

KUNGLIGA TEKNISKA HÖGSKOLAN
Matematiska institutionen
Examensarbete

*Värdering av callables: modellering och
implementering*

Författare:
Erik Sjöberg

Sammanfattning

I detta examensarbete studeras värdering av callables. Detta är en särskild typ av obligationer där möjligheten finns för utställaren att vid ett antal bestämda tidpunkter under löptiden lösa in obligationen i förtid. Tre olika modeller för den korta räntan; Hull och Whites modell, Black, Derman och Toys modell samt Black och Karsinsikis modell har anpassats för prissättning av callables. Modellerna har implementerats i Quantlab, en miljö för kvantitativ analys, där prissättning sker i realtid.

Abstract

This master's thesis explores pricing of callables. These are special bonds that allow the issuer at a number of certain times to buy back the bond. Three different models for the short-term interest rate; Hull and Whites model, Black, Derman and Toys model as well as Black and Karsinsikis model have been adjusted for pricing callables. The models have been implemented in Quantlab, a program for quantitative analysis, and pricing is done according to real-time data.

Innehållsförteckning

1	Introduktionskapitel.....	9
1.1	Bakgrund.....	9
1.2	Problemformulering.....	9
1.3	Syfte.....	10
1.4	Avgränsningar.....	10
1.5	Metod och rapportdisposition.....	10
2	Teori.....	11
2.1	Inledande modellteori.....	11
2.2	Ränteträd.....	13
2.3	Callables.....	15
2.4	Hull och Whites modell.....	15
2.5	Black, Derman och Toys modell.....	16
2.6	Black och Karsinskis modell.....	16
2.7	Hull och Whites algoritm för konstruktion av trinomialträd.....	16
2.7.1	Steg 1.....	17
2.7.2	Steg 2.....	18
2.7.3	Anpassning till andra räntemodeller.....	20
2.8	Räntornas antagna fördelning.....	21
2.9	Kalibrering.....	22
2.10	Tidsberoende volatilitetsparametrar.....	22
3	Modellanpassning.....	25
3.1	Val av tidpunkter för nodernas placering.....	25
3.2	Storlek på räntans förändring mellan två intelligande noder.....	25
3.3	Val av metoder i trädet.....	26
3.4	Anpassning till rådande räntestruktur.....	26
3.5	Kalibrering.....	26
4	Implementering och resultat.....	29
4.1	Implementeringsmiljö.....	29
4.2	Trädrepresentation.....	29
4.3	Funktionalitetskrav.....	30
4.4	Programmering i Quantlab.....	30
4.5	Användande i Quantlab.....	32
4.6	Resultat i Quantlab.....	34
4.7	Jämförelser mellan de olika modellernas resultat.....	36
4.8	Exempel på ytterligare användning.....	36
5	Slutord och förslag till framtida studier.....	39
	Referenser.....	41
	Bilaga 1.....	43
	Bilaga 2.....	45

Förord

Detta examensarbete utfördes under perioden oktober 2002 – mars 2003 i samarbete med Algorithmica Research AB. Under arbetet har jag fått hjälp och stöd och vill därför uttrycka min tacksamhet till följande personer som hjälpt mig att göra detta arbete möjligt:

Magnus Nyström, min handledare på Algorithmica Research AB, för förslag på område att studera samt hjälp under arbetet.

Harald Lang, min handledare på KTH, för hjälp och stöd.

Övrig personal på Algorithmica Research AB för hjälp med såväl det teoretiska arbetet som med implementering och programmering.

1 Introduktionskapitel

Introduktionskapitlet syftar till att ge läsaren en introduktion till denna rapport och det utförda examensarbetet. Inledningsvis beskrivs bakgrunden för att skapa förståelse för varför det är av vikt att studera det valda ämnet. Därefter följer en problembeskrivning som översiktligt beskriver de områden och problem som studerats. Examensarbetets syfte och de avgränsningar som gjorts presenteras sedan. Kapitlet avslutas med ett avsnitt som behandlar examensarbetets metod samt rapportens fortsatta disposition.

1.1 Bakgrund

På de finansiella marknaderna pågår en betydande handel med olika typer av räntederivat. Dessa är instrument som på något sätt beror på framtida räntenivåer. Under de två senaste decennierna har volymen av handel med dessa derivat ökat mycket snabbt. Värdering av dessa är en viktig del för aktörer på de finansiella marknaderna. Obligationer med möjlighet till inlösen under obligationens löptid, s.k. callables, är en typ av instrument som handlas på dessa marknader. Särskilt populära har den typ av callables som ger möjlighet till inlösen vid ett antal fixa, i förväg bestämda, tidpunkter kommit att bli. Prissättning av dessa instrument är därför av stor vikt.

Algorithmica Research AB säljer mjukvara till aktörer på de finansiella marknaderna och deras produkter används som stöd för kvantitativ analys och handel på olika räntemarknader. Företaget ser ett behov av att kunna erbjuda sina kunder möjlighet att värdera olika typer av ränteinstrument, däribland callables. I nuläget finns möjlighet för användaren att genom egna implementeringar använda modeller i den miljö som Algorithmicas mjukvara ger. Företaget vill dock i framtiden låta färdiga implementeringar av olika modeller, för värdering av olika typer av instrument, ingå som en del i deras produkter.

Under de senaste 15 åren har en mängd olika modeller för den s.k. korta räntan konstruerats och presenterats. Tre av de mest använda och populära modellerna är Hull och Whites modell, Black, Derman och Toys modell samt Black och Karsinskis modell. Samtliga dessa modeller kan användas för modellering av den korta räntan och lämpar sig särskilt för representation i ränteträd.

1.2 Problemformulering

De modeller som studeras i detta examensarbete tillhör inte de allra mest avancerade men för att kunna vidareutveckla och implementera dem på ett effektivt sätt krävs förståelse för bakomliggande teori samt kunskap om hur modellerna fungerar i deras ”grundutförande”.

För att kunna använda modellerna för värdering av callables måste en anpassning göras. Vid representation av modellerna i ränteträd måste deras flexibilitet utökas.

Implementering av modellerna utgör den största delen av arbetet vid användning av räntemodeller. Det är av stor vikt att modellerna kan användas i en lämplig miljö samt att användaren kan styra de ingående parametrarna och de data som modellen baserar sig på.

1.3 Syfte

Syftet med detta examensarbete är att komma fram till hur tre modeller för den korta räntan kan anpassas för att kunna användas vid värdering av callables. Avsikten är vidare att praktiskt implementera modellerna i Algorithmicas programmeringsmiljö Quantlab där värdering ska kunna ske i realtid.

1.4 Avgränsningar

I implementeringsfasen av detta examensarbete kommer inte kalibreringar att utföras. Denna process kommer dock att beskrivas och förslag på hur den ska ske kommer att presenteras.

1.5 Metod och rapportdisposition

Arbetet med detta examensarbete kan i grova drag delas in i tre delar. Den första delen består av teoristudier, den andra av modellanpassning och den tredje delen av implementering av modellerna. Detta motsvarar också den kronologiska ordning enligt vilken examensarbetet fortskridit. På detta sätt är också den fortsatta rapporten upplagd, först följer ett avsnitt där den grundläggande teorin presenteras. Sedan presenteras de modellanpassningar som gjorts och därefter redovisas hur implementeringen skett och resultatet av denna. Rapporten avslutas med ett kapitel med slutord och rekommendationer för fortsatta studier.

2 Teori

I detta kapitel beskrivs de viktigaste delarna av den teori som examensarbetet grundar sig på. Viss förkunskap krävs, men förhoppningen är att kapitlet ska vara tillräckligt för att ge läsaren den kunskap som är nödvändig för att kunna få en god förståelse för det genomförda arbetet.

2.1 Inledande modellteori

De allra enklaste modellerna för värdering av räntederivat baseras på att räntorna är lognormalfördelade. Derivat värderas sedan med modeller som baseras på Black och Scholes antaganden. En av de mest populära modellerna av denna typ är Blacks modell som är en generalisering av Black och Scholes modell för optionsvärdering. Dessa modeller används i stor utsträckning av marknadens aktörer och är i många fall fullt tillräckliga för värdering av derivat av europeisk typ. De ger dock inte en beskrivning av räntornas stokastiska natur. Modellerna kan därför inte användas för värdering av räntederivat där värdet beror på aktörernas agerande under derivatets löptid. Derivat där så är fallet är exempelvis amerikanska ränteoptioner av olika typ och då i synnerhet det instrument som ska värderas i detta examensarbete, callables. Detta instrument presenteras senare i denna rapport.

För värdering av derivat av den typ som inte kan modelleras med hjälp av de enkla metoderna, t.ex. Blacks modell, används modeller som beskriver hur räntorna utvecklas över tiden, s.k. terminstrukturmodeller. Dessa modeller kan översiktligt delas in i två typer. Den första typen av modeller beskriver framtida forwardräntor. Exempel på populära modeller av denna typ är Heath, Jarrow och Mortons modell samt Brace, Gatarek och Musielas modell.

Den andra typen av modeller beskriver utvecklingen av den ränta som gäller under nästkommande korta tidsintervall – den korta räntan. Ofta är denna typ av modeller svårare att förstå än modeller för forwardräntor. Fördelen är dock att dessa modeller kan implementeras med rekombinerande träd liknande de träd som används vid modellering av aktier. Detta gör att modellerna blir snabba och användbara för värdering av alla typer av räntederivat. Det är också denna typ av modeller som används i detta examensarbete.

De i detta examensarbete studerade modellerna för den korta räntan är Markovmodeller. Utvecklingen av den framtida korta räntan beror därför inte på hur den tidigare utvecklas.

Mean reversion

Med mean reversion (“medelåtergång”) menas att processen som beskriver räntorna tar hänsyn till att extrema värden är osannolika. Om de modellerade räntorna avviker stort från de räntor som fås från den rådande räntestrukturen kommer de därför att återgå mot denna struktur enligt ett visst samband. Detta är åskådliggjort i figur 1.

