen med $S_0 = K = 100$, $\sigma = 0.4$ og $T = 3$. Denne option er valgt, da vi forventer en højere grad af ustabilitet for denne. Standard error ligger på 0.000209. Vi ser ud fra den lave standard error på den LSM simulerede pris, at LSM algoritmen er en stabil prissætnings metode. Modellen har desuden en lav MAE = 0.04797 med binomial modellen som reference pris.

Den relativt lave afvigelse mellem LSM-og binomial prisen indikerer, at LSM algoritmen, ikke nok med at være stabil, også er en nøjagtig prissætningsmodel. Vi bemærker, at LSM prisen konsekvent er lavere end binomial prisen, hvilket stemmer overens med Proposition 3.

Computationstiden for LSM algoritmen er højere end for binomial modellen. Binomial modellen er deterministisk, hvilket vil sige, at der ikke er nogen usikkerhed i prisen givet med denne model. Dette er et tradeoff i LSM algoritmen, der trods den lave standard error altid vil være forbundet med Monte Carlo usikkerhed.

På en simpel option, inden for Black Scholes modellen, er der altså ikke nogen grund til at bruge LSM modellen frem for binomial modellen, da denne både prissætter langsommere og med usikkerhed. Vi vil nu undersøge, hvorvidt det er muligt at forbedre LSM algoritmen ved at introducere machine learning til algoritmen.

I det kommende afsnit vil vi lægge det teoretiske grundlag for at introducere machine learning til LSM.

4 Neurale netværk

Siden modellerne, som vi kigger på i dette projekt, bygger på at lave simuleringer og estimer, er det oplagt for os at undersøge hvorvidt machine learning kan forbedre disse. Hele princippet bag machine learning er nemlig at bruge store mængder data til at træne modeller, som bruges til at estimere en sammenhæng mellem input og output. En af disse modeller kaldes et neuralt netværk, som trænes med data for at kunne træffe beslutninger ud fra ukendt data.

4.1 Multi Layer Perceptron

Den model vi kigger på i dette projekt kaldes multilayer perceptron (MLP). Dette er et netværk, som består af et input layer, et output layer og et eller flere hidden layers. Hvert layer består af et antal neuroner. Indholdet af disse neuroner er parametrene, som bestemmer, hvordan inputtet skal afbilledes fra et layer til det næste. Antallet af neuroner beskrives som m^i for det i 'te layer. For input layer gælder det, at $m^1 = k$ for $x = (x_1, x_2, \dots, x_k)$. Altså er antallet af neuroner i det første layer lig med størrelsen på vores input vektor. Hele netværket består af sammensatte funktioner på følgende måde

$$F(x) = f_1 \circ f_2 \circ \dots \circ f_{n+1} \quad (4.1)$$

$$\text{hvor } f_i : \mathbb{R}^{m^{i-1}} \longrightarrow \mathbb{R}^{m^i} \text{ for } i = 1, 2, \dots, n+1 \quad (4.2)$$

Her er n antallet af hidden layers. For hver funktion gælder følgende

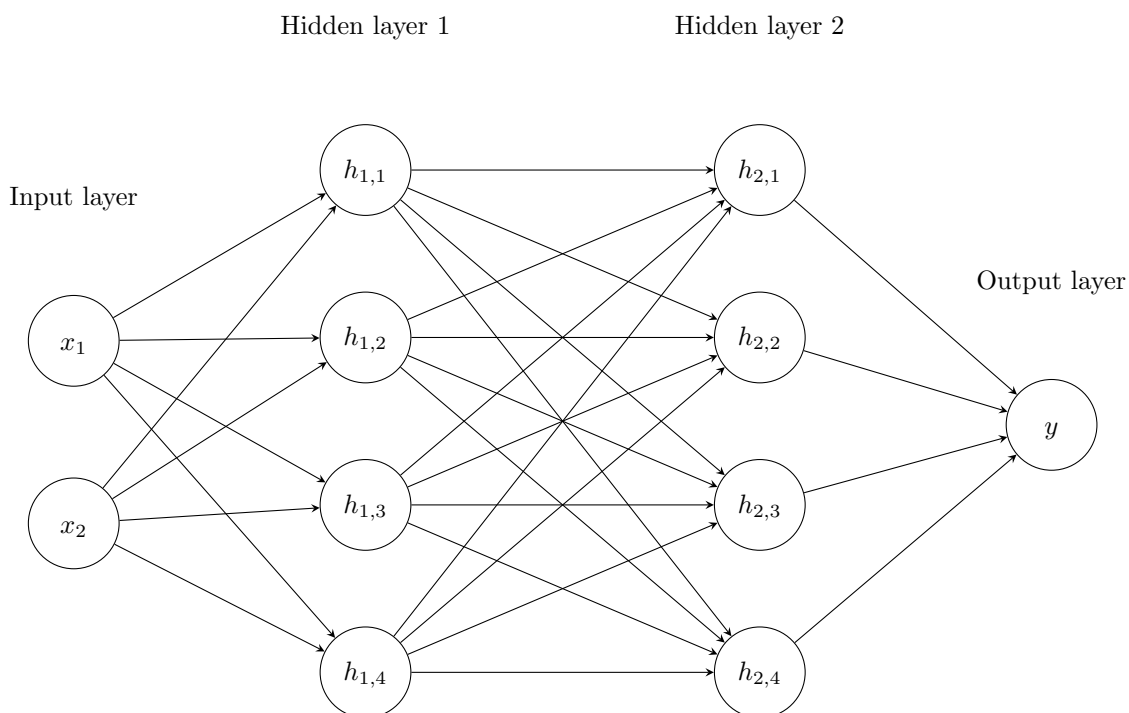
$$f_i(x) = g(W^T x + w_0) \quad (4.3)$$

$$\text{hvor } x \in \mathbb{R}^{m^{i-1}}, \quad W \in \mathbb{R}^{m^{i-1} \times m^i}, \quad w_0 \in \mathbb{R}^{m^i} \quad (4.4)$$

Her beskriver $W_j \in \mathbb{R}^{m^{i-1}}$ og $w_{0,j} \in \mathbb{R}$ de parametre, som et enkelt neuron j indeholder. W kaldes vægtene, og w_0 er bias. Udover dette så bruges en funktion g i afbildningen, hvilket er en aktiverings-funktion. Aktiverings-funktionen i et MLP netværk introducerer ikke-linearitet til modellen. Dette muliggøre det at fange komplekse sammenhænge, som en lineær model ikke ville kunne. Vi bruger aktiverings-funktionen RELU, som er defineret således:

$$g(x) = \max(0, x) \quad (4.5)$$

Dette er en af de mest almindelige aktiverings-funktioner. I figur 8 viser vi et neuralt netværk med to hidden layers.



Figur 8: MLP neural netværk med to hidden layers, hver med 4 neuroner, og ét output layer

Måden hvorpå netværket trænes er ved at minimere en cost funktion ved at ændre i vægtene og bias. Vi vælger mean squared error (MSE) som vores cost funktion. Dette er defineret som

$$C(w) = \frac{1}{n} \sum_{i=1}^n (x_i - y_i(w))^2 \quad (4.6)$$

hvor x_i er den sande værdi, y_i er den estimerede, og w er et fælles udtryk for alle parametrene i modellen. For at minimere denne cost funktion bruges en optimerings algoritme. I vores projekt bruger vi Adam algoritmen. Denne algoritme er at fortrække, hvis man skal træne netværket på store datasæt. Dette er, fordi algoritmen bruger mindre "batches" af datasættet til at estimere gradienten af cost funktionen. Dette er en mindre præcis, men meget mindre beregnings tung metode, end en algoritme, som estimerer gradienten for hele datasættet på en gang. Siden vi arbejder med datasæt med mulighed for flere hundrede tusinder af observationer, er dette mere eller mindre en nødvendighed.

Ydermere, så har Adam algoritmen endnu en fordel, nemlig at den har en fleksibel learning rate (η), som ændrer sig alt afhængig af cost funktionens overflade. Learning rate er et udtryk for, hvor meget parametrene ændres for hver iteration af algoritmen på et batch.

$$w_{ny} = w_{gammel} - \eta \nabla C(w_{gammel}) \quad (4.7)$$

I dette projekt vil vi bruge MLP netværket på to forskellige måder til at prissætte en amerikansk put option. De to modeller vil blive introduceret i de næste afsnit.

4.2 LSM-MLP

I LSM-MLP genbruger vi LSM algoritmens teknikker. Denne model kræver stadig et sæt Monte Carlo simulerede stier som vist i afsnit 2.6.1. Herefter bliver der som i LSM algoritmen trævlet baglæns gennem perioderne for at finde det optimale indfrielsestidspunkt τ^* ligesom i ligning (2.17). Dog bliver den forventede forsættelsesværdi ikke vurderet på samme måde som i ligning (2.16). I stedet trænes et nyt neuralt netværk til hver periode med aktiepriserne og payoff til næste periode for alle stier, som er itm. Altså bliver den forventede fortsættelsesværdi i denne model vurderet som

$$\mathbb{E}^Q[e^{-r(\tau_{k+1}^* - \tau_k)} C(\bar{\omega}, \tau_{k+1}, T) | S_{\bar{\omega}, \tau_k}] \approx F(S_{\bar{\omega}, \tau_k}) \quad (4.8)$$

hvor F beskriver det færdigtrænede neurale netværk til indfrielsestidspunkt τ_k . I denne model bruger vi følgende parametre for netværket F , inspireret af Peter Pommergård Linds arbejde[9]:

$$\begin{aligned} \text{batch size} &= 512 \\ \eta &= 0.001 \\ m^1 \times m^2 \times m^3 \times m^4 &= 1 \times 40 \times 40 \times 1 \end{aligned} \quad (4.9)$$

I Tabel 3 viser vi performance for LSM-MLP prissætnings modellen for forskellige optioner. Som nævnt tidligere vælger vi grundet computationstiden at prissætte optioner med 50 årlige indfrielsestidpunkter i LSM-MLP. Antallet af stier benyttet i LSM-MLP er 200 000 ligesom for LSM. Vi kigger på de samme optioner som i tabel 2 og prissætter disse med LSM-MLP.

