Amerikanske optioner og finansielle beregninger

Rolf Poulsen

Med den hårdtarbejdende redaktions egne ord så *tilstræbes det, at artikler i* FAMØS *henvender sig til alle på IMF (ansatte og studerende).* Jeg vil dog straks begrænse nærværende skriverier til den delmængde af førnævnte, der opfylder mindst en af følgende ikke-disjunkte betingelser: (a) har et åbent sind, (b) har haft Fin1, (c) har læst [Lando (1999)]. Hovedresultatet i bemeldte fortrinlige artikel kan fortælles bemærkelsesværdigt kort i arbitrageprisfastsættelsesteoriens første[10] hovedsætning: En finansiel model (π, δ, r) (*priser, dividender, rente*) på den stokastiske basis $(\Omega, \{\mathcal{F}_t\}_{t=0,1,\ldots,T}, P)$ (*udfaldsrum, filtrering, sandsynlighedsmål*) er arbitragefri, hvis og kun hvis den har et ækvivalent martingalmål Q (*risiko-neutralt mål*).

Et centralt element i en videnskabelig artikel er at placere sig i forhold til litteraturen. Denne er ingen undtagelse. Men de friere tøjler tillader mig at forstå litteratur på samme måde som folk udenfor den akademiske verdens beskyttede værksted. Nemlig som — litteratur: I viktorianske kriminalromaner afsløres i det sidste kapitel, hvor obersten, de uvorne børn, gammeljomfruerne

[10] Som ordenstalsbetegnelsen indikerer, er der også en hovedsætning nummer to. Den siger, at en arbitragefri model er komplet, hvis og kun hvis Q er entydigt. Fra et praktisk synspunkt er dette mere vigtigt, da det fortæller at enhver (ny) kontrakt kan replikeres. Den har derfor kun en eneste fornuftig pris. De konkrete modeller, vi møder i denne artikel, er komplette. Men beregningsmæssige teknikker og trick er ofte uahængige af hvilket af de måske mange martingalmål, man benytter. Der er ikke nogen almindeligt accepteret tredje eller højere hovedsætning.

mv. er forsamlet i biblioteket, hvordan og hvorfor butleren gjorde det. På samme måde vil jeg tilsidst afsløre, hvad amerikanske optioner burde blive kaldt. Undervejs skal vi så grueligt meget igennem, dvs. jeg kommer ind på en række emner fra min forskning. Med en karikeret '70-formulering "siger det enormt meget om samfundet". Holdbarheden af litteratur med dette som hovedmål har vist sig tvivlsom. Jeg lader det være op til læseren, om det bliver bedre eller værre, hvis man som overordnet princip søger at bytte "litteratur" ud med "forskning".

Arbitragefri prisfastsættelse: tænk globalt, regn lokalt

Først omskriver vi den globale definition af et martingalmål fra [Lando (1999)] til en lokal karakterisation. Vi betragter et vilkåligt[11] aktiv med en givet dividende-proces δ tilpasset vores filtrering og lader $\tilde{}$ angive diskontering med det lokalt riskofrie aktiv, dvs. division med $R_{0,t} = (1 + r_0) \cdot (1 + r_1) \cdot \ldots \cdot (1 + r_{t-1})$, hvor r er den — muligvis stokastiske — korte rente. I en arbitragefri model skal aktivets prisproces, π, opfylde

$$
\begin{aligned}
\widetilde{\pi}(t) &= \mathbb{E}_t^Q \left(\sum_{j=t+1}^{T} \widetilde{\delta}(t) \right) \\
&= \mathbb{E}_t^Q \left(\frac{\delta(t+1)}{R_{0,t+1}} \right) + \mathbb{E}_t^Q \left(\underbrace{\mathbb{E}_{t+1}^Q \left(\sum_{j=t+2}^{T} \widetilde{\delta}(t) \right)}_{=\widetilde{\pi}(t+1)} \right),
\end{aligned}
$$

[11] "Arbitrært" kunne man sige, men det ord er belastet i netop denne sammenhæng.

hvor vi har brugt reglen om itererede forventninger (eller: tårnegenskaben) og linearitet af betinget forventning. Heraf får vi ved brug af definitionen af R-processen — specielt at $R_{0,t+1}$ er $\mathcal{F}_t$-målelig — at

$$\pi(t) = \frac{1}{1+r_t}\mathbb{E}_t^Q(\pi(t+1)+\delta(t+1)). \tag{1}$$

Denne lokaliserede ligning (t på venstresiden, kun $t+1$ på højresiden) er overmåde nyttig. Den er hjertet i effektive beregninger i modeller med Markov-struktur, den fortæller, hvordan vi håndterer dividendebetalinger og stokastiske renter,[12] og den er central for at forstå, hvordan amerikanske optioner prisfastsættes.

Amerikanske optioner: mere subtilt end som så?