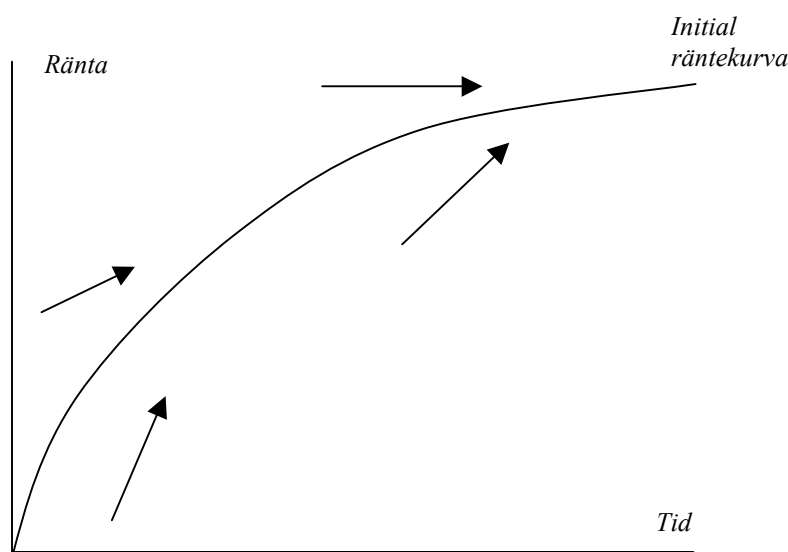
Den första modellen med denna egenskap var Vasiceks modell som presenterades 1977. Processen för den korta räntan antas då utvecklas enligt ekvation 1:

$$dr = a(b - r)dt + \sigma dz \quad \text{Ekvation 1}$$

I ekvation 1 betecknar dr differentialen av räntan, vilket innebär den momentana förändringen av räntan. Parametern dt är tiddifferentialen och dz är en wienerprocess.

Mean reversion kommer i modellen att ske mot nivån b . För korta räntor som avviker från denna nivå kommer en anpassning mot b att ske. Variabeln a betecknar takten i vilken återgång mot denna nivå på korta räntan sker. Det är denna variabel som avgör storleken på effekten av modellens mean reversion. Variabeln σ motsvarar standardavvikelsen för de årliga förändringarna av den korta räntan.

Modellerna som används i detta arbete har alla olika sätt att använda mean reversion i modelleringen.



Figur 1. Mean reversion

Ickearbitrage modeller

Räntemodeller som behandlar stokastiska räntor kan vara av olika typ. En tydlig distinktion görs mellan jämviktsmodeller och modeller som bygger på antaganden om icke-arbitrage.

Jämviktsmodeller är modeller som bygger på antaganden om ekonomiska variabler. Dessa antaganden leder fram till en process som beskriver räntorna. Slutligen studeras hur detta påverkar priser på obligationer och optioner. Vasiceks modell som beskrivs ovan är av denna typ.

Det främsta problemet med jämviktsmodeller är att de i vissa fall ger en modell som inte överensstämmer med rådande marknadsdata. Modeller som bygger på antaganden om icke-arbitrage utgår istället från den rådande räntestrukturen. Dessa modeller är konstruerade så att de istället ska vara helt konsistenta med rådande räntor. En stor och tydlig skillnad mellan en jämviktsmodell och en icke-arbitrage modell är att vid användande av jämviktsmodeller är räntestrukturen ett resultat av modellen medan den vid användandet av icke-arbitrage modell antas vara känd.

Första gången som en modell av icke-arbitrage typ presenterades var 1986, då Ho och Lee presenterade sin modell för de korta räntorna. Ursprungligen utvecklades denna modell för representation i binomialträd men visade sig också ha vissa analytiska användningsområden.

Detta gäller även för de flesta modeller av denna typ. I Ho och Lees modell följer den korta räntan processen som beskrivs i ekvation 2:

$$dr = \theta(t)dt + \sigma dz \quad \text{Ekvation 2}$$

I ekvationen betecknar dr differentialen av räntan och dt tidsdifferentialen. Den sista differentialen, dz , utgör en wienerprocess. Variabeln σ motsvarar standardavvikelsen för de årliga förändringarna av den korta räntan.

Variabeln $\theta(t)$ är en funktion som väljs så att modellen följer den rådande räntestrukturen. Det är denna variabel som gör att modellen blir av ickearbitrage typ, d.v.s. modellen kommer att prissätta befintliga obligationer efter den rådande räntestrukturen och därmed till marknadspris.

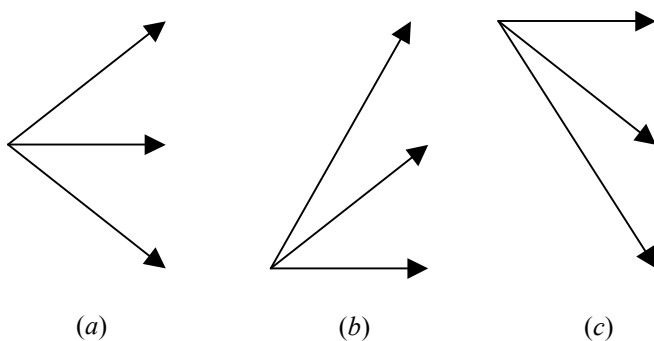
De tre studerade modellerna i detta examensarbete är av ickearbitrage typ och de rådande räntorna används vid anpassning av modellen. Dessa kan ses som vidareutvecklingar av Ho och Lees modell.

2.2 Ränträd

Rekombinerande träd är ett mycket kraftfullt och illustrativt sätt att implementera modeller för den korta räntan. Det slutgiltiga resultatet av modelleringen utgörs då av ett träd som beskriver de korta räntorna. Trädet kan sedan användas för värdering av räntederivat.

Ett ränträd är en tidsdiskret representation som visar den stokastiska process som ränteutvecklingen utgör. Vid konstruktion av ett ränträd delas den aktuella tidsperioden, i det enklaste fallet, in i ett antal tidsteg av längden Δt . Skillnaden mot ett träd som beskriver exempelvis en stokastisk process för en aktie är den använda räntan för ett visst tidsteg. När man använder ett aktieträd antas oftast räntan vara densamma för alla tidsteg. I ett ränträd är det just denna ränta som skiljer sig åt. Varje nod i trädet motsvarar en viss ränta och räntorna skiljer sig åt mellan de olika noderna.

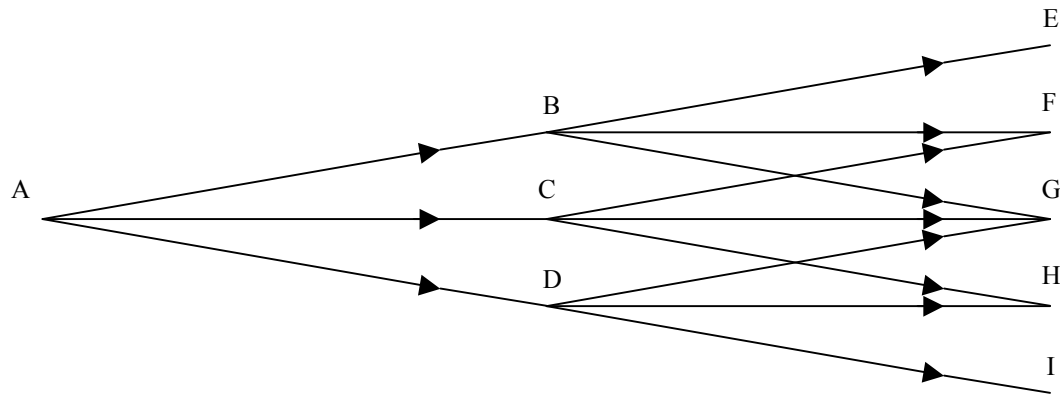
För räntemodeller används ofta trinomialträd istället för binomialträd som är vanligt förekommande vid åskådliggörande av andra processer, t.ex. aktiepriser. Anledningen till detta är att modellen då får ytterligare en grad av frihet och därmed kan innefatta egenskaper som mean reversion. Den ytterligare frihetsgraden vid användande av trinomialträd visas i figur 2 nedan. Stegen enligt figur 2b och figur 2c är särskilt användbara då man uppnått extremvärden och vill att trädet ska söka sig mot ett föreddefinierat värde, s.k. mean reversion (se ovan).



Figur 2. Möjliga metoder i ett trinomialträd

I trädet kommer olika sannolikheter att gälla för de olika möjliga stegen. I figur 2a ovan motsvaras detta av en sannolikhet för att ett steg uppåt ska ske, p_u , (pil uppåt), en sannolikhet för att en vågrät förändring ska ske, p_m , (vågrät pil) och en sannolikhet för att ett steg neråt ska ske, p_d (pil neråt).

Den värderingsprocedur som används vid modellering med trinomialträd är generell och motsvarar till stor del värdering av derivat i ett binomialträd. Resonemanget för hur detta sker i ett trinomialträd utgår från ränteträdet i figur 3 nedan.



Figur 3. Ränteträd

Ett räntederivat är ett instrument vars avkastning beror på räntan vid en viss tidpunkt. I figur 3 kan ett sådant derivat utgöras av en utbetalning vid tiden $2\Delta t$, dvs. vid trädets slut, som på något sätt beror på de under löptiden rådande räntorna. För att värdera detta derivat beräknas värdet för derivatet för de olika möjliga räntorna enligt trädstrukturen vid tiden $2\Delta t$. De olika räntorna utgörs alltså av räntorna i noderna E-I i figur 3. Beräkningar sker sedan så att var och en av noderna motsvarar ett värde på derivatet vid tiden $2\Delta t$.