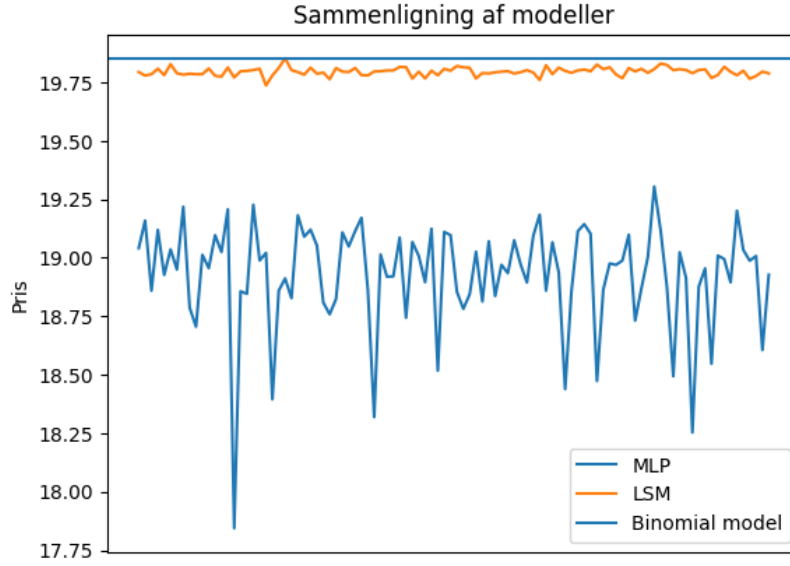
S_0	σ	T	Binomial model	LSM-MLP	Afvigelse	Tid (LSM-MLP)
95	0.2	1	8.143015	7.177227	0.965788	660.190833
95	0.2	3	10.100636	9.534320	0.566316	886.260402
95	0.4	1	15.386793	14.798099	0.588694	344.611736
95	0.4	3	21.602032	20.769052	0.832980	940.663086
100	0.2	1	5.798864	5.522225	0.276639	282.708424
100	0.2	3	8.047987	7.606689	0.441298	839.770941
100	0.4	1	13.295550	12.817399	0.478151	299.979592
100	0.4	3	19.853866	19.044095	0.809772	1230.377547
105	0.2	1	4.052937	3.755637	0.297300	248.182758
105	0.2	3	6.414068	6.054644	0.359424	1007.859331
105	0.4	1	11.470386	11.046978	0.423407	410.727800
105	0.4	3	18.272736	17.335377	0.937359	1348.741070

Tabel 3: Priser på amerikanske put optioner bestemt med hhv. binomial modellen og LSM-MLP modellen. Computationstiden for LSM-MLP er angivet i sekunder.

MAE for LSM-MLP på disse optioner er 0.5814273. Dette er over 10 gange større end for LSM algoritmen, hvilket sammen med computationstiden placerer denne metode langt under den almindelige LSM metode.

Derudover har vi regnet standard error for 100 estimerede priser for den samme option som for LSM. Altså optionen med $S_0 = 100$, $K = 100$, $T = 3$ og $\sigma = 0.4$. Standard error for disse estimer er 0.022658, hvilket er cirka 100 gange så højt som LSM.

Sammenlignet med LSM har LSM-MLP en meget høj standard error, hvilket indikerer, at den ikke er særlig stabil. Dette undersøger vi ved at visualisere de 100 priser, som standard error er regnet for. Dette gør vi for både LSM og LSM-MLP. Figur 9 viser resultatet af dette.



Figur 9: Stabilitetsgraf LSM & MLP

Vi ser ud fra tabel 3 og figur 9, at LSM-MLP har store afvigelser fra binomial modellen og generelt prissætter ustabil. Derfor vurderer vi, at denne model ikke har nogen nytte på en simpel amerikansk option i en dimension.

Vi introducerer nu en anden implementering af et MLP neuralt netværk til options prissætning.

4.3 Binomial-MLP

Binomial-MLP (B-MLP) er en ren machine learning model. Modellen er en MLP model, hvor modellens parametre igen er inspireret af Peter Pommergård Lind. [9]. Input layer og output layer er givet ved:

$$x = \begin{pmatrix} \frac{S_0}{K} \\ \sigma \\ T \end{pmatrix}, \quad y = (p) \quad (4.10)$$

Derudover er modellens parametre givet som

$$\begin{aligned} \text{batch size} &= 64 \\ \eta &= 0.0001 \\ m^1 \times m^2 \times m^3 \times m^4 \times m^5 &= 3 \times 120 \times 120 \times 120 \times 1 \end{aligned} \quad (4.11)$$

Modellen trænes på et datasæt, hvor prisen til input variableerne er bestemt med binomial modellen med $n = 5000$.

Fordelen ved sådan en model er, at den er meget hurtig, når først den er trænet. Derudover virker den ligesom LSM også på andre modeller end Black Scholes, det kræver blot, at man har tilstrækkelig med god data til at træne den. I teorien kan man træne modellen med data fra den virkelige verden.

4.3.1 Træningsdata

Selve datasættet består af uafhængige observationer med put optionspriser, optionens strike og det underliggende aktivs parametre. For at mindske antallet af variabler i modellen, definerer vi $M = \frac{S_t}{K}$. Den nye payoff funktion for put optionen ser således ud:

$$\Phi(S_t) = K \left(1 - \frac{S_t}{K}\right)^+ = K (1 - M)^+ \quad (4.12)$$

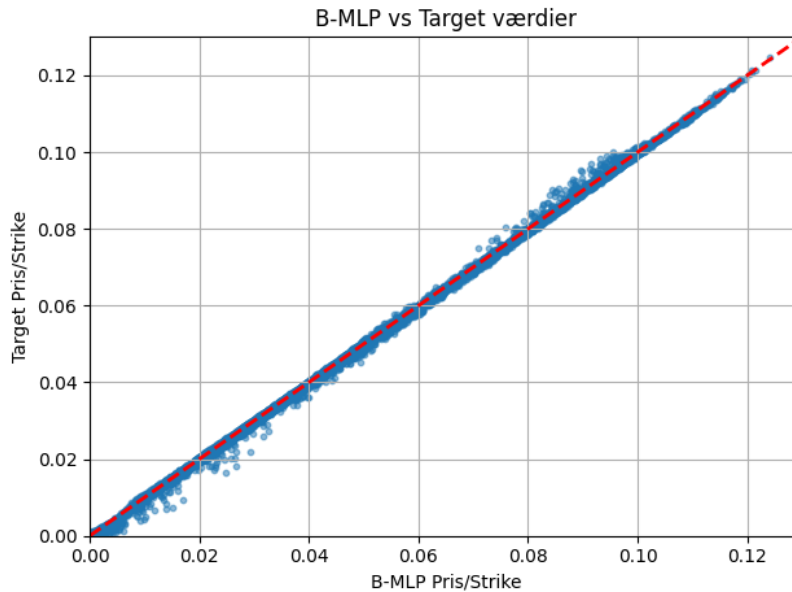
Hermed mindskes antallet af variabler fra 4 til 3 i input layeret i netværket. Renten holdes fortsat konstant på $r = 0.06$.

Optionens og det underliggende aktivs parametre bliver trukket fra en uniform fordeling i følgende intervaller:

	M	σ	T
Interval	0.9 – 1.1	0.15 – 0.25	0.01 – 1.0

Tabel 4: Intervaller til parametre i træningsdata til B-MLP

Vi laver et trænings datasæt bestående af 200 000 observationer. Modellen trænes på dette datasæt, hvorefter vi tester nøjagtigheden af modellen på 10 000 out of sample observationer. Figur 10 viser target værdierne vs de af B-MLP beregnede værdier.



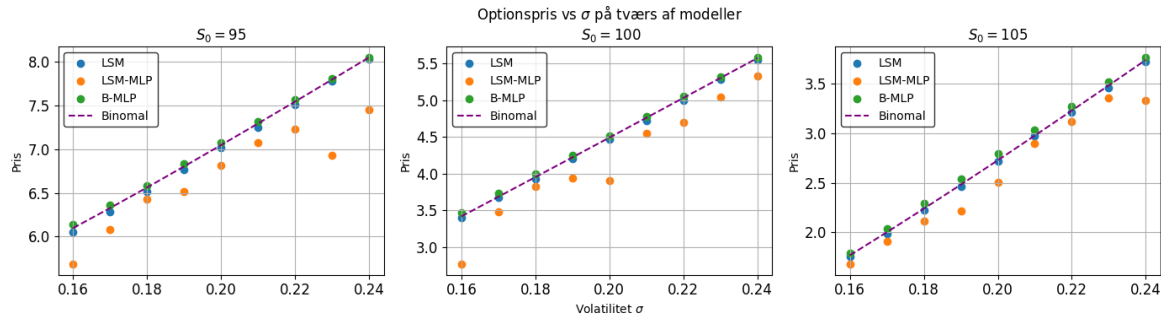
Figur 10: Target- vs B-MLP værdier

Vi ser, at B-MLP og target værdierne ligger pænt på en linje, hvilket viser at B-MLP prissætter nøjagtigt og stabilt. Vi bemærker dog, at den generelt prissætter højere end target værdierne, hvilket bliver tydeligere, når optionsprisen går mod 0. Modellen har en lav MAE på 0.0005093 for $\frac{\text{pris}}{\text{strike}}$, der sammen med en gennemsnitlig beregningstid pr pris på 0.23 millisekunder gør B-MLP til en nøjagtig og ekstrem hurtig prissætningsmodel, når først den er trænet.

4.4 Resultater for amerikansk put option

Nu har vi introduceret alle modellerne og præsenteret deres respektive prissætningsresultater. Inden vi undersøger deres performance på mere komplicerede optioner, vil vi samle op på resultaterne for simple optioner på et underliggende aktiv.

Figur 11 viser alle modellernes priser op i mod binomial modellen som reference for en itm, atm og otm option med stigende volatilitet i det underliggende aktiv.



Figur 11: Model priser vs stigende volatilitet med fast $K = 100$ og $T = 0.5$

Vi ser, at LSM-MLP underperformer LSM og B-MLP på samtlige priser for alle tre optioner. Dette, sammen med den høje computationstid, gør LSM-MLP til en overflødig model på simple optioner. Både LSM og B-MLP er stabile henover alle priser, hvor B-MLP har tendens til at prissætte for højt og LSM for lavt. Tabel 5 viser MAE for de tre modeller på alle optionerne samt en samlet MAE.

S_0	LSM	LSM-MLP	B-MLP
95	0.036043	0.368403	0.025069
100	0.023778	0.322039	0.029955
105	0.014355	0.168258	0.044996
∪	0.024725	0.286233	0.033340

Tabel 5: MAE for modellerne målt op i mod binomial modellen

LSM har den laveste totale MAE, mens B-MLP har den laveste på in the money optioner. Overordnet set er begge modeller nøjagtige og stabile på simple optioner. LSM og B-MLP viser nogle rigtig gode resultater for de simple optioner her. Derimod er LSM-MLP meget ustabil, har høj computationstid og har store afvigelser. LSM-MLP vil altså ikke være at foretrække for optioner som disse.