En europæisk option giver dens ejer ret, men ikke pligt, til at gøre et eller andet på et (eneste) på forhånd fastlagt tidspunkt T (med svag misbrug af notation). Som regel er valget trivielt, og vi siger derfor blot, at ejeren modtager beløbet $g(S(T))$ på tidspunkt T, hvor g er payoff-funktionen. Fx er $g(x) = \max(K-x,0) =: (K-x)^+$ for en strike-K put-option,[13] hvor ejeren har ret, men ikke pligt, til sælge det underliggende aktiv til prisen K. For en amerikansk option vælger ejeren selv, hvornår optionen skal indfries, lad os kalde tidspunktet τ, hvorefter han (straks) modtager $g(S(\tau))$. Ordene amerikansk og europæisk har her ingen geogra-

[12] En fordel ved den diskrete ramme er, at vi får det hele i et hug; studerende i FinKont-kurser kan skrive under på, at så let går det ikke altid.

[13] Hvis det underliggende aktiv ikke udbetaler dividender i optionens løbetid (og renten er positiv), så er det aldrig optimalt at indfri en amerikansk call-option førtidigt. Derfor er put-optionen det gennemgående eksempel her.

fisk betydning; vi kunne ligeså godt kalde optionerne orange eller varmforzinkede.[14]

Ejeren af en (endnu ikke indfriet) amerikansk option skal hele tiden overveje, om hans option er mere værd "død end i live". Hvis han undlader at indfri og derved holder optionen i live, så får han et lignende valg i næste periode. Optionsejeren gør, hvad der er mest fordelagtigt, og oversat til ligninger giver det en dynamisk programmeringsformulering eller Bellman-ligning for prisen, π^A, på en amerikansk option:

$$\pi^A(t) = \max\left\{g(S(t)), \frac{1}{1+r_t}\mathbb{E}_t^Q(\pi^A(t+1))\right\}. \tag{2}$$

Jeg håber ovenfor at have kommunikeret det intuitivt rimelige i ligning (2), og som vi om lidt vil se, er det kodningsmæssigt lettere gjort end sagt. Men matematisk/teoretisk er det mere subtilt. Thi: Hvis det er 1. led på højresiden i (2) (kaldet indfrielsesværdien; 2. led kaldes tidsværdien), der er størst, så strider det lodret imod ligning (1). Ups. Der, hvor kaninen forsvinder ned i hatten, er at (1) udtaler sig om priser for aktiver, hvis betalinger er givet på forhånd. Men altså igen: "givet" som tilpasset stokastisk proces, dvs. når et træ repræsenterende informationsstrukturen er blevet lavet, så kendes aktivets betalinger i hver eneste knude. For en amerikansk option har ejeren derimod en yderst ikke-triviel indvirkning på betalingsprocessen. Ikke desto mindre kan man vise, at (2) vitterlig giver den eneste arbitragefri amerikanske op-

[14] En historie fortæller, at den (amerikanske) økonom Paul Samuelson som den første brugte betegnelsen "amerikansk" om optioner med førtidige indfrielsesrettigheder. Han gjorde det for at signalere noget bedre, noget mere avanceret. Selveste internettet har dog været ude af stand til at bekræfte den historie, så ...

```
S0<-100; r<-0.03; alpha<-0.07; sigma<-0.20
expiry<-1; strike<-100
n<-expiry*252; dt<-expiry/n
u<-exp(alpha*dt+sigma*sqrt(dt)); d<-exp(alpha*dt-sigma*sqrt(dt))
R<-exp(r*dt)
q<-(R-d)/(u-d)
```

```
S<-put<-matrix(0,nrow=(n+1),ncol=(n+1))
put[,n+1]<-pmax(strike-S0*u^(0:n)*d^(n:0),0)
for (i in n:1) {
  for (j in 1:i){
    S[j,i]<-S0*u^(j-1)* d^(i-j)
    put[j,i]<-max(max(strike-S[j,i],0), (q*put[j+1,i+1]+(1-q)*put[j,i+1])/R)
  }
}
```

Tabel 1 R-kode med en naturlig algoritme til prisfastsættelse af en amerikansk put-option. For at prisfastsætte den europæiske option fjernes den *kursiverede* del af koden. Koden under stregen vil senere rykke ud for at forøge beregningshastigheden. Koden ligger i www.math.ku.dk/~rolf/FAMOES

tionspris.[15] Det kræver en udflugt til optimale stoppetider, hvilket gøres glimrende i kapitel 2 i [Lamberton & Lapeyre (1996)]. Og i Fin2.

Amerikanske optioner: nemmere gjort end sagt

For at illustrere, hvordan man faktisk regner amerikanske optionspriser ud, lad os da trække en arbejdshest af stalden: standardbinomialmodellen, som vist i Figur 1. Her udvikler aktiekursen sig

[15] En anden detalje, der afsløres, når man arbejder sig igennem beviset for (2), er at vi rettelig burde sige, at $\pi^A(t)$ er tid-t prisen på den amerikanske option *givet den endnu ikke er blevet indfriet.*

som

$$S(t+\Delta t) = S(t) \times \begin{cases} \exp(\alpha\Delta t + \sigma\sqrt{\Delta t}) =: u & \text{m/ ssh } \frac{1}{2} \\ \exp(\alpha\Delta t - \sigma\sqrt{\Delta t}) =: d & \text{m/ ssh } \frac{1}{2} \end{cases},$$