Eftersom de olika sannolikheterna för samtliga steg i trädet är kända används sedan en iterativ metod för att få fram priset som ska gälla vid noden A, dvs. värdet av detta derivat vid tidpunkten noll. Värdet på derivatet i noderna B-D beräknas genom att använda sig av de kända värdena i noderna E-I samt sannolikheterna för att dessa noder ska nås. Exempelvis kommer värdet i nod B, V_B , att bli:

$$V_B = (p_{B-E}V_E + p_{B-F}V_F + p_{B-G}V_G)e^{-r_B\Delta t} \quad \text{Ekvation 3}$$

I ekvationen motsvarar V_E , V_F , V_G de olika värdena i noderna E-G och p_{B-E} , p_{B-F} , p_{B-G} sannolikheterna för de olika möjliga stegen från nod B. Räntan enligt vilken värdet ska diskonteras är den ränta som gäller i nod B, r_B . På motsvarande sätt beräknas sedan värdena i nod E och G och slutligen erhålles priset i nod A, V_A :

$$V_A = (p_{A-B}V_B + p_{A-C}V_C + p_{A-D}V_D)e^{-r_A\Delta t} \quad \text{Ekvation 4}$$

Denna metod är generell och kan användas i träd med olika antal steg samt, som visas senare i denna rapport, för träd med steg av varierande längd.

2.3 Callables

Callable är en kortare benämning för *callable bond* som är en obligation med möjlighet för utställaren att lösa in obligationen under löptidens gång. Möjligheten till tidig inlösen medför att det finns en ”inbyggd” option i instrumentet. Köparen av en sådan obligation kan sägas ha sålt en köpoption till utställaren av obligationen. Priset för denna köpoption utgörs av skillnaden mellan priset på en obligation utan dessa möjligheter till inlösen och motsvarande callable. Med bakgrund av detta kommer priset för en callable vara lägre än priset för motsvarande obligation utan möjlighet för utställaren att utnyttja tidig inlösen.

Med callable avses ofta instrument av typen ”bermudian callable”. Utställaren av en sådan obligation har möjlighet att vid vissa förutbestämda tidpunkter, ofta i samband med utdelningar på obligationen, lösa in obligationen till i förväg fastställda priser. Callables av andra typer, såsom amerikanska där inlösen kan ske när som helst under löptiden, förekommer också men instrument av typen ”bermudian callable” har kommit att bli så dominerande att det är detta instrument som avses när man använder benämningen callable. Det är också denna typ av instrument som studeras i detta examensarbete.

Värdering av callables i räntetrad följer till stor del den beskrivna arbetsgången i kapitel 2.2 ovan.

2.4 Hull och Whites modell

Hull och Whites modell är en vidareutveckling av Vasiceks modell från 1977. Den modellen var av jämviktstyp och Hull och White ville med sin modell ta hänsyn till den rådande räntestrukturen. Modellen kan också ses som Ho och Lees modell med mean reversion egenskaper. Hull och Whites modell presenterades för första gången i en artikel 1990.

Matematiskt förändras räntan enligt Hull och Whites modell på följande vis:

$$dr = (\theta(t) - a(t)r)dt + \sigma(t)dz \quad \text{Ekvation 5}$$

I ekvation 5 är dr differentialen av räntan, vilket innebär den momentana förändringen av räntan. På samma vis är dt tidsdifferentialen. Den sista differentialen, dz , betecknar en wienerprocess.

Modellen antar normalfördelade korta räntor. Variabeln a betecknar takten i vilken återgång mot den rådande räntestrukturen sker. Det är denna variabel som avgör storleken på effekten av modellens mean reversion. Variabeln σ motsvarar standardavvikelsen för de årliga förändringarna av den korta räntan. I sin ursprungliga utformning av modellen antas parametrarna a och σ vara konstanta. Båda dessa parametrar kan dock modelleras som funktioner av tiden. Resonemang runt detta förs senare i denna rapport.

Variabeln $\theta(t)$ är en funktion som väljs så att modellen följer den rådande räntestrukturen. Det är denna variabel som gör att modellen blir av ickearbitrage typ, d.v.s. modellen kommer att prissätta befintliga obligationer efter den rådande räntestrukturen och därmed till marknadspris.

2.5 Black, Derman och Toys modell

Denna modell skiljer sig från Hull och Whites modell framför allt i två avseenden. Den första skillnaden, och den som röner störst intresse i detta examensarbete, är att räntorna i denna modell antas vara lognormalfördelade. Den andra aspekten som skiljer sig är att Black, Derman och Toy redan från början antog att volatiliteten och mean reversion parametern utgörs av funktioner av tiden. Modellen härstammar från 1990. Processen för den korta räntan beskrivs av ekvation 6.

$$d \ln r = \left[\theta(t) + \frac{\sigma'(t)}{\sigma(t)} \ln(r) \right] dt + \sigma(t) dz \quad \text{Ekvation 6}$$

Till skillnad från de modeller som presenteras tidigare i denna rapport visar ekvationen inte differentialen för den korta räntan utan istället differentialen av den naturliga logaritmen för den korta räntan. Denna betecknas $d \ln r$. Det är även logaritmen av räntan som tillsammans med mean reversion parametern bestämmer återgångens storlek. I fallet med Black, Derman och Toys modell bestäms mean reversion parametern som kvoten mellan volatilitetsfunktionens derivata och volatiliteten själv. Funktionen $\theta(t)$ fungerar på motsvarande sätt som i Hull och Whites modell för att anpassa trädet till rådande räntestruktur. I detta fall anpassas dock logaritmen av den korta räntan istället för räntan som tidigare. Differentialerna dt och dz har samma betydelse som i Hull och Whites modell. Volatilitetsparametrarnas betydelse är dock något annorlunda. Detta behandlas i kapitel 2.8 nedan. Resonemang kring parametrarnas tidsberoende förs i kapitel 2.10.

Modellen utarbetades från början för implementering i binomialträd enligt en särskild algoritm. Senare har dock Hull och White visat hur deras trädkonstruktionsalgoritm för trinomialträd kan anpassas till Black, Derman och Toys modell. Detta diskuteras vidare senare i denna rapport.

2.6 Black och Karsinskis modell

Black och Karsinski presenterade 1991 en modell som kan ses som en generalisering av Black, Derman och Toys modell. I Black, Derman och Toys modell bestäms mean reversion parametern som en kvot mellan volatiliteten och dess derivata. Black och Karsinski lät istället denna parameter vara en godtycklig funktion av tiden. I övrigt är modellen identisk med Black, Derman och Toys modell. Processen för den korta räntan beskrivs enligt denna modell av ekvation 7:

$$d \ln r = [\theta(t) - a(t) \ln(r)] dt + \sigma(t) dz \quad \text{Ekvation 7}$$

Parametrarna i Black och Karsinskis modell är desamma som i Black, Derman och Toys modell med skillnad endast avseende mean reversion parametern. Parametern $a(t)$ utgör i modellen denna, mer generella, parameter. Redogörelsen för tidsberoende parametrar, räntans fördelning och trädkonstruktionsalgoritm är gemensamma för de båda modellerna och presenteras senare i rapporten.

2.7 Hull och Whites algoritm för konstruktion av trinomialträd

Hull och White presenterade 1994 en tvåstegsalgoritm för att implementera deras modell i ett trinomialträd. Metoden har sedan utvecklats till att bli mer generell och fungerar för flera olika

räntemodeller, däribland de två övriga modeller som studeras i detta examensarbete. Algoritmen är indelad i två steg. Det första steget beskriver hur trädstrukturen konstrueras för en approximativ ränta. Därefter konverteras i steg två strukturen för att vara korrekt även för den verkliga räntan. I detta kapitel presenteras algoritmen i sitt standardutförande för att i kapitel 3 utvecklas för att kunna värdera callables. Till en början görs en genomgång av algoritmen så som den används vid modellering med Hull och Whites modell. Därefter diskuteras de generaliseringar och ändringar som krävs för att kunna använda algoritmen även för andra modeller.

2.7.1 Steg 1

Vid användande av träd delas tiden, i det grundläggande fallet, in i ett antal tidsperioder med konstant längd Δt . Ett grundantagande i denna procedur är att den ränta, R , som gäller under denna period följer samma process som räntan r enligt Hull och Whites modell (ekvation 5). Processen för R kan då skrivas enligt ekvation 8 nedan.

$$dR = (\theta(t) - aR)dt + \sigma dz \quad \text{Ekvation 8}$$

Första steget består nu i att konstruera ett träd för en variabel R^* som från början är noll och följer processen

$$dR^* = -aR^* dt + \sigma dz \quad \text{Ekvation 9}$$

I detta första steg tas alltså inte hänsyn till funktionen $\theta(t)$ och alltså sker ingen anpassning till den rådande räntestrukturen och trädet uppfyller ännu så länge inte kraven för att betraktas som en icke arbitrage modell. Under en period kommer räntan att förändras enligt trädet. Figur 2 visar i vilka riktningar räntan kan förändras. Förändring som i trädet motsvaras av ett steg uppåt eller neråt betecknas ΔR . För att minimera felet vid användande av modellen väljs

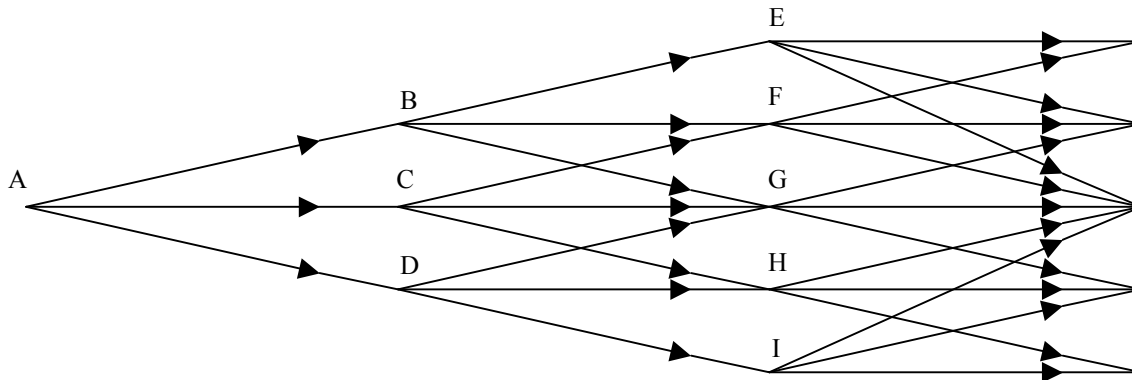
$$\Delta R = \sigma \sqrt{3\Delta t} \quad \text{Ekvation 10}$$

Målet med detta första steg är att konstruera ett träd som liknar det träd som illustreras i figur 4 nedan. För att kunna göra detta måste man komma fram till vilken typ av de möjliga stegen i figur 2 som ska användas i konstruktionen av trädet. Det är detta som kommer att avgöra hur trädet kommer att se ut. När väl detta är gjort måste de olika sannolikheterna i stegen beräknas.