Vi vil nu undersøge, hvorvidt dette billede ændrer sig, når vi kigger på mere komplicerede optioner, der afhænger af flere underliggende aktiver.

5 Optioner på to aktiver

I det følgende betragter vi en put minimum option på to underliggende risikofyldte aktiver med følgende payoff funktion:

$$\Phi(S_{1,t}, S_{2,t}) = \left(K - \min(S_{1,t}, S_{2,t}) \right)^+ \quad (5.1)$$

Vi vil i de kommende afsnit undersøge, hvordan henholdsvis binomial modellen, LSM og MLP modellerne performer på denne slags option. Da formålet med undersøgelsen er at sammenligne de forskellige modellers performance, vælger vi at antage, at de to aktiver er uafhængige. Dette er selvfølgelig en urealistisk antagelse, når det kommer til den virkelige verden, men det passer godt til vores formål og gør implementeringen af modellerne simple.

5.1 Europæiske minimum optioner - Black Scholes model

Vi starter med at betragte den europæiske udgave af denne option, som indenfor Black Scholes modellen har en lukket løsning ligesom sin en dimensionelle modpart. Løsningen blev fremvist af Rene Stulz i 1982.[10]

Proposition 4 (Black Scholes formel for minimum option:). *Givet to uafhængige risikofyldte aktiver, er prisen på en europæisk call minimum option med strike K , tid til udløb T , og payoff funktion*

$$\Phi(S_{1,T}, S_{2,T}) = \left(\min(S_{1,T}, S_{2,T}) - K \right)^+$$

givet ved formelen $c(t, S_{1,t}, S_{2,t})$, hvor

$$\begin{aligned} c(t, S_{1,t}, S_{2,t}) = & S_{1,t} N_2[(d_3(t, S_{2,t}, S_{1,t}), d_1(t, S_{1,t})), \begin{pmatrix} 1 & -\rho_1 \\ -\rho_1 & 1 \end{pmatrix}] \\ & + S_{2,t} N_2[(d_3(t, S_{1,t}, S_{2,t}), d_1(t, S_{2,t})), \begin{pmatrix} 1 & -\rho_2 \\ -\rho_2 & 1 \end{pmatrix}] \\ & - e^{-r(T-t)} K N_2[(d_2(t, S_{1,t}), d_2(t, S_{2,t})), \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}] \end{aligned}$$

Her angiver N_2 fordelingsfunktionen for normalfordelingen i to dimensioner med middelværdi $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

og

$$\begin{aligned}
d_1(t, S_t) &= \frac{1}{\sigma\sqrt{T-t}} \left(\ln \left(\frac{S_t}{K} \right) + \left(r + \frac{1}{2}\sigma^2 \right) (T-t) \right) \\
d_2(t, S_t) &= d_1(t, S_t) - \sigma\sqrt{T-t} \\
\sigma_3^2 &= \sigma_1^2 + \sigma_2^2 \\
\rho_1 &= \frac{\sigma_1}{\sigma_3} \\
\rho_2 &= \frac{\sigma_2}{\sigma_3} \\
d_3(t, S_{1,t}, S_{2,t}) &= \frac{1}{\sigma_3\sqrt{T-t}} \left(\ln \left(\frac{S_{1,t}}{S_{2,t}} \right) - \frac{1}{2}\sigma_3^2(T-t) \right)
\end{aligned}$$

Det gælder ligeledes i to dimensioner, at der findes en put-call paritet for de europæiske udgaver af optionerne.

Proposition 5 (Put-call paritet i to dimensioner:). *Givet en europæisk call og put option med samme strike K og tid til udløb T . Med prisfunktioner givet ved hhv. $c(t, S_{1,t}, S_{2,t}, K)$ og $p(t, S_{1,t}, S_{2,t}, K)$, gælder følgende relation:*

$$p(t, S_{1,t}, S_{2,t}, K) = Ke^{-r(T-t)} + c(t, S_{1,t}, S_{2,t}, K) - c(t, S_{1,t}, S_{2,t}, 0)$$

Udtrykket $c(t, S_{1,t}, S_{2,t}, 0)$ beskriver call optionen med en strike på 0. Værdien af denne option kan findes som grænseværdien for $K \rightarrow 0$, hvilket giver følgende

$$\begin{aligned}
c(t, S_{1,t}, S_{2,t}, K) &\rightarrow S_{1,t}N_1[d_3(t, S_{2,t}, S_{1,t}, 1)] + S_{2,t}N_1[d_3(t, S_{1,t}, S_{2,t}, 1)] \\
&\text{for } K \rightarrow 0
\end{aligned}$$

Implementeringen af denne lukkede løsning for den europæiske put minimum option gør det muligt for os at verificere vores implementering af binomial modellen for to aktiver.

5.2 Binomial modellen i to dimensioner

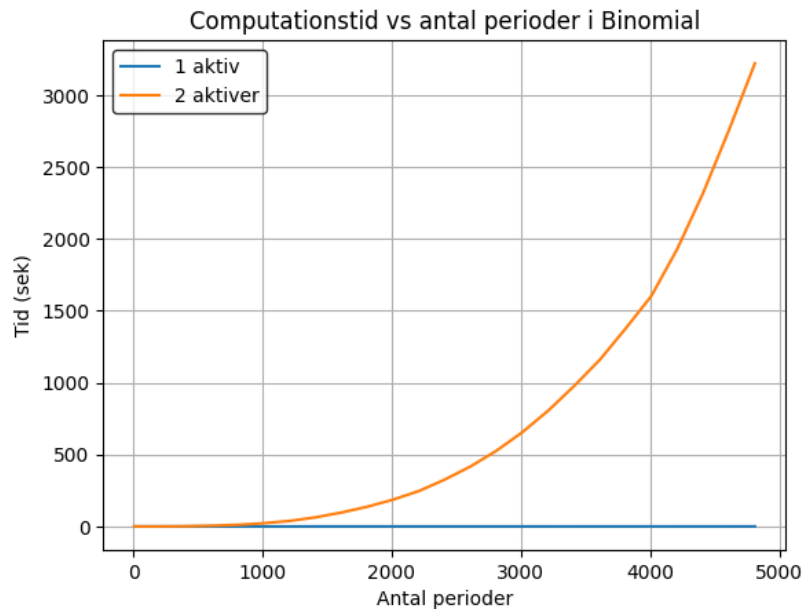
I flere dimensioner skal man bruge den i 1989 af Phelim P. Boyle, Jeremy Evnine og Stephen Gibbs udviklede ovenbygning på standard binomial modellen.[3]

Her har hver node i binomial modellen 2^d grene, hvor d er antallet af aktiver i modellen. I to dimensioner er der altså fire udviklingsmuligheder for hver node. Ligesom i Black Scholes modellen, antager vi for simpelhedens skyld, at aktierne er uafhængige. På samme måde som i en dimension stiger aktierne med sandsynlighed q_1 for aktie 1 og q_2 for aktie 2. Sandsynlighederne for de fire udviklingsmuligheder er givet ved:

Sandsynlighed	Udvikling
$q_{uu} = q_1 \cdot q_2$	Op, Op
$q_{ud} = q_1 \cdot (1 - q_2)$	Op, Ned
$q_{du} = (1 - q_1) \cdot q_2$	Ned, Op
$q_{dd} = (1 - q_1) \cdot (1 - q_2)$	Ned, Ned

Tabel 6: Udviklingssandsynligheder for binomial model i to dimensioner

For at beskrive aktieudviklingerne og dermed optionsprisen med binomial modellen på flere aktiver skal man altså beregne 2^d grene for hver knude i træet. Hermed vokser computationsbyrden eksponentielt. Dette betyder, at der ved udløb er $(1 + n)^d$ noder, hvor n er antallet af perioder. Figur 12 illustrerer den eksponentielle vækst i computationstiden, når antallet af dimensioner hæves fra en til to i binomial modellen.



Figur 12: Computationstid i sekunder vs antal perioder i binomial modellen for både en og to dimensioner.

I tre dimensioner vil udviklingen i computationstiden være endnu voldsommere, og i fire dimensioner er det mere eller mindre umuligt at implementere. Dette gør binomial modellen til en mindre oplagt model for høj dimensionelle optioner.

Implementeringen af selve algoritmen er en del mere kompliceret end i en dimension, men følger mange af de samme principper. Lad $V_{k_1, k_2, t}$ beskrive værdien af en knude til tidspunkt t med k_1

opryk for aktiv 1 og k_2 opryk for aktiv 2. Derved er knudens værdi bestemt ved

$$\text{Europæisk option: } V_{k_1, k_2, t} = F(t, S_{1,t}, S_{2,t}) = \frac{\mathbb{E}^q[\Phi(S_{1,T}, S_{2,T})]}{R^{T-t}} \quad (5.2)$$

$$\text{Amerikansk option: } V_{k_1, k_2, t} = \max(\Phi(S_{1,t}, S_{2,t}), \frac{\mathbb{E}^q[V_{t+\Delta t} \mid V_{k_1, k_2, t}]}{R}) \quad (5.3)$$

hvor

$$\mathbb{E}^q[V_{t+\Delta t} \mid V_{k_1, k_2, t}] = q_{uu}V_{k_1+1, k_2+1, t+\Delta t} + q_{ud}V_{k_1+1, k_2, t+\Delta t} + q_{du}V_{k_1, k_2+1, t+\Delta t} + q_{dd}V_{k_1, k_2, t+\Delta t} \quad (5.4)$$

Implementeringen i praksis kan ses i algoritme 4.