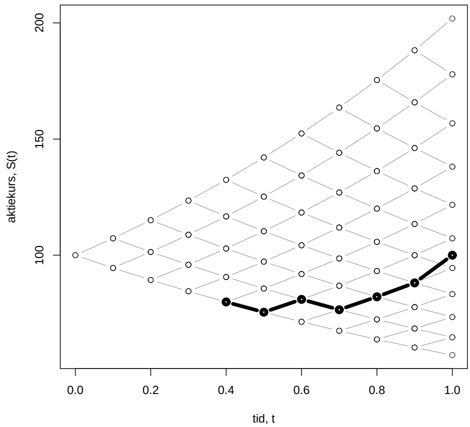
hvor α og σ er konstanter. Aktieafkastenes spredning bestemmes af σ, der kaldes volatiliteten. Igen har vi foretaget en subtil udvidelse af modelrammen: Vi kan ikke blot variere antallet af skridt, men også og uafhængigt heraf tidsskridtlængden Δt. Man kan spørge, hvorfor tidsskaleringen er som den er; specielt hvorfor $\pm\sigma\sqrt{\Delta t}$ modellerer usikkerheden. Det vil dog føre for vidt at komme ind på det her, men tro mig: Det er det.[16] Vi antager, renten er konstant og parametriserer diskonteringsfaktoren som $R_{0,t} = R^{t/\Delta t}$, hvor $R = e^{r\Delta t}$. Den betingede Q-op-sandsynlighed er den samme i alle knuder,

$$q = \frac{R-d}{u-d}.$$

Taylor-udvilker man q i Δt ser man, at led af formen $\sigma\sqrt{\Delta t}$ dominerer for små tidsskridt (hvor $\sqrt{\Delta t} \gg \Delta t$). Kombineres det med en tilpas kraftig version af den centrale grænseværdisætning, se kapitel 7 i [Lando & Poulsen (2006)], kan vise man, at α asymptotisk ingen effekt har på optionspriser — og derfor ofte udlades. Sandsynlighedsteoretikere kan skimte Girsanovs sætning her.

Tabel 1 giver en R-implementation af en naturlig algoritme til udregning af prisen på en amerikansk put-option. R er et *open source* programmeringssprog, det kan downloades fra `www.`

[16] En indikation af rimeligheden kan man få således: Benyt empiriske horisont-h log-afkast, $A_h(t) = \ln(S(t\Delta)/S(t-h\Delta t))$, til at estimere α og σ. Foretag estimationen for forskellige horisonter, h, og se, at α og σ er nogenlunde upåvirkede heraf.



Figur 1 Aktiekursgitter for standardbinomialmodellen (med 10 tidsskridt og iøvrigt parametre som i Tabel 1). De fede knuder (og aht. grafiske udtryk: linjerne, der forbinder dem) angiver den kritiske indfrielsesgrænse for en amerikansk strike-100 put-option: Første gang aktiekursen falder til dette niveau, indfries optionen.

`r-project.org` og er uhyre nemt at installere. Som noget meget nyttigt ifm. forskningssamarbejde eller projektvejledning, så er R-kode "meget transportabel". Man kan køre hinandens programmer uden den udvidede eksamen som systemadministrator. R har stærke statistik-værktøjer (med en bred fortolkning af statistik, der inkluderer fx lineær algebra og grafik) og minder syntaksmæssigt om Matlab. Det er, hvad dataloger kalder et *domain-specific language* (DSL); det retter sig imod brugere, der skal løse bestemte typer opgaver, og har så nyttig funktionalitet hertil indbygget.

Modsætningen til DSL er *general-purpose languages* (GPL) som fx C++, Java eller Python. Jeg kalder som regel DSL for højniveau-sprog og GPL for lavniveau-sprog. Datalogiske pedanter kan sikkert give grunde til, at jeg ikke burde gøre det.

I prisfastsættelsesalgoritmen trævler vi os rekursivt baglæns fra det senest mulige slutidspunkt, idet vi hele tiden udnytter, at vi netop har beregnet $(t+1)$-elementet i ligning (2). Udvidelsen ift. prisfastsættelse af den europæiske option beløber sig til en halv linjes kode. Førtidig indfrielse af put-optionen sker for lave aktiekurser. For ethvert (tilstrækkeligt stort) tidspunkt, t_i, findes i gitteret et maksimalt tid-t_i aktiekursniveau, $S^*(t_i)$, således at for aktiekurser under $S^*(t_i)$ indfries optionen. Kurven $t \mapsto S^*(t)$ kaldes den kritiske indfrielsesgrænse, en illustration ses i Figur 1.[17] Den optimale strategi for optionsejeren — der ikke kan se ind i fremtiden — er at indfri optionen første gang, den kritiske indfrielsesgrænse rammes. Som beregnere bevæger vi os altså fra højre mod venstre i gitteret, som optionsejere fra venstre mod højre.