För att kunna få fram vilken av metoderna i figur 2 som ska användas ser man på de olika noderna. En nod betecknas med två koordinater, i och j . Den första koordinaten, i , betecknar hur många steg i horisontell riktning som skett vilket alltså visar hur många tidsperioder som förflutit fram till denna nod. Koordinaten j visar hur många steg i vertikal riktning som skett fram till den aktuella noden och kan vara positiv eller negativ. Den visar sålunda hur många steg uppåt eller neråt från ursprungsräntan som tagits. En nod betecknas alltså av (i, j) där $t = i\Delta t$ och $R^* = j\Delta R$. Den metod som ska användas i en viss nod måste leda till att sannolikheterna för de tre olika alternativa stegen är positiva. Den metod som oftast ska användas är ”standardmetoden” i figur 2a.

När variabeln a i ekvation 5 ovan är positiv sker mean reversion. Detta innebär att metoden enligt figur 2b ska användas för starkt negativa värden på j medan metoden enligt figur 2c ska användas för stora positiva värden på j . Detta ses också tydligt i figur 4 där ett tillräckligt

positivt värde på j uppnått vid nod E och ett tillräckligt negativt värde uppnått vid nod I. De värden som utgör gräns för vilken ett byte av metod ska ske betecknas j_{max} respektive j_{min} . I grundutformningen av Hull och Whites algoritim väljs j_{max} till det minsta heltal som överstiger $0.184/(a\Delta t)$ och j_{min} väljs till $-j_{max}$. Det kan då visas att positiva sannolikheter för de olika stegen alltid uppnås.



Figur 4. Trinomialträd för Hull och Whites modell, steg 1

Sannolikheten för det högsta möjliga steget betecknas p_u , för mittsteget p_m och sannolikheten för steget som ger den mest neråtgående utvecklingen betecknas p_d . Sannolikheterna väljs så att de motsvarar den väntade förändringen av R^* och dess varians under följande tidsperiod, Δt . Vidare måste sannolikheternas summa, som vanligt, uppgå till ett. Dessa tre förhållanden leder till tre ekvationer som måste vara uppfyllda för sannolikheterna. Tre obekanta och tre ekvationer resulterar i ett lösbart system och de olika sannolikheterna kan beräknas. Detta sker för de olika typerna av steg enligt figur 2. Beräkningar och resultat av dessa redovisas i bilaga 1.

Sannolikheterna beror, förutom på vilken typ av metod som används, endast på koordinaten j . Detta innebär att sannolikheterna i nod B och nod F i figur 4 ovan är desamma. Dessutom är trädet symmetriskt, vilket innebär att sannolikheterna i nod D är en spegelbild av sannolikheterna i nod B (sannolikheten p_u i nod D är samma som p_d i nod B och vice versa).

När man gått igenom de olika delarna i steg ett är resultatet ett träd liknande det i figur 4 med kända sannolikheter för de olika möjliga stegen.

2.7.2 Steg 2

I detta steg ska trädet anpassas efter de gällande räntorna. Trädet som konstruerades i steg 1 som gäller för R^* ska nu konverteras för att gälla för R , den ränta som gäller under en viss tidsperiod, Δt . Metoden för att göra detta är att ersätta noderna i det träd som steg 1 resulterade i så att den initiala räntestrukturen efterföljs. För att få en exakt överensstämmelse med rådande räntestruktur används en iterativ metod. Denna metod beskrivs nedan.

För att kunna använda den iterativa metoden krävs två hjälpvariabler, $Q_{i,j}$ och α_i . Den första av dessa variabler betecknar nuvärdet av en säkerhet som ger utbetalning av en valutaenhet om nod (i,j) nås och noll annars, det s.k. Arrow-Debreu priset. Den andra hjälpvariabeln, α_i , definieras som $\alpha(i\Delta t)$, värdet av R vid tiden $i\Delta t$ på R -trädet minus det motsvarande värdet för R^* vid tiden $i\Delta t$ på R^* -trädet. Denna variabel visar alltså avvikelsen mellan räntan i det slutgiltiga trädet och räntan som modellerats i R^* -trädet. De båda hjälpvariablerna kan väljas så att den initiala räntestrukturen exakt efterliknas.

Det första momentet består nu i att komma fram till $Q_{0,0}$ och α_0 . Eftersom Q_0 ska motsvara nuvärdet av en säkerhet som ger 1 valutaenhet vid tidpunkt noll sätts denna variabel till 1. Värdet av α_0 väljs så att en nollkupongs obligation med inlösen vid tiden Δt ges rätt pris. Förutsatt att Δt är densamma som periodlängden för räntan väljs denna parameter därför till den gällande enperiodsräntan. Räntan för noden $(0,0)$ i R^* -trädet förändras nu till att motsvara räntan som motsvaras av α_0 .

Nästa moment syftar till att bestämma $Q_{1,1}$, $Q_{1,0}$, $Q_{1,-1}$ samt α_1 . Q -värdena bestäms genom att använda sannolikheterna i R -trädet. Sannolikheten för att noden $(1,1)$ ska nås är p_u för noden $(0,0)$ (motsvarande nod A i figur 4 ovan). Nuvärdet av $Q_{1,1}$ erhålles då genom att diskontera sannolikheten för att denna nod nås. På motsvarande sätt beräknas nuvärdena av $Q_{1,0}$ och $Q_{1,-1}$. När värdena för Q väl är bestämda kvarstår att bestämma värdet för α_1 . Detta väljs så att rätt pris ges för en nollkupong obligation med inlösen vid tiden $2\Delta t$. Priset för denna obligation vid nod B i figur 4 ovan är:

$$P_B = e^{-(\alpha_1 + \Delta R) \times \Delta t} \quad \text{Ekvation 11}$$

På motsvarande sätt är priset för obligationen vid nod C:

$$P_C = e^{-\alpha_1 \times \Delta t} \quad \text{Ekvation 12}$$

Vid nod D är priset:

$$P_D = e^{-(\alpha_1 - \Delta R) \times \Delta t} \quad \text{Ekvation 13}$$

Priset vid den första noden, A, blir då:

$$P_A = Q_{1,1}P_B + Q_{1,0}P_C + Q_{1,-1}P_D \quad \text{Ekvation 14}$$

Det verkliga priset, P , för denna obligation beräknas ur marknadsdata för räntan och ekvation 14 sätts lika med detta pris:

$$P = Q_{1,1}P_B + Q_{1,0}P_C + Q_{1,-1}P_D \quad \text{Ekvation 15}$$

Ur ekvation 15 löses sedan värdet på α_1 och räntorna i noderna B, C och D anpassas därefter. I fallet med normalfördelade räntor som i Hull och Whites modell är detta steg enkelt och sker analytiskt. Ekvation 16 nedan ger det analytiska uttrycket för α_1 .

$$\alpha_1 = \ln \left[\frac{Q_{1,1}e^{-\Delta R^* \Delta t} + Q_{1,0} + Q_{1,-1}e^{\Delta R^* \Delta t}}{P} \right] / \Delta t \quad \text{Ekvation 16}$$

Metoden fortsätter sedan iterativt för att bestämma värden på $Q_{2,j}$ för samtliga j . När detta är gjort fastställs α_2 och trädet anpassas efter detta. Vid beräkningarna av värdena på Q kan man med fördel använda sig av resultaten från beräkningarna av tidigare Q . Som exempel kan nämnas beräkningen av $Q_{2,1}$. Detta är nuvärdet av den säkerhet som ger en valutaenhet i utdelning om nod F i figur 4 nås och noll annars. Denna nod kan endast nås via nod B eller C och räntorna i dessa noder är kända enligt tidigare beräkningar. Sannolikheterna för stegen B-F (p_m i nod B) och C-F (p_u i nod C) är också kända och detta innebär att värdet av säkerheten i nod B, P_B , och i nod C, P_C kan beräknas enligt nedan:

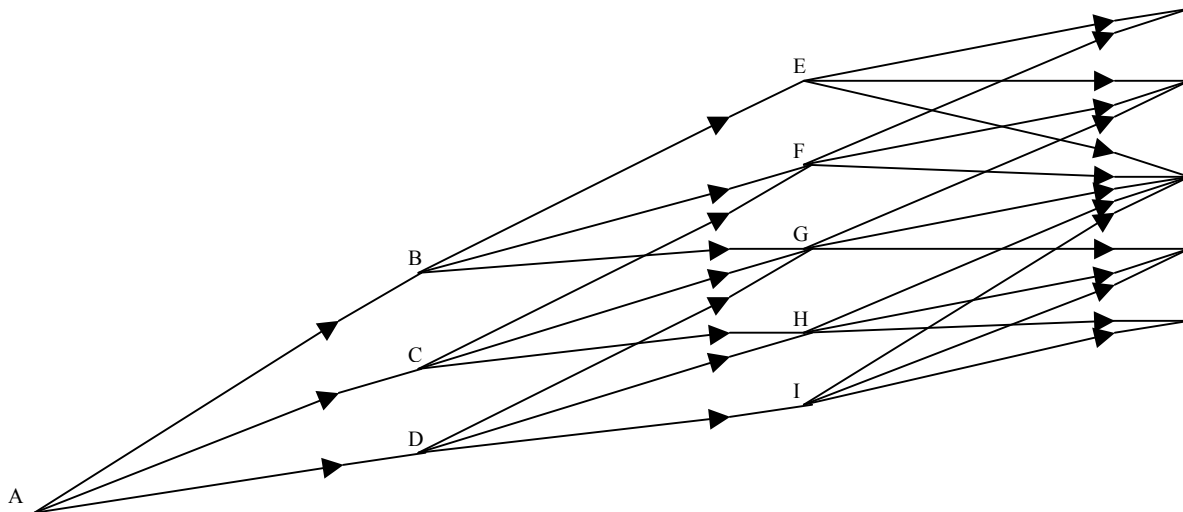
$$P_B = p_{B-F} e^{-(\alpha_1 + \Delta R)^* \Delta t} \quad \text{Ekvation 17}$$

$$P_C = p_{C-F} e^{-\alpha_1^* \Delta t} \quad \text{Ekvation 18}$$

Variabeln $Q_{2,1}$ sätts till P_B multiplicerat med nuvärdet av en valutaenhet som utbetalas vid nod B plus P_C multiplicerat med nuvärdet av en valutaenhet som utbetalas vid nod C. Detta ger:

$$Q_{2,1} = V_B \times Q_{1,1} + V_C \times Q_{1,0} \quad \text{Ekvation 19}$$

På motsvarande sätt beräknas resterande värden på $Q_{2,j}$ och α_2 kan därefter beräknas. Efter det kan sedan alla värden för $Q_{3,j}$ beräknas och därefter α_3 . På samma sätt fortsätter beräkningarna till ett träd av önskad storlek erhållits. Ett tänkbart utseende på ett sådant träd illustreras i figur 5. I figuren illustreras hur modellen anpassar sig efter en räntekurva med avtagande lutning.