$S_{1,0}$: Start pris for aktiv 1, $S_{2,0}$: Start pris for aktiv 2, r : Risikofri rente, σ_2 : Volatilitet for aktiv 1, σ_2 : Volatilitet for aktiv 2, n : Antal stier, T : År til udløb, K : Strike, EU/AM : Optionens type

$$P \leftarrow n+1 \left\{ \underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}}_{n+1} \right\};$$

```

return  $P[0,0]$ 

```

31

5.3 Korrekthed af implementeringen

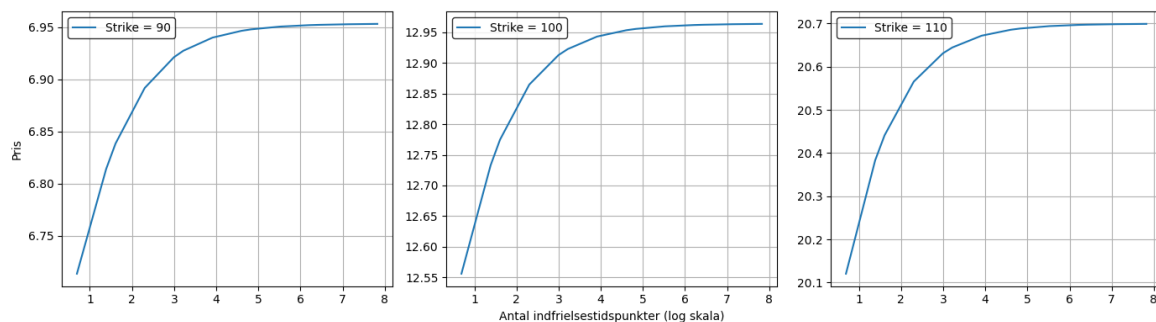
Tabel 7 viser priser for forskellige europæiske put minimum optioner bestemt med den to dimensionelle binomial model samt Black Scholes prisen som reference. Binomial prisen er regnet med $n = 2500$ perioder. Start pris og strike er konstante med $S_0 = K = 100$, og renten fastholdes fortsat på $r = 0.06$.

T	σ_1	σ_2	Binomial model	Black Scholes	Afvigelse
0.1	0.1	0.2	2.849055	2.849267	0.000212
0.1	0.3	0.4	7.012757	7.013154	0.000397
0.1	0.5	0.6	11.140248	11.140833	0.000586
0.5	0.1	0.2	5.211867	5.212406	0.000539
0.5	0.3	0.4	14.098873	14.099755	0.000882
0.5	0.5	0.6	22.725433	22.726651	0.001218
1.0	0.1	0.2	6.260480	6.261318	0.000838
1.0	0.3	0.4	18.247940	18.249184	0.001244
1.0	0.5	0.6	29.693453	29.695082	0.001629
3.0	0.1	0.2	6.889682	6.891377	0.001695
3.0	0.3	0.4	24.468268	24.470404	0.002135
3.0	0.5	0.6	40.553234	40.555653	0.002418
5.0	0.1	0.2	6.282255	6.284476	0.002220
5.0	0.3	0.4	25.714306	25.717008	0.002702
5.0	0.5	0.6	42.973684	42.976434	0.002750

Tabel 7: Europæiske put minimum priser bestemt med hhv. binomial modellen og den lukkede løsning fra Black Scholes som reference

Det kan ses, at priserne beregnet med binomial modellen ligger meget tæt på dem fra den lukkede løsning. Faktisk er der en MAE på kun 0.001431. Ydermere ses det, at afvigelserne stiger, som volatilitet og løbetid stiger, hvilket er, hvad der forventes. Altså er vi meget sikre på implementeringen af binomial modellen for to aktiver.

Computationstiden i to dimensioner, især for LSM-MLP, er relativ høj for mange indfrielsestidpunkter. Vi vælger derfor ligesom i en dimension at approksimere den amerikanske option med en bermuda option med 50 årlige indfrielsestidspunkter. Figur 13 viser, at dette fortsat er en acceptabel beslutning i to dimensioner:

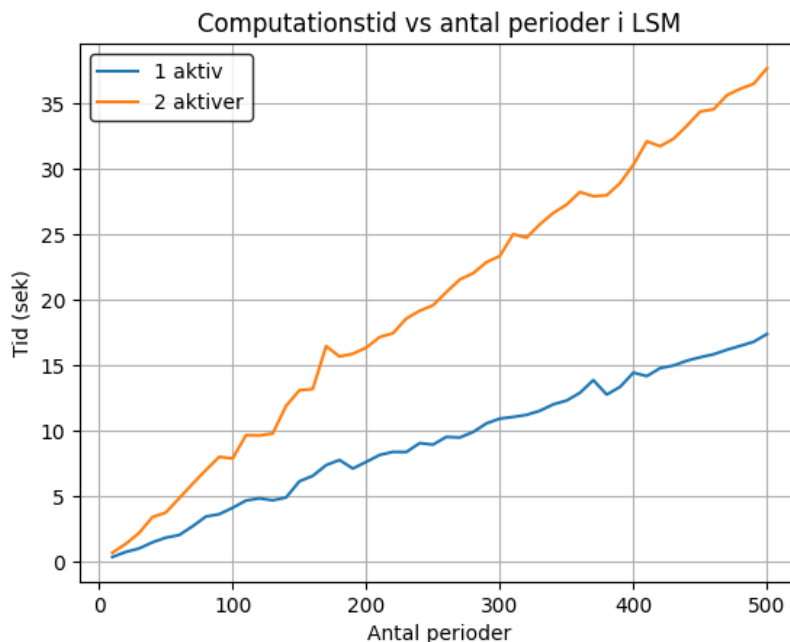


Figur 13: Pris på amerikanske put minimum option på to underliggende aktiver med startpriser $S_{1,0} = S_{2,0} = 100$, løbetid $T = 1$ år, volatilitet $\sigma_1 = 0.2$ og $\sigma_2 = 0.3$ samt risikofri rente $r = 0.06$ vs antal indfrielsestidspunkter i optionens løbetid.

Vi ser her, at 50 årlige indfrielsestidspunkter fortsat approksimerer den sande pris tilstrækkelig godt nok til, at vi vælger dette.

5.4 LSM i to dimensioner

LSM modellen lider ikke på samme måde som binomial modellen under dimensions udvidelse. Computationsbyrden stiger her kun lineært, da implementeringen blot kræver $d \cdot n$ stier, hvor d beskriver antal aktiver. Dette er illustreret i Figur 14.



Figur 14: Computationstid i sekunder vs antal diskreterings- og indfrielsestidspunkter i LSM algoritmen for en og to dimensioner.

I praksis vil LSM modellen for denne type put minimum option være implementeret nærmest identisk med put optionen for et aktiv. Altså minder implementeringen om den i algoritme 3. Forskellen er, at der nu bliver simuleret n par af stier for to uafhængige aktiekurser. Derudover bliver cash_flow og temp_cash_flow beregnet udfra den nye payoff funktion i ligning (5.1). Ydermere er den estimerede fortsættelsesværdi selvfølgelig også anderledes end i ligning (2.16). Til dette opstiller vi tre forskellige designmatricer til lineær regression, hvor følgende approximation bestemmer den forventede fortsættelsesværdi

$$\mathbb{E}^Q[e^{-r(\tau_{k+1}^* - \tau_k)} C(\bar{\omega}, \tau_{k+1}, T) | S_{1,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k}] \approx \beta_0 + \beta \mathbf{X} \quad (5.5)$$

Her beskriver $\bar{\omega}$ de par af stier, for hvilke optionen er in the money.

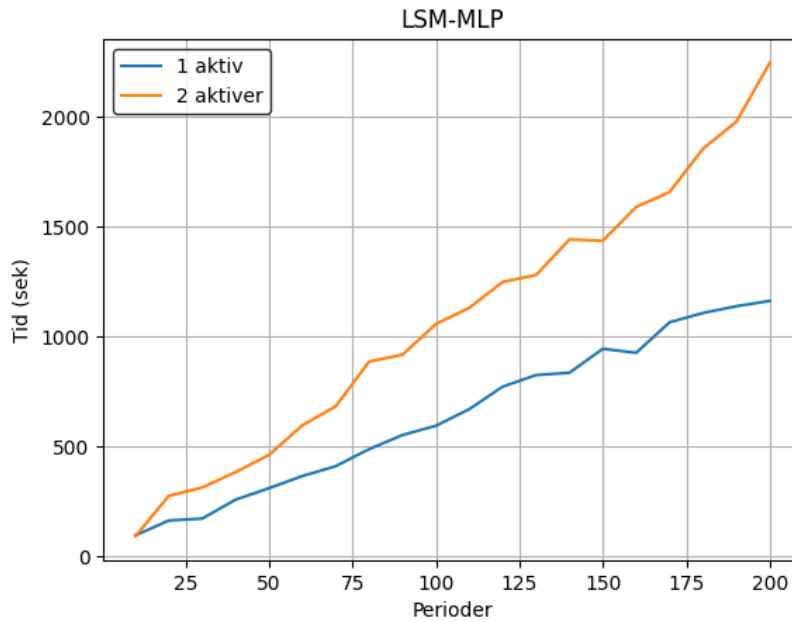
De tre forskellige designmatricer kan ses herunder.

$$\begin{aligned} \mathbf{X}_1 &= (S_{1,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k}^2, S_{2,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k}^2) \\ \mathbf{X}_2 &= (S_{1,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k}^2, S_{2,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k}^2, S_{1,\bar{\omega},\tau_k} \cdot S_{2,\bar{\omega},\tau_k}) \\ \mathbf{X}_3 &= (S_{1,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k}^2, S_{2,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k}^2, S_{1,\bar{\omega},\tau_k} \cdot S_{2,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k}^2 \cdot S_{2,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k} \cdot S_{2,\bar{\omega},\tau_k}^2) \end{aligned} \quad (5.6)$$

Vil vil undersøge hvilke af disse tre designs, som performer bedst. Inden da introducerer vi LSM-MLP i to dimensioner.

5.5 LSM-MLP i to dimensioner

LSM-MLP fungerer ligesom den almindelige LSM algoritme rigtig fornuftigt i flere dimensioner, da computationsbyrden ligeledes her vokser lineært med antal aktiver. Dette er illustreret i Figur 15.



Figur 15: Computationstid i sekunder vs antal diskreterings- og indfrielsestidspunkter i LSM-MLP algoritmen får en og to dimensioner.

Den forventede fortsættelsesværdi til hver periode vurderes på samme måde af et neuralt netværk som med et aktiv

$$\mathbb{E}^Q[e^{-r(\tau_{k+1}^* - \tau_k)} C(\bar{\omega}, \tau_{k+1}, T) | S_{1,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k}] \approx F(S_{1,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k}) \quad (5.7)$$

Vi bruger her samme opsætning for netværket, som vi gjorde for et aktiv. Dette kan ses i ligning (4.9). Dog undersøger vi her også tre forskellige designs af de hidden layers, hvilket kan ses herunder

$$\begin{aligned} \mathbf{m}_1 &= m^1 \times m^2 \times m^3 = 1 \times 100 \times 1 \\ \mathbf{m}_2 &= m^1 \times m^2 \times m^3 \times m^4 = 1 \times 40 \times 40 \times 1 \\ \mathbf{m}_3 &= m^1 \times m^2 \times m^3 \times m^4 = 1 \times 100 \times 100 \times 1 \end{aligned} \quad (5.8)$$

5.6 Sammenligning af forskellige designs

Tabel 8 viser sammenligningen af de forskellige designs, som vi har opstillet for de to dimensionelle modeller. Vi prissætter en amerikansk put minimum option med fast strike $K = 100$, $\sigma_2 = 0.4$, $T = 1$ og 50 årlige indfrielsestidspunkter. Derudover holder, vi start prisen S_0 ens for de to aktiver, altså $S_{1,0} = S_{2,0}$. LSM og LSM-MLP priserne er bestemt med $n = 200\,000$ stier af hvert aktiv, hvoraf halvdelen er antitetiske og binomial modellen er med $n = 2500$ perioder.