Amerikanske optioner: rettidig omhu i regnerier

Der er en række måder at forbedre beregningstiden for den naturlige algoritme i Tabel 1:

1. Hav nyeste version installeret. For "anstændig" software sker der ofte forbedringer bag scenen, dvs. skjult for slutbrugeren,

[17] Den opmærksomme læser ser, at vi på indfrielseskurven har tegnet punktet (udløbstid, strike) (eller: (T, K)) skønt dette ikke er et gitterpunkt. Man får glattere konvergens (for $\Delta t \to 0$) af (specielt europæiske) optionspriser, hvis man (ved valg af skridtlængde) sikrer, at K altid ligger på et gitterpunkt. Den rene af sind påpeger det inkonsistente i at bruge modeller for det underliggende aktiv, der afhænger af den option, man skal prisfastsætte. *Yeah, well* ...

```
qUR<-q/R; qDR<-(1-q)/R;  u.pow<-u^(0:n); d.pow<-d^(0:n)
S<-S0*u.pow[1:(n+1)]*d.pow[(n+1):1] ; put<-pmax(strike - St,0)
for (i in n:1) {
  S<-S0*u.pow[1:i]*d.pow[i:1]
  put[1:i]<-pmax(strike-S,(qUR*put[2:(i+1)]+qDR*put[1:i]))
}
```

Tabel 2 Hastighedsforbedret kode til prisfastsættelse af en amerikanske put-option. Koden ovenfor erstatter koden under stregen Tabel 1,

og de har positiv indvikning på tidsforbruget. Her betød et skift fra R 2.11.1 til R 2.14.0 omtrent en halvering af kørselstiden.

2. Undgå unødvendige beregninger. Flyt så mange funktionskald som muligt udenfor løkker. Her gav det en ca. 10% forbedring af kørselstiden. Dimensionèr i videst muligt omfang vektorer og matricer på forhånd. Højniveau-sprog tvinger typisk ikke brugeren til dette. Det kan være behageligt, da det ligger i numeriske metoders natur, at mange ting først bliver kendt, efterhånden som programmet løser ens opgave. Men der kan være tid at spare. Et konkret R-eksempel: Skriver man skriver cbind inde i en løkke, skal en rød lampe lyse.
3. Brug indbygget funktionalitet. Her specielt: Udnyt R's evne til at arbejde med vektorer og matricer, undgå element-for-element-operationer.[18] Her nedbringer det beregningstiden med omkring en faktor 7.

[18] Ikke alle funktioner virker på fler-dimensionale objekter på den måde, du måske tror/håber. Lad fx x være en vektor og sammenlign max(x,0) med pmax (x,0). Eller lad X og Y være matricer og sammenlign X*Y med X%*%Y.

252 skridt

R-kode		C++-kode	
Metode	Tid (ms)	Metode	Tid (ms)
Windows; Tabel 1; R 2.11.1	~ 750	Visual Studio; ~Tabel 1	9
Windows; Tabel 1; R 2.14.0	720	Visual Studio; ~Tabel 2	1,5
Windows; Tabel 2; R 2.11.1	~ 30		
Windows; Tabel 2; R 2.14.0	15		
Linux; Tabel 1; R 2.11.1	(fys., *p*) 458 (fys., *o-p*) 260	GNU g++ ; ~Tabel 1	35
	(CPU, *p*) 126 (CPU, *o-p*) 119	GNU g++ ; ~Tabel 2	6,0
Linux; Tabel 2; R 2.11.1	(fys., *p*) 209 (fys., *o-p*) 57		
	(CPU, *p*) 55 (CPU, *o-p*) 51		
Europæisk B/S-formel	0,04		
10^6 B/S-sim. $\rightsquigarrow \pm$ 0,3%	310		
10^4 Tabel 4 stier $\rightsquigarrow \pm$ 1,4%	65.000		

(30 × 252) skridt

R-kode		C++-kode	
Metode	Tid (s)	Metode	Tid (s)
Windows; ~Tabel 1; R 2.14.0	600	Visual Studio; ~Tabel 2	2,3
		GNU g++ ; ~Tabel 2	12
Windows; Tabel 2; R 2.14.0	6,8		

Tabel 3 Beregningstider for algoritmer til prisfastsættelse af en amerikansk put-option i standarbinomialmodellen. Parametrene er som angivet i Tabel 1. Al kode findes i `www.math.ku.dk/~rolf/FAMOES`. Windows-kørslerne er foretaget på en Intel P8700 Core2Duo CPU med 2.5 GHz og 3 GB RAM. Eller mere forståeligt: På min et par år gamle computer, der da den var ny, var ret kraftig af en bærbar at være. Linux-kørslerne er fra IMF's Shannon-server, og her indikerer *p*/*o-p* om kørslen skete på et travlt tidspunkt eller ej (*peak*/*off-peak*).

I de meget læseværdige [Higham (2002)] og [Eddelbuettel (2009)] kan man finde mere. Køreklar R-kode ses i Tabel 2. Man kan rettelig mene, at ovenstående liste mangler et punkt: ”0. Kommenter/dokumenter dine programmer.” Lad os sige grunden til, at det ikke er gjort her, er hensynet til det typografiske udtryk.

Selv Schrödinger kunne ikke svinge en måske død kat i Tabel 3 uden at ramme en beregningstid. Allerførst ser vi, at med de forholdsvis simple justeringer nævnt ovenfor, falder tiden, det tager at udregne prisen på en 1-årig amerikansk put-option i en standardbinomialmodel med daglige skridt ($\Delta t = 1/252$), fra 750 til 15 millisekunder, dvs. med en faktor 15. Der er beregningstider både fra en (Windows-)PC og fra IMF's (Linux-)server Shannon. Når man skal dele regnekraften med andre, så er der forskel på den tid, det fysisk tager at køre programmet, og den tid, det ville have taget, hvis man havde tingene for sig selv (CPU-tiden). Der kan derfor være tydelige fordele (her en faktor 4) ved at køre sine programmer udenfor spidsbelastningsperioder, dvs. om natten og i weekenden.