Figur 5. Trinomialträd för Hull och Whites modell, steg 2

2.7.3 Anpassning till andra räntemodeller

Som nämnts tidigare kan den presenterade algoritmen användas för flera olika modeller. Så är dock inte fallet med Black, Derman och Toys modell samt Black och Karsinskis modell. I dessa modeller antas räntorna istället vara lognormalfördelade och en anpassning är därför bland annat nödvändig i steg 2.

Algoritmen kan användas för alla modeller som har en form enligt ekvation 20.

$$df(r) = [\theta(t) - af(r)]dt + \sigma dz \quad \text{Ekvation 20}$$

Här motsvarar $f(r)$ den funktion som man antar för räntorna i modellen följer. I fallet med Hull och Whites modell som presenteras ovan, där räntorna antas vara normalfördelade, är sambandet enklast möjliga: $f(r) = r$. För de andra två studerade modellerna antas räntorna vara lognormalfördelade och denna funktion är då istället $f(r) = \ln r$. Vid konstruktion av trädets för dessa modeller skapas trädets för $x = f(r) = \ln r$. Analogt med konstruktionen för Hull och Whites modell sker konstruktionen i två steg. I det första skapas ett träd för x^* som initialt är 0 och för vilken $\theta(t)$ är 0. Vid modelleringen gäller samma ekvationer som för Hull och Whites modell med x^* och x istället för R^* och R . Detta gäller även formlerna för sannolikheterna i trädets som redovisas i bilaga 1. I trädets noder visas dock $\exp(x) = r$ eftersom det är detta som ska användas vid värdering i trädets.

I konstruktionens andra steg ska trädet anpassas efter de rådande räntorna. Även detta sker på ett liknande sätt som för Hull och Whites modell. En skillnad infinner sig dock när man kommit till den punkt i anpassningen då man använder ekvation 11 i beskrivningen av steg 2 ovan. Från och med detta moment kommer modelleringen att skilja sig från den beskrivna, på grund av att räntefördelningen i modellen inte är densamma. Vid steg m i modelleringen kommer formeln för priset för en nollkupongobligation med löptid $m+1$ nu istället få ett utseende enligt ekvation 21 nedan.

$$P_{m+1} = \sum_{j=-n_m}^{n_m} Q_{m,j} \exp[-g(\alpha_m + j\Delta x)\Delta t] \quad \text{Ekvation 21}$$

För att anpassa modellen till rådande räntestruktur ska värdet på x i trädet justeras med variabeln α_m . Denna variabel fås genom att använda någon enkel numerisk metod, exempelvis Newton-Raphsons metod.

Även beräkningen av hjälpvariabeln Q påverkas på motsvarande sätt. För en viss vertikal position i trädet, j , ges nu variabeln Q ur ekvation 22 nedan. De olika värdena på k motsvarar de noder i nästkommande tidssteg som utgör den aktuella nodens söner och $q(k,j)$ utgör sannolikheten för steget från noden (m,k) till noden $(m+1,j)$.

$$Q_{m+1,j} = \sum_k Q_{m,k} q(k,j) \exp[-g(\alpha_m + k\Delta x)\Delta t] \quad \text{Ekvation 22}$$

På motsvarande sätt görs anpassningen för samtliga steg i trädet. Precis som efter det första steget lagras sedan $\exp(x)=r$. Resultatet är då ett träd med de lognormalfördelade korta räntorna som kan användas för värdering av olika räntederivat.

2.8 Räntornas antagna fördelning

Vid värdering av vissa räntederivat med modeller för den korta räntan uppvisar Hull och Whites modell en stor fördel framför de andra modellerna. Denna modell har nämligen analytiska uttryck för framtida obligationspriser. Fördelen med detta är att man vid modelleringen för många typer av räntederivat inte behöver modellera hela den underliggande tillgångens löptid. Istället är det tillräckligt att modellera fram till derivatets inlösen där dess värde kan fås fram från det analytiska uttrycket för den underliggande tillgången. De andra studerade modellerna antar lognormalfördelade räntor. Därför finns det för dessa modeller inga analytiska uttryck för obligationspriserna under dess löptider.

I detta examensarbete har man dock ingen nytta av de analytiska uttrycken eftersom modellering måste ske fram till instrumentets inlösen. Ingen av modellerna har därför någon fördel i detta avseende gentemot någon av de andra.

En annan sak som påverkas av valet av modell och därmed fördelning av räntorna är möjligheten att få negativa räntor i trädet. Vid användning av Hull och Whites modell är det möjligt att räntorna blir negativa. Med lognormalfördelade räntor som i de övriga modellerna finns däremot ingen sådan möjlighet. I detta avseende kan antagande om lognormalfördelade räntor vara att föredra då man intuitivt vill undvika möjligheten, om än med liten sannolikhet, att under perioder ha negativa räntor.

De volatilitetsparametrar som ingår i modellerna får också olika betydelse beroende på vilken räntefördelning som väljs, något som också diskuteras i presentationen av de olika modellerna ovan. I fallet med normalfördelade räntor motsvarar σ standardavvikelsen för de årliga förändringarna av den korta räntan. När en lognormalfördelad fördelning antas betecknar denna parameter standardavvikelsen för den proportionella förändringen av den korta räntan. Marknadens aktörer är i större utsträckning vana att se volatilitet som resultat från en lognormalfördelad modell, som är fallet med Black och Scholes och besläktade modeller, och lognormalfördelade räntor kan därför vara att föredra.

2.9 Kalibrering

Under beskrivningen av modellerna och hur man använder denna i ett trinomialträd har parametrarna a och σ antagits vara givna. Så är dock inte fallet och en viktig del vid användande av modellen består just i att välja dessa parametrar. Det är detta moment som utgör kalibreringen av modellen.

Kalibreringen består i att en anpassning till marknadsdata sker. Här utgår man från marknadsdata som rör liknande instrument som det derivat som ska prissättas. Grundidén är att man genom en kalibrering efter dessa marknadsdata erhåller en modell som prissätter befintliga instrument korrekt och därmed har en bra grund för att kunna prissätta andra instrument av samma typ. För att få en så korrekt modell som möjligt är det viktigt att kalibrera modellen efter instrument som är så lika som möjligt de instrument som man vill prissätta.

Vid kalibreringen utgår man från något mått som sedan minimeras enligt en numerisk metod. Det vanligaste måttet som används för minimering är:

$$M = \sum (P_i - V_i)^2 \quad \text{Ekvation 23}$$

Parametern P_i betecknar här marknadspriset och V_i står för det pris som erhållits ur modellen för samma instrument. Summeringen av differensen görs sedan för alla olika instrument som används vid kalibreringen. Det modellerade priset, V_i , och därmed differensen är beroende av de ingående parametrarna i modelleringen. Genom kalibreringen fås de värden på de ingående parametrarna som minimerar kalibreringsmålet. Därmed kommer modellen att prissätta instrumenten som ligger till grund för kalibreringen till ett pris som ligger så nära det verkliga marknadspriset som möjligt. Diskussion om hur kalibreringen kan genomföras för de anpassade modellerna i detta examensarbete förs i kapitlet om modellenanpassning.

2.10 Tidsberoende volatilitetsparametrar

Vid användande av modeller för korta räntan består ett viktigt beslut i hur man väljer sina volatilitetsparametrar. Möjligheten finns att antingen att se dessa som konstanta eller som funktioner av tiden. Det finns ett antal olika uppfattningar om hur parametrarna ska behandlas. Som en inledning på resonemanget kan man slå fast att valet av hur parametrarna ska behandlas styr modellens effektivitet att prissätta instrument, beräkningens omfattning, modellens stabilitet samt kalibreringens komplexitet. Fördelen med att låta volatilitetsparametern och/eller mean reversion parametern vara funktioner av tiden är att modellerna på så sätt mer exakt kan anpassas till marknadspriser för instrument som handlas aktivt.

Vid användande av tidsberoende sker en parametrisering. Då väljs hur funktionerna för de tidsberoende parametrarna tolkas. Vanligt är till exempel att välja dessa funktioner som stegvis konstanta funktioner. Stegen kan med fördel väljas så att de anpassas till de instrument som används till kalibrering samt de instrument som ska värderas.

Ett problem när tidsberoende parametrar används är att man indirekt gör en utsaga om volatilitetsstrukturens utseende vid framtida tidpunkter. Detta problem uppstår för alla Markov modeller och alltså även de i detta examensarbete studerade modellerna

Undersökningar har gjorts för att testa hur tidsberoende parametrar påverkar modellernas resultat. Upphovsmännen till en av de studerade modellerna, Hull och White, har genomfört utförliga undersökningar av denna typ. Dessa visar att modellernas effektivitet att prissätta befintliga instrument korrekt påverkas marginellt. Hull och White slår därför fast att tidsberoende volatilitetsparametrar snarast leder till en överparametrisering av modellen. De anser därför att det för Markov modeller endast ska finnas en tidsberoende parameter. Vidare anser Hull och White att denna parameter ska användas för att anpassa modellen till den befintliga räntestrukturen.