S_0	σ_1	LSM \mathbf{X}_1	LSM \mathbf{X}_2	LSM \mathbf{X}_3	LSM-MLP \mathbf{m}_1	LSM-MLP \mathbf{m}_2	LSM-MLP \mathbf{m}_3	Binomial
95	0.2	18.666950	18.637529	19.036914	18.918549	18.458514	18.718904	19.290886
95	0.4	21.087428	20.877910	21.468954	20.752823	20.001949	19.728960	21.790109
100	0.2	15.800990	15.721647	16.123707	15.864095	15.459339	15.455130	16.365174
100	0.4	18.235169	17.923136	18.536631	18.446932	17.967173	17.331220	18.929590
105	0.2	13.222982	13.096982	13.502859	13.272351	12.897235	12.687870	13.806390
105	0.4	15.697813	15.391329	16.051169	15.902391	15.338774	15.462772	16.348214
MAE		0.636505	0.813638	0.301688	0.562203	1.067897	1.190918	

Tabel 8: Forskellige designmatricer og hidden layers vs binomial

Vi ser nogle rimelige klare trends blandt de forskellige designs i de to modeller. \mathbf{X}_3 og \mathbf{m}_1 har de klart laveste MAE indenfor deres modeller. Dette er MAE med binomial modellens priser som reference. Derudover, har de over hele linjen højere priser end deres to modparter. Dette er et godt tegn, da det vides fra proposition 3, at LSM modellen er en nedre grænse for den sande pris. Vi vælger derfor \mathbf{X}_3 og \mathbf{m}_1 til videre benyttelse.

5.7 B-MLP i to dimensioner

Udvidelsen af vores B-MLP model til to aktiver er rimelig klar. Ligesom for et aktiv regnes et stort datasæt af forskellige priser gennem binomial modellen med $n = 500$. Input og output layers bliver som nedenstående

$$x = \begin{pmatrix} \frac{S_{1,0}}{K} \\ \frac{S_{2,0}}{K} \\ \sigma_1 \\ \sigma_2 \\ T \end{pmatrix}, \quad y = \begin{pmatrix} p \end{pmatrix} \quad (5.9)$$

De øvrige parametre for denne model er valgt ligesom for vores model for en dimension, som er vist i ligning (4.11). Der tages også igen udgangspunkt i $M = \frac{S_t}{K}$ ligesom i ligning (4.12), men denne gang er payoff givet ved

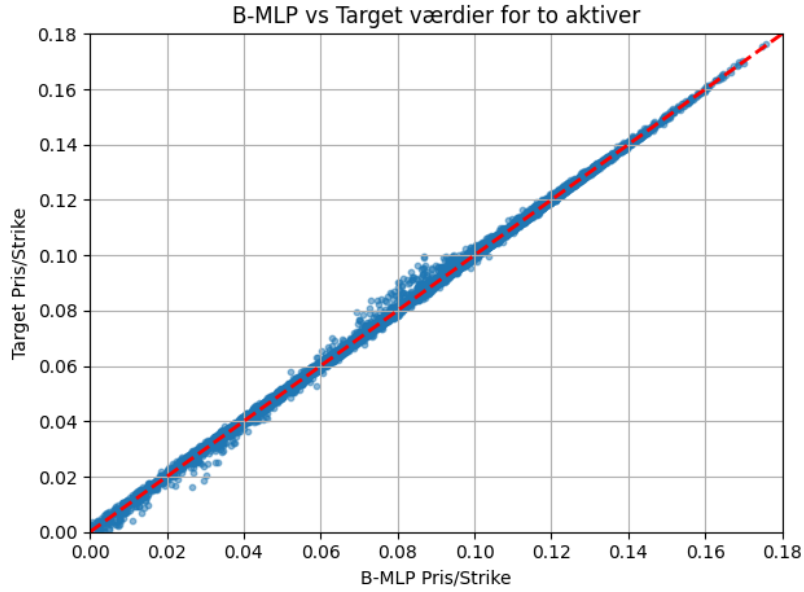
$$\Phi(S_{1,t}, S_{2,t}) = K \left(1 - \min \left(\frac{S_{1,t}}{K}, \frac{S_{2,t}}{K} \right) \right)^+ = K (1 - \min(M_1, M_2))^+ \quad (5.10)$$

De forskellige parametre for optionerne trækkes på ligevis fra en uniform fordeling indenfor følgende intervaller:

	M_1, M_2	σ_1, σ_2	T
Interval	0.9 – 1.1	0.15 – 0.25	0.01 – 1.0

Tabel 9: Intervaller til parametre i træningsdata til B-MLP for to aktiver

Vores datasæt består ligesom i en dimension af 200 000 observationer. Modellen trænes på dette datasæt, hvorefter vi tester nøjagtigheden af modellen på 10 000 out of sample observationer. Figur 16 viser target værdierne vs de af B-MLP beregnede værdier for to underliggende aktiver.



Figur 16: B-MLP vs target for to underliggende aktiver

Target værdierne og B-MLP værdierne ligger fint på en linje. B-MLP har ligesom for et underliggende aktiv en tendens til at prissætte for højt i de lave priser. Omkring midten prissætter B-MLP for lavt sammenlignet med binomial. Generelt ligger priserne dog rigtig flot på en linje. Dette ses også på den lave MAE, som for $\frac{\text{pris}}{\text{strike}}$ er på 0.0007632.

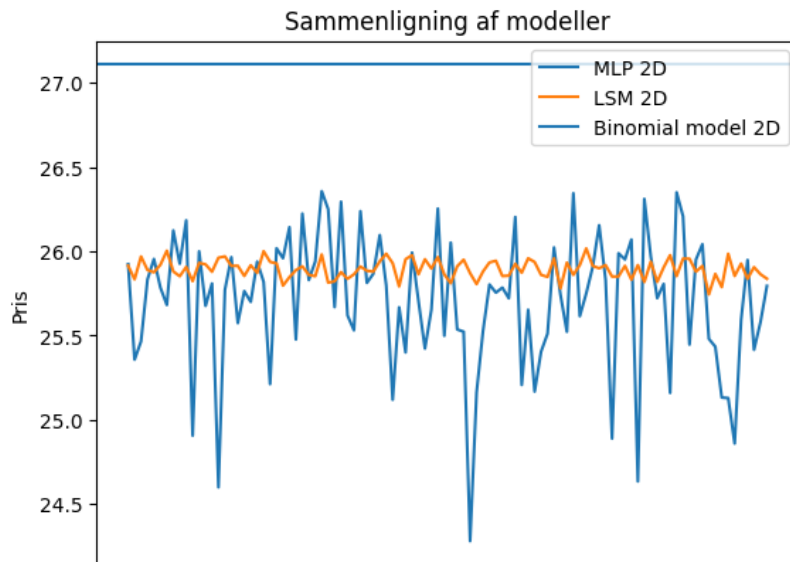
5.8 Resultater for amerikansk put minimum option

Vi samler nu op på alle prissætnings resultaterne på vores put minimum type option. Lad os først betragte LSM og LSM-MLP. Tabel 10 viser priser på put minimum optionen med fast strike $K = 100$, risikofri rente $r = 0.06$ og $\sigma_2 = 0.4$. Afvigelsen mellem de to prissætnings metoder og binomial modellen er angivet.

S_0	σ_1	T	Binomial	LSM	LSM afvigelse	LSM-MLP	MLP afvigelse
95	0.2	1	19.290886	19.006727	0.284159	18.233276	1.057610
95	0.2	3	25.758013	24.952173	0.805839	24.016792	1.741221
95	0.3	1	21.790109	21.464673	0.325436	21.117287	0.672822
95	0.3	3	29.352529	28.267777	1.084753	28.294022	1.058507
100	0.2	1	16.365174	16.079383	0.285791	15.995592	0.369582
100	0.2	3	23.446434	22.630930	0.815504	22.114877	1.331557
100	0.3	1	18.929590	18.605636	0.323954	18.489061	0.440528
100	0.3	3	27.107610	25.879891	1.227719	24.786934	2.320676
105	0.2	1	13.806390	13.528144	0.278246	13.390939	0.415450
105	0.2	3	21.348973	20.485845	0.863128	20.398862	0.950110
105	0.3	1	16.348214	16.015898	0.332317	15.881450	0.466765
105	0.3	3	25.024943	23.704871	1.320071	23.243381	1.781561

Tabel 10: Tabel over resultater med de optimale indstillinger i 2D.

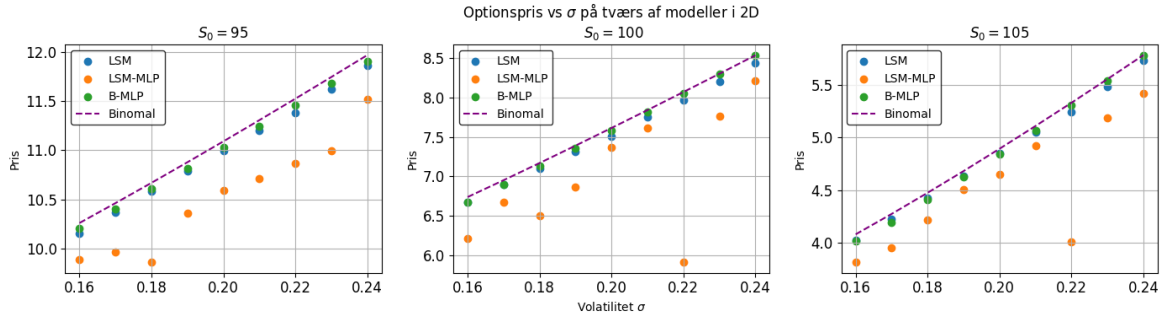
Vi ser, at begge algoritmer har relativt høje afvigelser fra binomial modellen, hvor de største afvigelser er helt oppe på 4.7% og 9.4% for hhv. LSM og LSM-MLP. Afvigelserne er generelt stigende i volatilitet og løbetid. Der er igen regnet standard error for 100 priser beregnet med hhv. LSM og LSM-MLP på atm optionen med $S_{1,0} = S_{2,0} = K = 100$, $T = 3$, $\sigma_1 = 0.3$ og $\sigma_2 = 0.4$. Standard error for LSM er også her i to dimensioner meget lavere på 0.005556 end for LSM-MLP på 0.040146. Stabiliteten for modellen visualiseres ligeledes på samme option og er vist i figur 17.