En skeptiker kunne nu sige: “Ja, jo, men er det besværet værd? Forbedringen er jo mindre end et sekund.” Til det behøves bare henvises til nedre del af Tabel 3. Her beregnes en amerikansk optionspris for en 30-årig option i en model med daglige skridt; skal man eksempelvis prisfastsætte konverterbare realkreditobligationer, så er det en relevant problemstørrelse.[19] Her tager Tabel 2-koden omkring 7 sekunder, mens tidsforbruget for Tabel 1-koden er 10 minutter.[20] Så er forskellen pludselig til at få øje på, når man

[19]Som den engelske komiker Andy Zaltzman plejer at sige i den fremragende og stærkt vanedannende podcast *The Bugle*: “That's a lie, but my point stands.”

[20]Antallet af knuder i et binomialgitter vokser kvadratisk i antallet af tidsskridt. Man ville derfor forvente, at en 30-årig option tog $30^2 = 900$ gange så

sidder foran computeren. Og hvis man så ikke-spor-hypotetisk (se [Pedersen, Weissensteiner & Poulsen (2011)]) skal udregne ikke 1, men 5.000 priser, der skal indgå i et dynamisk porteføljevalgsproblem, så kan den ene beregning køres over en nat, mens den anden tager en måned. Så selv med dagens computere kan beregningstid blive en flaskehals. Det betyder, at der er al mulig grund til forske i forbedrede berergningsmetoder.

En ide er, naturligvis, at man med blyant og papir leder efter lukkede formler. Det kan give dramatiske hastighedsforøgelser; eksempelvis ser vi i Tabel 3, at det kun tager 0,04 millisekund at udregne en optionspris fra Black-Scholes-formlen (der dog kun virker for europæiske optioner). En anden mulighed er at formulere sine prisfastsættelsesproblemer vha. partielle differentialligninger og så angribe dem numerisk. Binomial-metoden kan ses som en numerisk løsningsmetode, men som sådan er den en absolut letvægter, der findes væsentligt bedre teknikker. En tredje tilgangsvinkel er Monte Carlo-simulationsteknikker, eller i min forskningsverden blot: simulation. Dette er en bred betegnelse for beregningsmetoder, hvor tilfældige tal — samt asymptotiske resultater som fx store tals lov — spiller en stor rolle. Monte Carlo henviser så til casinoet sammesteds,[21] hvor der er masser af tilfældige udfald — men hvor vi godt ved, hvordan det går i det lange løb. Simulation er som udgangspunkt langsomt. Tabel 3 viser,

---

lang tid at prisfastsætte som en 1-årig. For den naturlige metode passer det så godt, som det nogenlunde kan. Den 30-årige optimerede R-kode er noget hutigere end "forventet", $6,8 < 13,5 = 0,0015 \times 30^2$; det skyldes at de 15 millisekunder inkluderer nogle faste beregningsomkostninger eksempelvis til kompilering. C++-koden taber omvendt hastighed, $2,3 > 1,35 = 0,0015 \times 30^2$. Grunden er ikke helt klar for mig, måske ekstra omkostninger til hukommelsesallokering ifm. lange vektorer.

[21] For administrative feinschmeckere er Monte Carlo et kommunaldisktrikt i fyrstendømmet Monaco.

at for at ramme en europæisk put-optionspris indenfor 1% skal der omkring en halv million udfald til. Det tager nogle hundrede millisekunder. Og det kræver endda, at man kan simulere $S(T)$ direkte, dvs. ikke skal lave sti-vise beregninger. Skal man det, så eksploderer beregningstiden. Den simulationsbaserede amerikanske optionsprisfastsættelseskode fra Tabel 4 tager 65.000 millisekunder. (Og den regner ikke engang det rigtige ud, men det har ikke noget med hastigheden at gøre.) Simulationsmetoders styrke ligger i deres alsidighed; mange mere komplicerede modeller og problemstillinger har de let ved at håndtere. Og ofte[22] er det vigtigere at have et approksimativt svar på det rigtige spørgsmål, end et ultra-præcist svar på et forkert spørgsmål.

En fjerde mulighed er at skifte hest ved at programmere prisfastsættelsesalgoritmerne i et lavniveau-sprog; C++ er den finansielle sektors foretrukne. Det stiller større krav til programmeringsfærdigheder, men belønningen skulle gerne være forøget hastighed; brugervenlighed i interfacet og ekstra funktionalitet koster. Algoritmerne fra Tabel 1 og 2 er stort set bare løkker indeni hinanden, så de er nemme at implementere i C++. I højre søjle i Tabel 3 ses beregningsstider. Man opnår en tydelig forbedring (under Windows er det 15 vs. 1,5 millisekunder for en 1-årig option og 6,8 vs. 2,3 sekunder for en 30-årig option). Men det gode højniveau-sprog er ikke håbløst bagud. Sammenlignes en naiv implementering (ala Tabel 1) i Microsofts Visual Studio-miljø med den optimerede R-kode, så er forskellen blot 10 vs. 15 millisekunder for 1-årige optioner. For 30-årige optioner er R-kodens tider en faktor 3 fra C++-kodens. Meget groft sagt er beregningstids-