I detta examensarbete har modellering och implementering för Hull och Whites modell samt Black och Karsinskis modell skett med konstanta volatilitetsparametrar. En av huvudsyftena med Black, Derman och Toys modell är dock att använda tidsberoende parametrar. I implementeringen av denna har därför tidsberoende parametrar används. De övriga implementeringarna kan, med små förändringar, anpassas till att kunna använda tidsberoende parametrar.

3 Modellanpassning

För att kunna använda modellerna som studeras i detta arbete för värdering av callables krävs en anpassning av den trädkonstruktionsalgoritm som beskrivs i teoridelen. Detta utgör en viktig och nödvändig del av examensarbetet. I detta kapitel presenteras också förslag på hur kalibrering av modellerna ska ske.

För värdering av callables i ränteträd förutsätts att trädet har noder motsvarande de tidpunkter vid vilka inlösen av instrumenten kan ske. För att detta ska vara möjligt krävs att trädet kan ha varierande längd på de olika tidsperioderna. I och med detta kan man vid konstruktion av trädet välja längden på de modellerade tidsperioderna så att tidpunkterna för möjlig inlösen motsvaras av en modellerad tidpunkt i trädet. Redogörelsen för modellanpassningen är generell och gäller för alla de tre studerade modellerna i de fall där inget annat anges. Redogörelsen baseras därför i enlighet med kapitel 2.7.3 för en variabel x som beskriver utvecklingen av räntans antagna funktion. I detta examensarbete kan detta väljas till $x=r$ (för Hull och Whites modell) alternativt $x=\ln r$ (för de övriga två modellerna).

3.1 Val av tidpunkter för nodernas placering

Det första steget i den anpassade algoritmen är att bestämma för vilka tidpunkter som noderna ska placeras. Som i all modellering väljs modelleringens första nod till tiden för modelleringens start och den sista diskreta tiden väljs till det aktuella instrumentets inlösen. Övriga noder väljs för att uppfylla de krav som ställs för modellering av callables. I fallet med callables ska noder bland annat finnas vid de tidpunkter då tidig inlösen kan ske. Dessutom förenklas modelleringsarbetet av att låta noder modelleras för de tidpunkter då utdelningar äger rum. Troligtvis vill man dela in den modellerade tiden i fler perioder än de noder som blir resultat av tidig inlösen samt utdelningsdatum för att öka noggrannheten i trädet. I detta examensarbete sker detta genom jämn fördelning av de återstående önskade tidsperioderna över den modellerade tiden.

3.2 Storlek på räntans förändring mellan två intilliggande noder

Nästa steg i modellanpassningen består i att bestämma de vertikala avstånden, mätt i skillnad i den korta räntan, mellan två närliggande noder vid samma diskreta tid i trädet. Här måste hänsyn tas till att tidsstegen är av varierande längd och man kan inte, som i algoritmens grundutförande, anta att detta avstånd är samma för alla de modellerade tiderna i trädet.

I varje tidpunkt placeras noder för $x=0$. Därefter placeras noder på samma sätt som i standardfallet enligt ett viss bestämt intervall. Vid varje tidsteg t_i i trädet placeras noderna med värden på den korta räntan enligt $\pm\Delta x_i, \pm 2\Delta x_i, \dots, \pm m_i\Delta x_i$ där m_i motsvarar den högsta respektive lägsta noden för det aktuella tidssteget. Hur denna bestäms redogörs för i följande avsnitt. För varje tidssteg, t_i , bestäms ränteintervallet mellan två intilliggande noder, Δx_i , så att det är tillräckligt för att representera volatiliteten för den korta räntan vid den aktuella tidpunkten. För snabbast möjliga konvergens väljs Δx_i enligt ekvation 24 nedan.

$$\Delta x_i = \sigma(t_{i-1})\sqrt{3(t_i - t_{i-1})} \quad \text{Ekvation 24}$$

3.3 Val av metoder i trädet

Nästa steg i trädkonstruktionen består i att välja hur noderna ska knytas samman. Detta kommer också att avgöra värdet på m_i för de olika stegen i trädet. I den anpassade algoritmen är man inte begränsad till de metoder som används i standardalgoritmen och visas i figur 2. Valet beror nu istället på hur den korta räntan antas utvecklas under kommande tidsperiod.

Vid varje nod i trädet binds den samman med tre söner, precis som i standardalgoritmen. De tre noderna motsvarar också tre intilliggande noder i nästkommande tidssteg och kan på samma sätt ses som ett steg uppåt, ett steg neråt och ett steg mellan dessa. För variabeln x i den aktuella noden beräknas den förväntade förändringen och variansen av förändringen. Trädet konstrueras sedan så att dessa är desamma i trädet som de beräknade.

Antag att trädets konstruerats fram till och med steg i och att trädet nu ska byggas vidare. Från noden $j\Delta x_i$ i steg i ska trädet utökas till de tre noderna $(k-1)\Delta x_{i+1}$, $k\Delta x_{i+1}$, och $(k+1)\Delta x_{i+1}$ i nästa steg; $i+1$. Den nod till vilken det mittersta steget ska ansluta till, $x_{i+1,k}$, väljs så att räntan i denna nod, $k\Delta x_{i+1}$, är så nära som möjligt den förväntade räntan. Väntevärdet av räntans förändring i noden från vilken steget ska ske är approximativt $E(dx) = x^* a(t_i)(t_{i+1}-t_i)$. Värdet på k som alltså betecknar vilken nod som det mittersta steget ska leda till väljs så att $x_{i+1,k}$ är så nära som möjligt det förväntade värdet som är $x_{i,j} + x_{i,j}^* a(t_i)(t_{i+1}-t_i)$. Genom att välja k till det heltal som är närmst $(j\Delta x_i + x^* a(t_i)(t_{i+1}-t_i)) / \Delta x_{i+1}$ uppnås detta. Samtidigt garanterar detta att nästa steg hamnar inom $\Delta x_{i+1}/2$ från väntevärdet.

Metoderna i respektive steg i i trädet avgör också den översta och understa möjliga noden i nästkommande steg. Den översta nådda noden i ett visst steg är den högsta nod som länkas till från den översta noden i steget innan. På motsvarande sätt bestäms också den understa noden från den understa noden i steget innan. Den högsta noden i steget $i+1$ som är m_{i+1} bestäms från den högsta noden i steget i , som är m_i och på samma sätt bestäms den lägsta noden, $-m_{i+1}$ bestäms av den lägsta noden, $-m_i$ i steget i .

Nästa steg av trädkonstruktionen är att beräkna de olika sannolikheterna för de möjliga stegen i trädet. Detta görs på motsvarande sätt i den anpassade algoritmen som i standardalgoritmen. Variansen för en viss nod i trädet kan skrivas $V(dx) = \sigma^2(t_i)(t_{i+1}-t_i)$. För sannolikhetsberäkningarna, som redogörs för i appendix 2, används väntevärdet av räntans förändring under en period samt densamma under två perioder.

3.4 Anpassning till rådande räntestruktur

Efter konstruktionen av trädet enligt avsnitt 3.1 – 3.3 ska nu trädet anpassas till rådande räntor. Detta sker till stor del enligt samma metod som i den ursprungliga algoritmen som presenteras i avsnitt 2.7.2. Diskontering mellan de olika tidpunkterna i trädet sker dock med varierande tid eftersom tidsstegen i trädet är av olika längd.

3.5 Kalibrering

Syftet med kalibreringen är att anpassa modellens ingående parametrar efter befintliga instrument med vilka handel sker. Ett allmänt resonemang om kalibrering finns i avsnitt 2.9 i teorikapitlet ovan.

Avsikten i detta examensarbete är att använda den presenterade algoritmen för värdering av callables. Modellen kan med vissa förändringar även användas till andra värderingar. De i detta avsnitt presenterade förslagen till kalibrering gäller dock för det fall då callables ska värderas.

Ett alternativ till kalibrering är att använda olika ränteoptioner, såsom caps och floors, för att ge en allmän kalibrering av modellen. På detta sätt kalibreras många andra räntemodeller och kalibreringsinstrumenten kan då väljas, avseende bland annat löptider, så att kalibreringen blir snabb och effektiv. Detta kan vara särskilt fördelaktigt då volatilitetsparametrarna är en funktion av tiden och parameteriserade. Tidpunkterna som utgör brytpunkter i de parameteriserade volatilitetsfunktionerna kan då väljas så att de överensstämmer med löptiderna för kalibreringsinstrumenten. Om den studerade marknaden är tillräckligt likvid och det finns ett tillräckligt stort utbud av olika optioner kan också hänsyn tas till att inlösenpriserna är olika för de optioner som används som kalibreringsinstrument.

Det förslag till kalibreringsförfarande som presenteras ovan ger en bra allmän kalibrering och skulle kunna användas i de fall då ränteoptioner av liknande typ som kalibreringsinstrumenten ska värderas. För att modellen på ett korrekt sätt ska värdera callables bör istället instrument av just denna typ användas vid kalibreringen.

Författarens uppfattning är att en enkel kalibrering med någon numerisk optimeringsalgoritm skulle ge ett bra resultat. Modellen skulle då anpassas efter de befintliga callables som handlas på den aktuella marknaden. En algoritm som ofta används vid denna typ av kalibreringar är Levenberg-Marquardts algoritm. Denna skulle även kunna användas i detta fall men friheten är stor vid val av algoritm.

En förutsättning för att den föreslagna kalibreringen ska ge ett bra resultat är att antalet kalibreringsinstrument är tillräckligt. I fallet med callables är dock marknaderna relativt illikvida och detta kan innebära att man inte kan åstadkomma en bra kalibrering.

4 Implementering och resultat

I detta kapitel vill författaren i grova drag visa hur implementeringen genomförts. Det förekommer inte någon redovisning av kod utan istället visas med hjälp av resonemang implementeringens viktigaste delar. I kapitlet redovisas också resultatet av implementeringen. Resultatredovisningen består till stor del av illustrationer av hur den färdiga implementeringen ser ut och hur den kan användas.