Figur 17: Stabilitetsgraf på to underliggende aktiver henover 100 prissætninger

Her tegner sig det samme billede som på den simple amerikanske put. LSM prisen er meget stabil, og LSM-MLP har meget høje afvigelser. Forskellen her er, at begge modeller prissætter dårligt med stor afvigelse fra referenceprisen. På den mere eksotiske put-minimum option har begge modeller altså høje afvigelser, men LSM er stadig at foretrække grundet stabiliteten.

Da B-MLP ikke er trænet på data, hvor volatiliteten er høj og løbetiden lang sammenligner vi performance af alle modeller på optioner med løbetid $T = 0.5$ år og stigende volatilitet i det ene aktiv med fast volatilitet i det andet på $\sigma_2 = 0.2$. Strike holdes fast på $K = 100$ og $S_{1,0} = S_{2,0}$. Dette er illustreret i figur 18.



Figur 18: Model priser vs stigende volatilitet med fast $K = 100$ og $T = 0.5$

LSM-MLP har også på kort løbetid og lavere volatiliteter høje afvigelser. Tabel 11 viser MAE for alle modellerne.

S_0	LSM	LSM-MLP	B-MLP
95	0.105990	0.570941	0.060599
100	0.084536	0.608943	0.027859
105	0.055992	0.389925	0.045086
\cup	0.082172	0.523270	0.044515

Tabel 11: MAE tabel for to dimensionelle modeller

Vi ser, at LSM-MLP har en MAE, der er mange gange højere end de to andre modeller. Ustabiliteten kommer her også til udtryk gennem de to outliers, som kan ses på atm og otm optionen.

LSM algoritmen performer markant bedre på den korte løbetid og lavere volatilitet. Den er ikke nok med at være stabil også nøjagtig, hvilket vi kan se ud fra den lave MAE i tabel 11.

B-MLP har den bedste performance på optionerne her. Når først modellen er trænet er den deterministisk, så den har ingen ustabilitet. Dens MAE ligger på ca. 0.061, 0.0279 og 0.0451, hvilket gør den til den mest nøjagtige prissætningsmetode på tværs af de tre start priser.

Hvis vi kigger på computationstiderne for de tre modeller er det ligeledes LSM og B-MLP, som performer bedst. I figur 14 og 15 ser vi, at LSM-MLP har en computationstid på 50 til 100 gange af,

hvad LSM har på en option med samme antal indfrielsestidspunkter. B-MLP's computationstid er beregnet ud fra 10 000 prissætninger, som giver et gennemsnit på 0.18 millisekunder.

Vi ser her altså, at B-MLP både er den mest præcise og hurtigste prissætningsmetode på optionerne med kort løbetid og lavere volatilitet. Dette tyder på, at med et godt nok datasæt, da kan B-MLP trænes til at være både hurtigere og nøjagtigere end de to andre modeller på et bredere udvalg af optioner. LSM-MLP ser igen ud til ikke at have nogen fordel over de to andre modeller her.

Nøjagtigheden af prisen kan dog ikke stå alene, når man evaluerer modeller til prissætning. Det er mindst lige så vigtigt, at modellen beskriver prisudviklingen i optionen på baggrund af stød i parametre såsom f.eks volatilitet, aktiekurs og tiden. Vi vil i det næste afsnit undersøge følsomheden overfor aktiekursen.

6 Delta hedging

Når man handler optioner, kan man mindske sin risiko ved at hedge sin position. Man hedger sin position ved at købe/sælge det underliggende aktiv. Hvor meget man skal handle i det underliggende aktiv er givet ved den første afledte af optionsprisen mht. det underliggende aktiv. Dette kaldes for optionens Δ . Lad optionens pris til tidspunkt t være beskrevet ved $P(t, S_t)$, så er Δ givet ved

$$\Delta(t, S_t) = \frac{\partial P(t, S_t)}{\partial S_t} \quad (6.1)$$

Det samme gør sig gældende for optioner, som afhænger af to aktiver

$$\Delta(t, S_{1,t}) = \frac{\partial P(t, S_{1,t}, S_{2,t})}{\partial S_{1,t}} \quad (6.2)$$

$$\Delta(t, S_{2,t}) = \frac{\partial P(t, S_{1,t}, S_{2,t})}{\partial S_{2,t}} \quad (6.3)$$

Modeller som LSM og LSM-MLP bruges til at estimere den optimale indfrielsesstrategi for en option. Prisen for optionen findes så ved at anvende denne strategi på stier med samme start pris, som den strategien er fundet for. Dette giver altså ikke et udtryk, som kan bestemme $P(0, S_0)$.

Stentoft og Létourneau fremstiller en metode i deres artikel "Simulated greeks for american options" [5], som kan komme uden om dette problem.

6.1 Delta hedging metode

Metoden beskrevet i [5] går ud på at genbruge den optimale indfrielsesstrategi fra (2.17). Man gemmer regressionsparametrene fra (2.16) for hvert indfrielsestidspunkt τ_k , hvormed man på et nyt sæt af simulerede stier bestemmer det sti-mæssige payoff $C(\omega, \tau_0, T)$ ved at følge den optimale indfrielsesstrategi. De nye stier er simuleret med forskellige S_0 startpriser, hvorfor man her ikke kan bestemme optionsprisen med et simpelt gennemsnit over de sti-mæssige payoffs. I stedet estimeres

prisfunktionen P ud fra startprisen og de sti-mæssige payoffs $P(\tau_0, S_{\omega, \tau_0}) = C(\omega, \tau_0, T) \cdot e^{-r(\tau_0^* - \tau_0)}$. Denne prisfunktion estimeres med en OLS estimator på de tilbagediskonterede sti-mæssige payoffs $C(\omega, \tau_0, T)$ og S_{ω, τ_0} :

$$P(\tau_0, S_{\omega, \tau_0}) \approx \beta_0 + \beta_1 S_{\omega, \tau_0} + \beta_2 S_{\omega, \tau_0}^2 + \dots + S_{\omega, \tau_0}^p \beta_p \quad (6.4)$$

hvor p beskriver graden af polynomiet, der estimeres med. Hermed kan man nu bestemme optionens Δ ved:

$$\Delta(\tau_0, S_{\omega, \tau_0}) = \frac{\partial P(S_{\omega, \tau_0})}{\partial S_{\omega, \tau_0}} \approx \beta_1 + 2\beta_2 S_{\omega, \tau_0} + \dots + S_{\omega, \tau_0} \beta_p^{p-1} \quad (6.5)$$

Dette estimat findes ikke kun for τ_0 , men derimod for alle $\tau_k \in \{0, \dots, \tau_k, \dots, \tau_K\}$, da Δ funktionen afhænger af både aktiekursen og den resterende løbetid for optionen. Dette er så metoden til at bestemme optionens Δ ud fra LSM algoritmen. For LSM-MLP er fremgangsmåden den samme - her benytter vi blot det trænedede netværk til hvert τ_k i stedet for regressionsparametrene.

Vi fitter i dette projekt P med et fjerde grads polynomium. Dette gør vi på baggrund af en kort numerisk undersøgelse, som viste en del ustabilitet med højere polynomiegrader.

For to underliggende aktiver estimeres P med følgende lineære regression

$$P(S_{1, \omega, \tau_k}, S_{2, \omega, \tau_k}) \approx \beta_0 + \beta \mathbf{X} \quad (6.6)$$

$$\text{hvor } \mathbf{X} = (S_{1, \omega, \tau_k}, S_{1, \omega, \tau_k}^2, S_{2, \omega, \tau_k}, S_{2, \omega, \tau_k}^2, S_{1, \omega, \tau_k} \cdot S_{2, \omega, \tau_k}, S_{1, \omega, \tau_k}^2 \cdot S_{2, \omega, \tau_k}, S_{1, \omega, \tau_k} \cdot S_{2, \omega, \tau_k}^2, S_{1, \omega, \tau_k}^2 \cdot S_{2, \omega, \tau_k}^2)$$

Δ funktionen er så givet som de afledte ligesom i ligning (6.2) og (6.3).

Optionens Δ i B-MLP kan bestemmes direkte ud fra det trænedede netværk. Eftersom et MLP neuralt netværk består af sammensatte funktioner (4.1 & 4.3), kan man ved brug af kædereglen differentiere det trænedede netværk mht. S_{ω, τ_k} . Grundet størrelsen af netværket er det dog ikke realistisk at finde det analytiske udtryk for den afledte, hvorfor vi i praksis vælger at bestemme sekanthældningen for sekanten mellem punkterne S_{ω, τ_k} og $S_{\omega, \tau_k} + \epsilon$. For ϵ tilstrækkelig småt estimerer dette den første afledte i S_{ω, τ_k} godt. Altså for et aktiv

$$\Delta(\tau_k, S_{\omega, \tau_k}) \approx \frac{F(S_{\omega, \tau_k} + \epsilon) - F(S_{\omega, \tau_k})}{\epsilon}$$

og for to aktiver

$$\begin{aligned} \Delta(\tau_k, S_{1, \omega, \tau_k}) &\approx \frac{F(S_{1, \omega, \tau_k} + \epsilon, S_{2, \omega, \tau_k}) - F(S_{1, \omega, \tau_k}, S_{2, \omega, \tau_k})}{\epsilon} \\ \Delta(\tau_k, S_{2, \omega, \tau_k}) &\approx \frac{F(S_{1, \omega, \tau_k}, S_{2, \omega, \tau_k} + \epsilon) - F(S_{1, \omega, \tau_k}, S_{2, \omega, \tau_k})}{\epsilon} \end{aligned}$$

Da Δ beskriver ændringen i optionsprisen givet et stød til prisen i det underliggende aktiv, kan vi opstille et hedging eksperiment for at evaluere Δ 'erne givet ved modellerne.