[22] Man kan mene, at "ofte" burde være "altid". Det synspunkt afspejles dog ikke udpræget i vore uddannelser.

forbedringen altså her halvanden størrelsesorden[23] ved, at man tænker sig om indenfor R, og en halv størrelsesorden ved, at man skifter R ud med C++. Så det er ikke en datalogisk dødssynd, at lade det afhænge af situationen,[24] om man skønner det umagen værd at bruge C++. I sådanne overvejelser er et realistisk forhold til egne programmeringsevner tilrådeligt.

Amerikanske optioner og simulation: hvordan ikke og hvordan så?

Et af de store forsknings- og anvendelsesområder indenfor kvanititativ finansierngsteori de seneste 10 år har været brugen af simulation til prisfastsættelse af amerikanske optioner. Det ”folkelige gennembrud” kom med [Longstaff & Schwartz (2001)]. Denne artikel var dog langt fra den første, der kombinerede amerikanske optioner og simulation. Forfatterne til eksempelvis [Carriere (1996)], [Broadie & Glasserman (1997)] og [Andersen (1999)] kan med en vis ret mugge over, at have fået stjålet rampelyset. Jeg tror en meget vigtig grund til, at det er [Longstaff & Schwartz (2001)], der løber med citationerne, er at forfatterne turde bruge — selv i et af de mest prestigefyldte tidsskrifter — de første 10 sider til detaljeret at gennemgå et simpelt taleksempel; 3 perioder, 8 stier, alle kan være med. Men inden jeg når til Longstaff og Schwartz' (simulation-regression)-ide, vil jeg først forklare, hvorfor det *er*

[23] Med det bevidst løse begreb “størrelsesorden” mener jeg noget ala ”ti-potens til forskel”, så faktor 10 er en størrelesorden, faktor 100 er to, og faktor $3 \approx 10^{0.5}$ er en halv.

[24] Et sted det *er* umagen værd at bruge C++, eller i hvert fald at kalde kompileret kode fra R, er ved løsning partielle differentialligninger med differensmetoder.

```
Nsim<-10000
DiscPayoff<-rep(0,Nsim)
S<-rep(S0,(n+1))

for (j in 1:Nsim){
  UpMoves<-cumsum((runif(n) < q))
  S[2:(n+1)]<-S0*u^UpMoves*d^((1:n)-UpMoves)
  SnellZ<-gS<-pmax(strike-S,0)
  for (i in n:1) SnellZ[i]<-max(gS[i],R*SnellZ[i+1])
  tau.index<-dummy[SnellZ==gS][1]
  DiscPayoff[j]<-disc^((tau.index-1))*gS[tau.index]
}

print(c(mean(DiscPayoff), 1.96*sd(DiscPayoff)/sqrt(Nsim)))
```

Tabel 4 En simulationsalgoritme, man kunne tro, udregner den amerikanske put-optionspris. Men det gør den ikke.

overraskende og langt fra oplagt, at det overhovedet kan lade sig gøre. Til det formål vil jeg vise, hvordan man *ikke* skal gøre.

Der er andre måder end ligning (2) at karakterisere den amerikanske optionspris på. I den sammenhæng er *stoppetider* vigtige. Definitionen er (her) simpel: en stoppetid, τ, er en stokastisk variabel, der tager værdier i mængden $\{0, 1, \ldots, T\}$ for hvilken hændelsen $\{\tau \leq n\}$ er $\mathcal{F}_n$-målelig for ethvert n. Intuitivt betyder det, at på tid n ved vi, om vi er stoppet eller ej. Den amerikanske optionspris kan repræsenteres som løsningen til det optimale stoppetidsproblem

$$\pi^A(0) = \max_{\tau \in \mathcal{T}} \mathbb{E}^Q \left(e^{-r\tau} g(S(\tau)) \right), \tag{3}$$

hvor $\mathcal{T}$ er mængden af alle stoppetider. Og det lyder jo rimeligt nok; vi stopper, så vi ("i forventning" og diskonteret) får mest muligt ud af vores option. Det optimale tidspunkt at stoppe, τ^*, er

$$\tau^* = \min\{t \mid g(S(t) = V(t)\}, \tag{4}$$

hvor den såkaldte Snell-foldning V er defineret rekursivt baglæns ved $V(T) = g(S(T))$ og

$$V(t) = \max\{g(S(t)), e^{-r\Delta t}\mathbb{E}_t^Q(V(t+\Delta t))\}. \tag{5}$$

Igen: Det lyder rimeligt nok. Ligningen ovenfor er jo stort set (2), og vi kan vel ligeså godt stoppe første gang, det ikke kan blive bedre.