4.1 Implementeringsmiljö

Ett delsyfte med examensarbetet är att implementera de aktuella modellerna i Algorithmicas utvecklingsmiljö Quantlab, som är ett utvecklingsverktyg för kvantitativ analys. I Quantlab finns möjligheten att via ett eget språk, Qlang, programmera och modellera. På grund av de i detta examensarbete ingående modellernas komplexitet genomförs implementeringen istället via Quantlabs API. Detta kan ses som en koppling mellan en extern programmeringsmiljö och Quantlabs utvecklingsmiljö. Genom att skapa program i en annan miljö och i ett annat språk, i detta fall C++, ökas flexibiliteten. Via kopplingen till Quantlab erhålles möjligheten att använda dess användargränssnitt och kopplingar till realtidsdata.

Den faktiska kopplingen sker via Windows dynamic link library (DLL). Det konstruerade C++ programmet resulterar i en DLL som Quantlab sedan läser in och det konstruerade programmets objekt och funktioner blir sedan tillgängliga i Quantlab. Där kan de användas på samma sätt som de standardobjekt och funktioner som finns tillgängliga i miljön.

4.2 Trädrepresentation

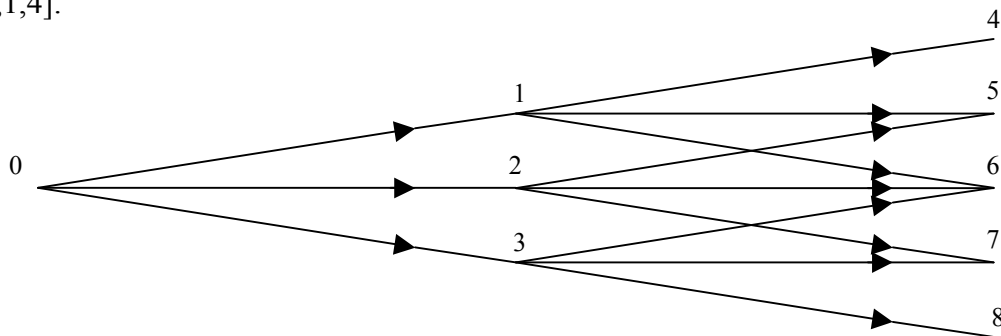
För att på ett bra och enkelt sätt kunna gestalta ränteträd i C++ har författaren i huvudsak valt att representera trädet med hjälp av två vektorer. Den första vektorn innehåller alla noder och deras position i vektorn bestäms av positionen i trädet. Varje nod representeras i sin tur av en datastruktur som innehåller all nödvändig information, såsom kort ränta, sannolikheter för de olika möjliga stegen, länkar till nodens söner o.s.v. I datastrukturerna finns även de olika parametrar som behövs vid anpassning av trädet till rådande markandsdata.

Den andra vektorn är en heltalsvektor och fungerar som en positionsvektor. I denna vektor ryms de heltal som utgör index i nodvektorn för de första noderna för ett visst tidssteg i trädet.

I syfte att visa hur nodernas placering i nodvektorn och positionsvektorns index bestäms hänvisas till figur 6 i kombination med följande beskrivning. Numreringen av noderna börjar med 0. Detta index tilldelas noden $(0,0)$, dvs. den nod som infaller vid modelleringens start. Nodernas index och position i nodvektorn räknas sedan upp med ett steg och vid ett nytt tidssteg tilldelas nästkommande index till den översta nod i detta tidssteg. Position 1 i nodvektorn kommer därför att upptas av nod $(1,1)$, nod $(1,0)$ tilldelas sedan position 2 osv. Numreringen av noderna fortsätter sedan på detta sätt till trädets samtliga noder placerats i nodvektorn.

Positionsvektorns uppgift är att på ett smidigt sätt möjliggöra för adressering av noderna i vektorn. På varje position i vektorn, som är av samma längd som antalet steg i trädet, lagras indexet för den första noden för det motsvarande steget. I vektorns första position, 0, återfinns heltalet 0 som visar att detta är indexet för den första noden för detta tidssteg. På motsvarande

sätt lagras heltalet 1 i positionsvektorns position 1 osv. I figur 6 nedan visas hur ett mycket enkelt träd skulle kunna se ut. Heltalet vid varje nod visar vilken placering i nodvektorn som den aktuella noden lagras på. För detta träd skulle positionsvektorn få följande innehåll: [0,1,4].



Figur 6. Exempel på träd för representation

Förutom de ovan redovisade vektorerna ingår också ett antal andra strukturer och parametrar i trädrepresentationen. Antal noder, antal tidssteg och användarens val av modellparametrar är exempel på saker som representeras i det slutgiltiga trädet.

4.3 Funktionalitetskrav

I den färdiga implementeringen ska användaren i Quantlab kunna välja de ingående parametrarna till modellerna. Vidare ska användaren beskriva det instrument som ska värderas. Förutom start och sluttid för modelleringen anger användaren de tider för vilka tidig inlösen kan ske samt de förutbestämda priser som inlösen kan ske till. Dessutom anges datum för kuponger samt dess storlek. Antalet steg som den modellerade tidsperioden ska delas in i avgör noggrannheten och omfattningen av modelleringen ska också kunna väljas. Användaren ska också välja vilken räntekurva som modellen ska anpassas efter.

4.4 Programmering i Quantlab

I Quantlabmiljön resulterar implementeringen i ett antal objekt med tillhörande funktioner. De viktigaste objekten utgörs av de olika typer av träd som kan konstrueras. Objekten *hwtree*, *bktree* och *bdttree* representerar i tur och ordning träd som resulterar av användande av Hull och Whites modell, Black och Karsinskis modell respektive Black, Derman och Toys modell. Dessa objekt har sedan ett antal funktioner för värdering av olika instrument, ändring av ingående parametrar osv. Dessutom rymmer dessa objekt i sin tur objekt av typen *node* motsvarande alla noder i trädet. Nodobjekten innehåller all information som finns lagrad i de olika noderna och denna information används såväl vid konstruktionen av trädet som vid värdering i trädet.

Följande redovisning ska förhoppningsvis ge en bra bild av hur den färdiga implementeringen används i Quantlab.

Quantlabs funktionalitet baseras på ett eget språk, Qlang, i kombination med ett antal strukturer. Det är i detta språk man har tillgång till de konstruerade objekten. Figur 7 nedan visar hur Hull och Whites modell representerad i ett trinomialträd, *hwtree*, används i Qlang. Funktionen *my_tree* tar ett antal inparametrar och returnerar ett antal utparametrar. I Quantlab kommer inparametrarna att matas in via dialogrutor och resultatet kommer att presenteras i en

tabell. Redovisning av de olika ingående parametrar följer nedan i samband med att inmatningen presenteras. Syftet med den funktion som skapas i figur 7 är att värdera en callable som beskrivs av användaren. Vidare studeras skillnaden mellan priset på en callable och priset på motsvarande obligation utan möjlighet till tidig inlösen. På så sätt får man en uppfattning om värdet på premien för möjligheten till tidig inlösen.

I figur 7 är ett antal positioner i Qlangkoden numrerade. Dessa ska med tillhörande beskrivningar nedan underlätta för läsaren att förstå grunderna i hur programmering i Quantlab fungerar samt hur den färdiga implementeringen är tänkt att användas. Det första steget i programmet är att initiera funktionen. Detta sker vid markeringen 1 i figur 7. Beteckningen `out` används i Quantlab för att möjliggöra att funktionens resultat kan användas i tabeller eller grafer. Därefter definieras typen för funktionens resultat. I detta exempel kommer funktionen att returnera en vektor av tal, vilka i Quantlab representeras som "number". I funktionsdefinitionen ingår också funktionens namn, "my_hwtree". Därefter namnges alla inparametrar som funktionen kräver. Dessa förklaras i samband med att de används i programmet.

Som tidigare behandlats i denna rapport använder sig den implementerade modellen av den rådande räntestrukturen som inparameter. Användaren väljer vilken räntemarknad som modellen ska anpassa till. Detta sker i Quantlab genom att namn på denna marknad väljs och fungerar som inparameter till funktionen "my_hwtree". Vid markering 2 i figur 7 tas den aktuella räntemarknadens namn, som betecknas `c_n` samt datumet för början av modelleringen som är en annan inparameter, `trade_d`, som inparametrar till Quantlabs inbyggda funktion `bootstrap`. Denna funktion skapar en räntekurva från instrumenten på den aktuella räntemarknaden genom att använda sig av bootstrapping. Denna funktion resulterar i `f_r` som är ett objekt av Quantlabtypen `fit_result`. Typen innehåller all information om den aktuella räntekurvan. Det resulterande objektet används senare i koden vid konstruktion av ränteträd. Istället för att skapa räntekurvan med bootstrapping kan också andra, mer avancerade, metoder användas vid konstruktion av räntekurvorna. Även dessa funktioner resulterar i objekt av typen `fit_reusult`.

I nästa steg av programmet används de trädobjekt som skapats i C++ och länkats till Quantlab via dess API. I detta exempel används Hull och Whites modell. Vid markering 3 i figur 7 konstrueras trädobjekt och några av dess medlemsfunktioner används för värdering. Första momentet består i att skapa trädobjektet. Detta görs genom att anropa funktionen `hwtree` med de inparametrar som krävs. Den första inparametern är en vektor med datum för vilka man vill att noder ska infalla. I fallet med callables utgörs dessa av de datum då möjlighet till tidig inlösen finns. I Quantlab representeras datum med den inbyggda datatypen `date`. Om det aktuella instrumentet även har utdelningar under dess löptid konstrueras trädet så att noder även finns för de datum då dessa kan ske. Utdelningsdatumen utgör därför också en inparameter till trädkonstruktorn. Den tidigare redovisade räntestrukturen, `f_r`, tas också som en inparameter av funktionen för att kunna anpassa det modellerade trädet till denna. För konstruktion av trädet används också start och slutdatum för modelleringen, antalet steg i trädet samt modellparametrarna `alpha` och `sigma` som inparametrar.