6.2 Hedging eksperiment

Vores hedging eksperiment vil give et mål for, hvor godt modellernes Δ beskriver den sande udvikling i optionsprisen. Vi opstiller en hedging portefølje V_ω for hver sti, som består af

$$V_{\omega, \tau_k} = \Delta(\tau_{k-1}, S_{\omega, \tau_{k-1}}) \cdot S_{\omega, \tau_k} + B_{\omega, \tau_{k-1}} \cdot e^{r(\tau_k - \tau_{k-1})} \quad (6.7)$$

$$\text{hvor } V_{\omega, \tau_0} = P(\tau_0, S_{\omega, \tau_0})$$

hvor B beskriver et risikofrit aktiv, som man køber/sælger af. For to aktiver er porteføljen givet ved

$$V_{\omega, \tau_k} = \Delta(\tau_{k-1}, S_{1, \omega, \tau_{k-1}}) \cdot S_{1, \omega, \tau_k} + \Delta(\tau_{k-1}, S_{2, \omega, \tau_{k-1}}) \cdot S_{2, \omega, \tau_k} + B_{\omega, \tau_{k-1}} \cdot e^{r(\tau_k - \tau_{k-1})} \quad (6.8)$$

$$\text{hvor } V_{\omega, \tau_0} = P(\tau_0, S_{1, \omega, \tau_0}, S_{2, \omega, \tau_0})$$

For hver periode opdateres porteføljen altså udfra aktivernes vækst. Ydermere, så justeres positionen i det risikofrie aktiv også hver periode. Dette er for et aktiv givet som

$$B_{\omega, \tau_k} = V_{\omega, \tau_k} - \Delta(\tau_k, S_{\omega, \tau_k}) \cdot S_{\omega, \tau_k} \quad (6.9)$$

og for to aktiver som

$$B_{\omega, \tau_k} = V_{\omega, \tau_k} - \Delta(\tau_k, S_{1, \omega, \tau_k}) \cdot S_{1, \omega, \tau_k} - \Delta(\tau_k, S_{2, \omega, \tau_k}) \cdot S_{2, \omega, \tau_k} \quad (6.10)$$

Da vores portefølje skal replikere prisen for optionen, vælger vi V_{ω, τ_0} til at være lig med optionsprisen regnet med den respektive model.

Desto bedre et Δ modellen har givet, desto mindre er forskellen mellem hedging porteføljens værdi og optionens payoff i indfrielsestidspunktet. Vi definerer følgende performance metrik for hhv. et aktiv og to aktiver:

$$H_\omega = \frac{e^{-r(\tau_0^* - \tau_0)}(V_{\omega, \tau_0^*} - \Phi(S_{\omega, \tau_0^*}))}{V_{\omega, \tau_0}} \quad (6.11)$$

$$H_\omega = \frac{e^{-r(\tau_0^* - \tau_0)}(V_{\omega, \tau_0^*} - \Phi(S_{1, \omega, \tau_0^*}, S_{2, \omega, \tau_0^*}))}{V_{\omega, \tau_0}} \quad (6.12)$$

hvormed vi har et udtryk for den sti-mæssige relative hedging afvigelse. Her beskriver τ_0^* det optimale indfrielsestidspunkt bestemt med indfrielsesstrategien fra modellen. Dog bestemmer vores B-MLP modellen ikke en indfrielsesstrategi, men bare en optionspris givet input. Indfrielsesstrategien for denne model vurderer vi på samme måde som i LSM algoritmen ved at trævle gennem perioderne bagfra. Fortsættelsesværdien bliver her bare bestemt af prisen fra B-MLP modellen

$$\mathbb{E}^Q[e^{-r(\tau_{k+1}^* - \tau_k)} C(\bar{\omega}, \tau_{k+1}, T) | S_{\bar{\omega}, \tau_k}] \approx K \cdot F(x) \quad (6.13)$$

$$\text{hvor } x^T = \left(\frac{S_{\bar{\omega}, \tau_k}}{K} \quad \sigma \quad T - \tau_k \right)$$

og

$$\mathbb{E}^Q[e^{-r(\tau_{k+1}^* - \tau_k)} C(\bar{\omega}, \tau_{k+1}, T) | S_{1, \bar{\omega}, \tau_k}, S_{2, \bar{\omega}, \tau_k}] \approx K \cdot F(x) \quad (6.14)$$

$$\text{hvor } x^T = \left(\frac{S_{1, \bar{\omega}, \tau_k}}{K} \quad \frac{S_{2, \bar{\omega}, \tau_k}}{K} \quad \sigma_1 \quad \sigma_2 \quad T - \tau_k \right)$$

Standard afvigelsen taget på H_ω henover alle stierne vil give os et mål for hvor godt et Δ de forskellige modeller har produceret.

$$\sigma_H = \sqrt{\frac{1}{n} \sum_{i=1}^n (H_{\omega_i} - \hat{\mu})^2} \quad (6.15)$$

$$\text{hvor } \hat{\mu} = \frac{1}{n} \sum_{i=1}^n H_{\omega_i}$$

Dette σ_H vil være et godt mål for vores modellers ydeevne out of sample. Algoritme 5 viser, hvordan hedging eksperimentet implementeres i praksis.

Data:

Stier: Simulerede stier, r : Risikofri rente, $\Delta(t, S_t)$: Delta funktion, P : Prisen på optionen

T : År til udløb, Cash_flow: Payoff til indfrielsestidspunkt for hver sti,

Indfri_tidspunkt: Indfrielsestidspunkt for hver sti

Result: Hedging fejl: H

n , perioder $\leftarrow \text{shape}(\text{stier})$;

$\Delta t \leftarrow \frac{T}{\text{perioder}-1}$;

$R \leftarrow e^{r\Delta t}$;

$H \leftarrow \underbrace{(0,0,\dots,0)}_n$;

$B \leftarrow \underbrace{(0,0,\dots,0)}_n$;

$V \leftarrow \underbrace{(0,0,\dots,0)}_n$;

for $i \leftarrow 0$ **to** $n-1$ **do**

$B[i] \leftarrow P - \Delta(0, \text{stier}[i,0]) \cdot \text{stier}[i,0]$;

end

for $i \leftarrow 1$ **to** perioder $- 1$ **do**

for $j \leftarrow 0$ **to** $n-1$ **do**

$V[j] \leftarrow \Delta(i-1, \text{stier}[j, i-1]) \cdot \text{stier}[j, i-1] + B[j] \cdot R$;

$B[j] \leftarrow V[j] - \Delta(i, \text{stier}[j, i]) \cdot \text{stier}[j, i]$;

if $i = \text{Indfri_tidspunkt}[j]$ **then**

$H[j] \leftarrow \frac{(V[j] - \text{Cash_flow}[j]) \cdot \frac{1}{R^i}}{P}$;

else

continue

end

end

end

return mean(hedging_fejl), std(hedging_fejl)

Algoritme 5: Hedging eksperiment for et aktiv

6.3 Hedging resultater

Vi har kørt hedging eksperimentet på alle vores modeller med to forskellige indstillinger. Den ene med kort løbetid og lav volatilitet, og den anden med lang løbetid og høj volatilitet. Vi forventer at se en højere hedging fejl σ_H på eksperimentet med lang løbetid.

Da vores B-MLP model er trænet på data med $T \leq 1$ og $\sigma \leq 0.25$, vil vi ikke inkludere denne model i eksperiment to.

Den optimale indfrielsesstrategi og Δ funktionen vurderes for LSM og LSM-MLP på to forskellige sæt af stier på $n = 500\,000$ for hvert aktiv. Herefter udføres eksperimentet på de samme $500\,000$ stier for alle modellerne. Dette er for at minimere Monte Carlo usikkerhed, som kan få den ene models resultater til at se bedre eller dårligere ud. Tabel 12 viser resultater fra det første eksperiment med $T = 1$, $\sigma_1 = 0.2$ og $\sigma_2 = 0.25$.

S_0	LSM 1D	LSM-MLP 1D	B-MLP 1D	LSM 2D	LSM-MLP 2D	B-MLP 2D
95	0.20678	0.18110	0.15780	0.13944	0.14036	0.13520
100	0.29070	0.26166	0.25276	0.19545	0.21748	0.19966
105	0.37271	0.33799	0.40128	0.27855	0.31330	0.30962

Tabel 12: Hedging fejl for hhv. amerikansk put og amerikansk put minimum option på $T = 1$, $\sigma_1 = 0.2$, $\sigma_2 = 0.25$, $K = 100$, $n = 500\,000$, $S_{1,0} = S_{2,0}$ og 50 årlige indfrielsestidspunkter.

Vi ser her generelt lavere hedging fejl på alle modellerne for itm optionerne. Vi bemærker, at for optioner, som er atm og otm, er fejlen en del større. En del af forklaringen for dette kan være, at det optimale indfrielsestidspunkt ligger senere i løbetiden end for itm og atm put optioner. Altså vil hedging perioden være længere, og fejlen vil derfor ofte være højere.

Det ses ydermere, at de to dimensionelle modeller har lavere fejl end de en dimensionelle, hvilket kan skyldes, at risikoen spredes ud på to aktiver. Dette gør hedging porteføljen mindre følsom overfor stød til kursen på et enkelt aktiv.

Vi observerer de laveste fejl for B-MLP på itm og atm optionen for både et og to aktiver. Dette er den samme trend, som vi så for prissætningen. Dog er fejlen for B-MLP en del højere for otm optionen for et aktiv, hvilket kan forklares ved figur 10, som viser større afvigelser for lavere optionspriser af B-MLP.

Fejlene på hhv. LSM og LSM-MLP modellerne er meget tætte, dog med LSM-MLP lidt bedre for et aktiv og LSM for to. Vi kører hedging eksperimentet igen, men med længere løbetid $T = 3$ og højere volatilitet $\sigma_1 = 0.3$ og $\sigma_2 = 0.4$ for at se, om dette tegner et tydeligere billede.

S_0	LSM 1D	LSM-MLP 1D	LSM 2D	LSM-MLP 2D
95	0.27436	0.37265	0.27517	0.25981
100	0.31307	0.32970	0.32167	0.31254
105	0.43019	0.39127	0.39235	0.36621

Tabel 13: Hedging fejl for hhv. amerikansk put og amerikansk put minimum option på $T = 3$, $\sigma_1 = 0.3$, $\sigma_2 = 0.4$, $K = 100$, $n = 500\,000$, $S_{1,0} = S_{2,0}$ og 50 årlige indfrielsestidspunkter.