Man kan nu forsøge regne på (3)-(5) med denne algoritme:

1. Simuler N aktiekurs-stier hver med n skridt, $\{S_j(t_i)\}_{i=0}^n$, hvor $\Delta t = T/n$, $t_i = i\Delta t$ og j-indekset angiver sti-nummeret.
2. Udregn foldningen som
$$V_j(t_i) = \max\left\{g(S_j(t_i)), e^{-r\Delta t}V_j(t_{i+1})\right\}.$$
3. Brug $\tau_j^* = \min\{t_i \mid V_j(t_i) = g(S_j(t_i))\}$ som indfrielsestidspunkt for sti j.
4. Udregn gennemsnittet over mange stier (j) af $e^{-r\tau_j^*}g(S_j(\tau_j^*))$ og brug det som estimat for den amerikanske optionspris.

Algoritmen er implementeret som R-kode i Tabel 4. I Tabel 5 sammenlignes beregnede priser. Simulationsalgoritmen ovenfor giver en pris, der er omtrent dobbelt så stor som den korrekte amerikanske put-optionspris. Ups. Så hvad er der galt? Fejlen ligger i algoritmens 2. skridt, foldningsudregningen. Her er den betingede forventningsoperator fra ligning (5) forsvundet. Beregningen af foldningen bruger derfor fremtidig information, hvorved det fundne indfrielsestidspunkt matematisk set ikke er en stoppetid. Et par faktorer gør en fejlagtig implementation som denne svær at fange. For det første er den forkerte pris ikke *åbenlyst* forkert; om 6,74 eller ca. 13 er det rigtige tal, er ikke umiddelbart oplagt. Det betyder så på positiv-siden, at en pris udregnet med den slags fuld forudsigelighed er en ikke-triviel øvre grænse

Optionstype	Algoritme	Pris ($\pm$ konf'bånd)
Eu. put		6,46
Am. put	Tabel 1	6,74
Am. put (misforstået)	Tabel 4	13,0 $\pm 0,2$
Am. put	L & S (2001)-sim.	6,7 $\pm 0,2$

Tabel 5 Den amerikanske put-option forsøgt prisfastsat med algoritmen og parametrene fra Tabel 1 (korrekt), med algoritmen fra Tabel 4 (forkert) og med simulationsmetoden fra Longstaff & Schwartz (2001) (appoksimativt korrekt; der bruges 2.-gradspolynomier i regressionerne). Ved at tillægge eller fratrække tallet angivet med $\pm$ fås 95%-konfidensintervallet for prisestimaterne opnået ved simulation.

for værdien af den amerikanske option. I mere avancerede sammenhænge kan den korrekte optimale indfrielsesstrategi være meget vanskelig at beregne, men den øvre grænse nem. I kapitel 3 i [Rasmussen, Madsen & Poulsen (2011)] bruges det til at finde øvre grænser for konverteringsgevinster i realkreditmarkeder, og folk snedigere end os bruger ideen i såkaldte primal-dual-simulationer. For det andet er foldningsberegningen "lokaliseret", den involverer kun t_i og t_{i+1} så man ved første øjekast "ikke ser ret langt ind i fremtiden" — men altså alligevel nok til at tingene bryder sammen.

Lad os nu se på, hvad man så kan gøre. Vi beholder vores N simulerede aktiekurs-stier, zoomer ind på et tidspunkt t_i og antager i den baglæns rekursive tradition, at alt t_{i+1}-relevant allerede er beregnet. Markov-egenskaben fortæller os, at tidsværdien af optionen har den funktionelle form

$$\mathbb{E}^{Q}_{t_i}(\pi^A(t_{i+1})/R) =: f_{t_i}(S(t_i)).$$

Langs sti j kender vi det rette *argument*, tallet $S_j(t_i)$, problemet er, at vi ikke kender *funktionen* f_{t_i}. Det er en betinget forventet værdi, så vi kunne prøve at starte N nye simulerede stier fra $S_j(t_i)$. Men for hver af dem ville vi på tid t_{i+1} skulle starte nye stier osv. hele vejen ud til tid T. Vi ender med at have noget ala N^n stier (fx 10.000^{252}), og det jo ikke til at have med at gøre.[25] Men vi kan jo prøve med et gæt eller en approksimation til f_{t_i}'s form, fx kvadratisk

$$f_{t_i}(x) = a_0 + a_1 x + a_2 x^2,$$

hvor a'erne er ukendte koeffcienter, som vi vil estimere. (De budre egentlig have t_i'er på, men det er sent i artiklen ...) Vi opfatter nu næste periodes (diskonterede) faktisk realiserede optionsværdi $\pi^A_j(t_{i+1})/R$ (husk: det tal har vi beregnet, da vi arbejder rekursivt baglæns) som et *signal* om tidsværdien. Det gør vi for alle stier og kombinerer det med den postulerede funktionelle form af f_{t_i}, så vi får ligningerne

$$\pi^A_j(t_{i+1})/R = a_0 + a_1 S_j(t_i) + a_2 S^2_j(t_i) + \epsilon_j \text{ for } j = 1, \ldots, N,$$

hvor ϵ_j'erne er fejlled. Vi kan nu estimere a'erne ved lineær regression, her specifikt

$$\widehat{a} = \frac{1}{R}(X^\top X)^{-1} X^\top \pi^A(t_{i+1}),$$

[25] Sådanne "nested simulations" er ikke helt så fjollede, som jeg her får dem til at se ud. Men beregningsmæsssigt dyre er de altid.