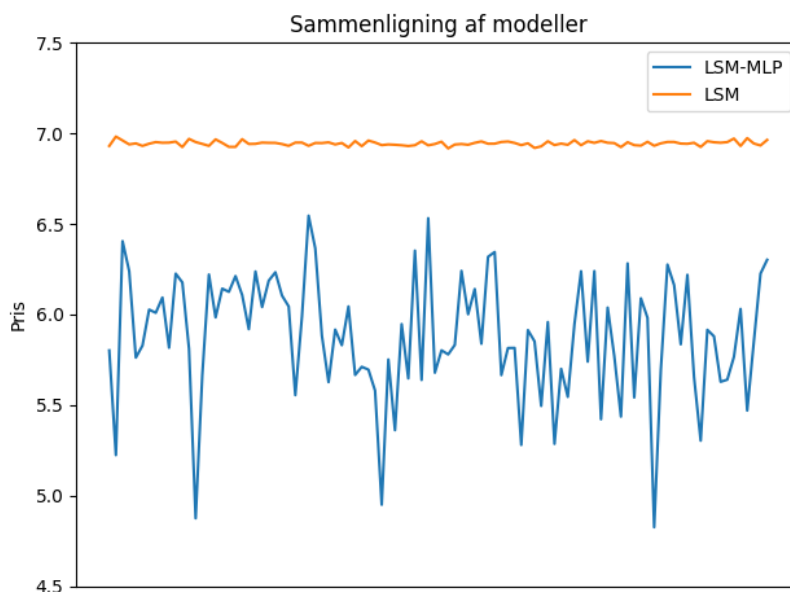
Her ser vi den omvendte trend for hedging fejlene for hhv. LSM og LSM-MLP. Der er altså ikke noget klart billede af, hvilken model der performer bedst out of sample. Dog ses det, at LSM samlet i tabel 12 og 13, har de laveste fejl. Dette er også det forventede, da LSM-MLP har vist en upræcis og ustabil prissætning. Denne prissætning er udledt af indfrielsesstrategien, som også bruges i hedging eksperimentet. Dette vil formentlig lede til større udsving i hedging fejlen. Ydermere, er hedging fejlen også et relativt mål givet optionsprisen fra modellen, hvilken LSM-MLP generelt underestimerer.

7 Diskussion

Vi har nu præsenteret samtlige resultater for modellerne og er klar til at samle op på dem. Ideen med at introducere et MLP netværk er, at det skal kunne fange ikke lineære sammenhænge i data. Payoff funktionen for optioner er netop ikke lineær, hvorfor vi forventede en forbedring af prissætningen. Vi observerede dog ingen klar forbedring. Tvært i mod blev især stabiliteten forværret i LSM-MLP. Vi havde en forventning om, at dette billede ville forbedres i flere dimensioner, hvor lineær regression ville være for simpel til at fange de mere komplekse sammenhænge i optioner med flere underliggende aktiver. For to underliggende aktiver var dette dog ikke tilfældet. Vi har undersøgt stabiliteten for LSM-MLP med fem underliggende aktiver. Hyperparametrene i LSM-MLP er de samme som for to aktiver og LSM regression er lavet med følgende designmatrix:

$$\begin{aligned} \mathbf{X} = & (S_{1,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k}^2, S_{2,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k}^2, S_{3,\bar{\omega},\tau_k}, S_{3,\bar{\omega},\tau_k}^2, S_{4,\bar{\omega},\tau_k}, S_{4,\bar{\omega},\tau_k}^2, S_{5,\bar{\omega},\tau_k}, S_{5,\bar{\omega},\tau_k}^2, \\ & S_{1,\bar{\omega},\tau_k} \cdot S_{2,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k} \cdot S_{3,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k} \cdot S_{4,\bar{\omega},\tau_k}, S_{1,\bar{\omega},\tau_k} \cdot S_{5,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k} \cdot S_{3,\bar{\omega},\tau_k}, \\ & S_{2,\bar{\omega},\tau_k} \cdot S_{4,\bar{\omega},\tau_k}, S_{2,\bar{\omega},\tau_k} \cdot S_{5,\bar{\omega},\tau_k}, S_{3,\bar{\omega},\tau_k} \cdot S_{4,\bar{\omega},\tau_k}, S_{3,\bar{\omega},\tau_k} \cdot S_{5,\bar{\omega},\tau_k}, S_{4,\bar{\omega},\tau_k} \cdot S_{5,\bar{\omega},\tau_k}) \end{aligned}$$

Resultatet ses i figur 19.



Figur 19: Stabilitetsgraf med fem underliggende aktiver.

$$S_{1,0} = S_{2,0} = S_{3,0} = S_{4,0} = S_{5,0} = 100, T = 1, K = 100 \text{ og} \\ \sigma = \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 \end{pmatrix}.$$

Stabiliteten ser fortsat ud til at være dårlig selv for fem underliggende aktiver. Det er dog vigtigt at nævne, at vi her hverken selv har undersøgt hyperparametre eller har kunne finde litteratur, som har undersøgt noget lignende for fem underliggende aktiver.

Vi så, at LSM-MLP havde hedging fejl på kort løbetid og lav volatilitet i en dimension, som var bedre end LSM. I to dimensioner havde den dog en højere hedging fejl. Her kommer ustabiliteten, som vi allerede har pointeret, igen til udtryk.

Modellen underperformer altså på alle parametre vi testede for i ift LSM.

Dette kan blandt andet skyldes et sub optimalt valg af hyperparametre i modellen. Vi har ikke valgt at bruge meget tid på at undersøge forskellige hyperparametre, hvorfor dette kunne være et interessant område at undersøge nærmere. Det kunne desuden være interessant at undersøge, hvorvidt modellens performance er bedre på andre typer af optioner, hvor payoff funktionen afhænger af for eksempel gennemsnitspriser henover en periode.

Som allerede gennem opgaven pointeret, viste B-MLP mere lovende resultater. På optioner med kort løbetid og lav volatilitet så vi gode resultater. Dette kombineret med en meget lav computationstid gør den til et muligt alternativ til LSM. Især på optioner med længere løbetid og høj volatilitet, hvor vi så, at nøjagtigheden af LSM modellen blev svækket. Vi har ikke haft tilstrækkelig med data tilgængelig til at træne modellen på disse typer af optioner, men forventer ikke at performance forværres så længe træningsdata er godt nok.

B-MLP viste også gode resultater i vores hedging eksperiment, hvor vi fik lavere hedging fejl end de andre modeller. Vi observerede blot for otm optionen med et underliggende aktiv, at hedging

fejlen var en del dårligere sammenlignet med LSM og LSM-MLP. Vi så i figur 10, at B-MLP havde udfordringer på optioner med en lav pris, hvilket otm optioner med kort løbetid har. Vi finder det derfor ikke overraskende, at hedging fejlen er større her. Alt i alt understreger hedging eksperimentet, at modellen både prissætter nøjagtigt og producerer et godt Δ .

Også med B-MLP kunne det være interessant at lave en større hyperparameter undersøgelse for at forbedre algoritmen.

En ulempe ved B-MLP er, at den tager lang tid at træne, hvis datasættet er stort, hvilket er et krav for en god model. Det mest tidskrævende ved B-MLP er dog dataindsamlingen. Uagtet om modellen trænes på simuleret data ligesom vi har gjort i projektet, eller om man ønsker at træne modellen på data fra den virkelige verden. Simulering kræver meget computerkraft, og virkelig data kræver en høj grad af data cleaning.

I praksis, hvis man ønsker at træne modellen på virkelig data, ville modellen skulle gentrænes ofte. Dette skyldes, at markedsomstændigheder er i konstant ændring i takt med teknologisk udvikling. Dette er dog ikke et større problem for store virksomheder, da de har ressourcerne og computerkraften til at opbygge store og komplicerede machine learning modeller, som hurtigt kan gentrænes med nyt data. Disse virksomheder har som regel også adgang til det bedste og nyeste data fra de finansielle markeder.

Vi har brugt binomial modellen som reference igennem hele projektet, og denne model vil altså stadig være den mest fornuftige givet en simpel option indenfor Black Scholes modellen. Dog, som vi har vist tidligere, så fungerer binomial modellen ikke optimalt for flere stokastiske parametre. Ydermere, så kan binomial modellen ikke gøre brug af sin effektive træstruktur ved optioner med payoff som f.eks. afhænger af gennemsnitskurser over tid eller andre stiafhængige optioner. Her viser LSM virkelig nogle stabile og præcise resultater. Fordelen ved LSM modellen er også, at den er simpel, kan bruges i alle finansielle modeller, på flere stokastiske variabler og på alle typer af payoff funktioner.

8 Konklusion

Vi har nu set forskellige metoder til at prissætte optioner inden for Black Scholes teorien. Fokus har været på amerikanske put og put minimum optioner på to underliggende aktiver. Binomial modellen har virket som reference pris for at undersøge, hvorvidt neurale netværk kan forbedre prissætnings resultaterne fra LSM.

Ud fra vores resultater tyder det ikke på, at et MLP neuralt netværk kan forbedre LSM algoritmen. En større hyperparameter undersøgelse er dog nødvendig for at afskrive brugen af MLP netværk til at estimere fortsættelsesværdier komplet.

LSM er fortsat den foretrukne simuleringsmodel, eftersom den er nøjagtig og stabil med mulighed

for anvendelse i andre modeller end Black Scholes.

B-MLP viste lovende resultater i forhold til nøjagtighed, computationstid og især skalering til flere dimensioner. I takt med teknologisk udvikling bliver machine learning modeller kun bedre, hvorfor vi mener, at modeller som B-MLP vil blive mere og mere udbredt.

Referencer

- [1] T. Björk. *Arbitrage theory in continuous time*. Oxford university press, 2009.
- [2] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.
- [3] P. P. Boyle, J. Evnine, and S. Gibbs. Numerical evaluation of multivariate contingent claims. *The Review of Financial Studies*, 2(2):241–250, 1989.
- [4] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of financial Economics*, 7(3):229–263, 1979.
- [5] P. Létourneau and L. Stentoft. Simulated greeks for american options. *Quantitative Finance*, pages 1–24, 2023.
- [6] F. A. Longstaff and E. S. Schwartz. Valuing american options by simulation: a simple least-squares approach. *The review of financial studies*, 14(1):113–147, 2001.
- [7] R. C. Merton. Theory of rational option pricing. *The Bell Journal of economics and management science*, pages 141–183, 1973.
- [8] B. Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.
- [9] P. Pommergård Lind. Classical option pricing theory and extensions to deep learning. pages 1–99, 2020.
- [10] R. Stulz. Options on the minimum or the maximum of two risky assets: analysis and applications. *Journal of Financial Economics*, 10(2):161–185, 1982.