hvor X er en matrix med indgange $X_{j,k} = S_j^{k-1}(t_i)$ for $j = 1, \dots, N$ og $k = 1, 2, 3$. Det er denne regression på tværs af stier, der sikrer, at fremtidige værdier ikke udnyttes på en "ulovlig" måde. For at forstå forskellen til det første simulationsforsøg kan man gøre sig følgende tankeeksperiment: I næstsidste periode har to simulerede stier begge aktiekursen 98. For den ene sti slutter aktiekursen i 102, for den anden i 97. I "snyde-algoritmen" indfries strike-100 put-optionen for den første sti (venter man får man 0; bedre at få 2 nu), men ikke for den anden (venter man får man 3 isf. 2 nu). Brugen af den estimerede funktionelle form af tidsværdien sikrer, at man for stier med samme aktiekurser træffer de samme indfrielsesbeslutninger.

Som tid-t_i optionsværdi(estimat)er bruger vi

$$\pi_j^A(t_i) = \max\left((K - S_j(t_i))^+, \sum_{k=1}^{3} X_{j,k}(t_i)\widehat{a}_k\right),$$

vi estimerer det kritiske niveau for indfrielse som $S^*(t_i) = \max\{x \leq K \mid K - x = f_{t_i}(x)\}$, og vi er klar til at fortsætte videre baglæns til tidspunkt t_{i-1}.

Der er et par ekstra julelege i Longstaff og Schwartz' fulde algoritme (fx at man kun bruger in-the-money-stier i regressionen), som gør koden for omfattende til at blive gengivet her, men den kan findes samme sted som artiklens øvrige programmer.

Sidste linje i Tabel 5 viser, at selv med den nok noget friske kvadratiske approksimation til optionens tidsværdi, så ligger den beregnede amerikanske put-optionspris ganske tæt på den (i binomialmodellen entydigt) sande værdi; 6,7 ± 0,2 ift. 6,74. En grund til dette er den forholdvis tilgivende natur af problemet: Vi skal finde det kritiske niveau, hvor vi skal indfri vores option. Det er netop der, hvor værdien af at indfri er den samme som af at holde optionen i live. Mindre fejltagelser er derfor næppe katastrofale.

Hvorfor amerikanske optioner burde blive kaldt Kierkegaard-optioner

Søren Kierkegaard skrev, at "livet leves forlæns, men forstås baglæns". Det beskriver præcis situationen for amerikanske optioner: Vi bestemmer den optimale indfrielsesstrategi (og samtidig prisen) rekursivt baglæns, men må som ejere af en amerikansk option følge strategien forlæns i tid uden at kunne se ind i fremtiden.

Litteratur

[Andersen (1999)] Leif Andersen. A simple approach to the pricing of Bermudan swaptions in the multifactor LIBOR market model. *Journal of Computational Finance*, 2:5–32, 1999.

[Broadie & Glasserman (1997)] Mark Broadie and Paul Glasserman. Pricing American-Style Securities by Simulation. *Journal of Economic Dynamics and Control*, 21:1323–1352, 1997.

[Carriere (1996)] J. F. Carriere. Valuation of the early-exercise price for derivative securities using simulations and splines. *Insurance: Mathematics and Economics*, 19:19–30, 1996.

[Eddelbuettel (2009)] Dirk Eddelbuettel. *Introduction to High-Performance Computing with R.* Præsentation ved The Institute of Statistical Mathematics Tachikawa, Tokyo, 2009. http://dirk.eddelbuettel.com/papers/ismNov2009introHPCwithR.pdf.

[Higham (2002)] Desmond J. Higham. Nine Ways to Implement the Binomial Method for Option Valuation in MATLAB. *SIAM Review*, 44:661–677, 2002.

[Lamberton & Lapeyre (1996)] Daniel Lamberton and Bernard Lapeyre. *Introduction to stochastic calculus applied to finance*. Chapman & Hall, 1996.

[Lando (1999)] David Lando. En hovedsætning i finansieringsteorien — og lidt om kreditobligationer. *FAMØS*, 13:9–20, 1999.

[Lando & Poulsen (2006)] David Lando and Rolf Poulsen. *Notes for Finance 1 (and More)*. Institut for Matematiske Fag, Københavns Universitet, 2006. http://www.math.ku.dk/~rolf/ifnotes.pdf.

[Longstaff & Schwartz (2001)] Francis A. Longstaff and Eduardo S. Schwartz. Valuing American Options by Simulation: A Simple Least-Squares Approach. *Review of Financial Studies*, 14:113–147, 2001.

[Pedersen, Weissensteiner & Poulsen (2011)] Anne Marie Boiden Pedersen, Alex Weissensteiner, and Rolf Poulsen. Financial planning for young households. Working paper, http://www.math.ku.dk/~rolf/MultistagePaper_260811.pdf, 2011.

[Rasmussen, Madsen & Poulsen (2011)] Kourosh Marjani Rasmussen, Claus Madsen, and Rolf Poulsen. *Realkreditrådgivning: Et studie af danskernes valg af realkreditlån og konverteringspraksis*. Boligøkonomisk Videncenter, 2011. http://www.bvc.dk/SiteCollectionDocuments/Analyser/Konverteringsanalyse.pdf.