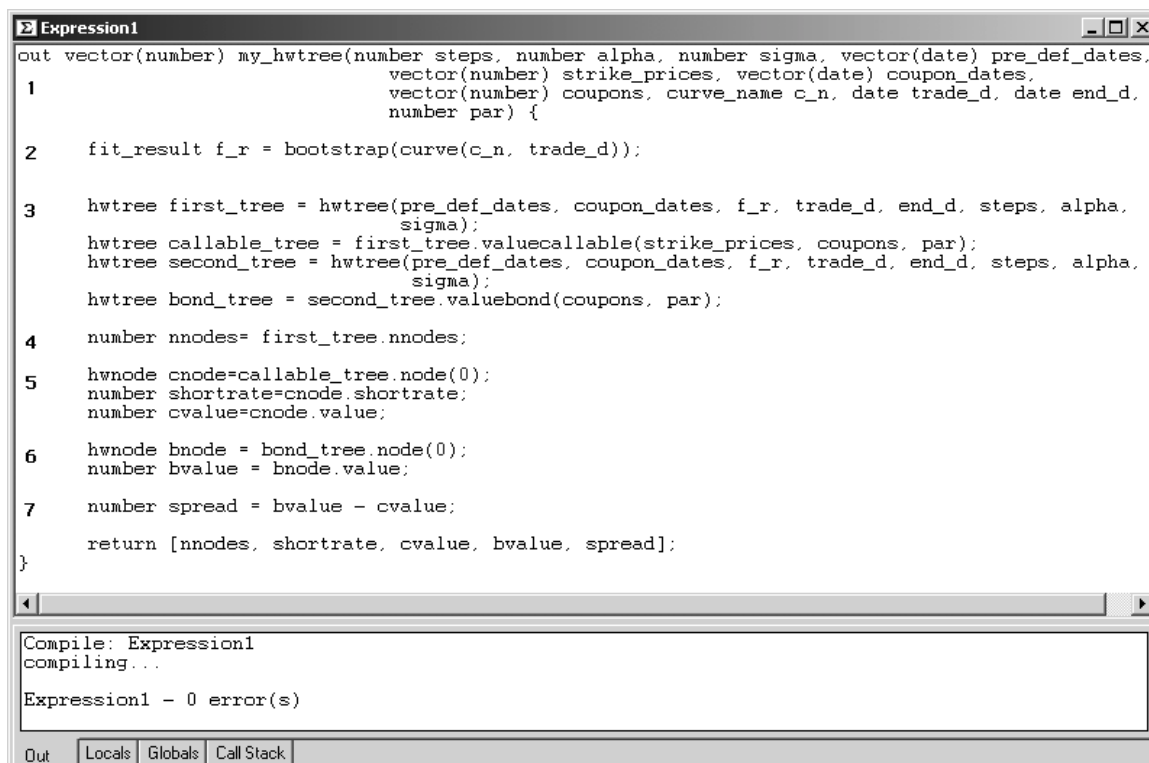
Efter att trädet konstruerats kan det sedan användas på olika sätt. I detta exempel skapas två träd med samma inparametrar, `first_tree` och `second_tree`. Det förstnämnda trädet används för värdering av en callable med funktionen `valuecallable`. Denna funktion tar de priser till vilka tidig inlösen kan ske, `strike_prices`, som en av sina inparametrar. Dessa utgörs av en vektor med tal, `number` i Quantlab. Dessutom tar funktionen en vektor med storleken av de förekommande kupongerna (i procent av par) och parvärdet som inparametrar. Det andra

trädet, `second_tree`, används sedan för att värdera motsvarande obligation utan möjlighet till tidig inlösen. Detta sker via funktionen `valuebond`. Denna funktion tar samma inparametrar som `valuecallable` med undantag för `strike_prices` som inte behövs när en vanlig obligation ska värderas. Båda dessa funktioner returnerar ett nytt träd vars noder beskriver räntans utvecklig och tillhörande parametrar samt värdet för det aktuella instrumentet. Dessa träd ges namnen `callable_tree` respektive `bond_tree`.

Vid markeringen 4 i programmet i figur 7 ges exempel på en medlemsfunktion. I detta fall är det funktionen `nnodes` som returnerar antalet noder i trädet. Liknande funktioner är implementerade för övriga parametrar i trädet.

Nästa steg i programmet är att få fram värdet på de värderade instrumentet. Detta sker vid markering 5 i figur 7. Genom att använda den funktion, `callable_tree.node`, som tar ett index som inparameter och returnerar noden med detta index i nodvektorn erhålles noden med index 0. Ur denna nod kan man sedan med hjälp av dess medlemsfunktioner få önskad information. Ur noden för `callable` tas först räntan och sedan instrumentets värde i denna nod, `cvalue`. Därefter tas vid markering 6 värdet för motsvarande vanliga obligation.

Vid markering 7 i figur 7 görs en jämförelse mellan de två värdena för att få ett värde på den inbyggda optionspremien. Slutligen returneras en vektor med några av de framräknade värdena. Antalet noder i trädet, den korta räntan i nod 0, `callable`värdet, värdet för den vanliga obligationen samt differensen mellan de två instrumenten returneras i detta exempel.



```

out vector(number) my_hwtree(number steps, number alpha, number sigma, vector(date) pre_def_dates,
1      vector(number) strike_prices, vector(date) coupon_dates,
      vector(number) coupons, curve_name c_n, date trade_d, date end_d,
      number par) {
2      fit_result f_r = bootstrap(curve(c_n, trade_d));
3      hwtree first_tree = hwtree(pre_def_dates, coupon_dates, f_r, trade_d, end_d, steps, alpha,
      sigma);
      hwtree callable_tree = first_tree.valuecallable(strike_prices, coupons, par);
      hwtree second_tree = hwtree(pre_def_dates, coupon_dates, f_r, trade_d, end_d, steps, alpha,
      sigma);
      hwtree bond_tree = second_tree.valuebond(coupons, par);
4      number nnodes= first_tree.nnodes;
5      hwnode cnode=callable_tree.node(0);
      number shortrate=cnode.shortrate;
      number cvalue=cnode.value;
6      hwnode bnode = bond_tree.node(0);
      number bvalue = bnode.value;
7      number spread = bvalue - cvalue;
      return [nnodes, shortrate, cvalue, bvalue, spread];
}

```

Compile: Expression1
compiling...

Expression1 - 0 error(s)

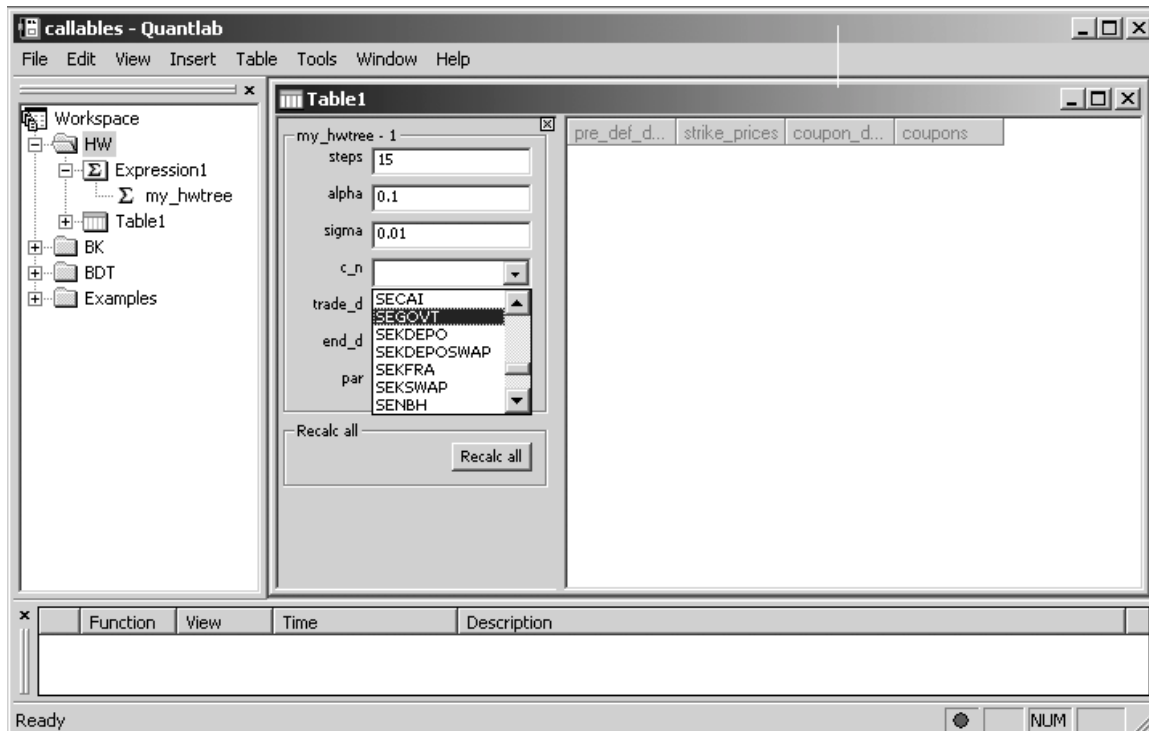
Out Locals Globals Call Stack

Figur 7. Programmering i Quantlab

4.5 Användande i Quantlab

Programmet som presenteras i avsnitt 4.4 beskriver hur trädobjekten ska användas vid programmering i Quantlab. Figur 8 och figur 9 visar den miljö som användaren möter och i vilken de ingående parametrarna ska väljas. Den enklaste typen av ingående parametrar är de

där användaren väntas ange tal via inmatning. Dessa ifylles av användaren i dialogrutan. Antalet steg i trädet, steps, alpha och sigma är av denna enkla typ. I figur 8 är dessa ifyllda av användaren. Nästa parameter som ska anges är namnet på den räntemarknad efter vilken man vill anpassa modellen. Valet av detta sker via Quantlabs dialogruta och visas i figur 8. Programmet presenterar de marknader till vilka data finns tillgängliga och användaren kan välja mellan dessa.

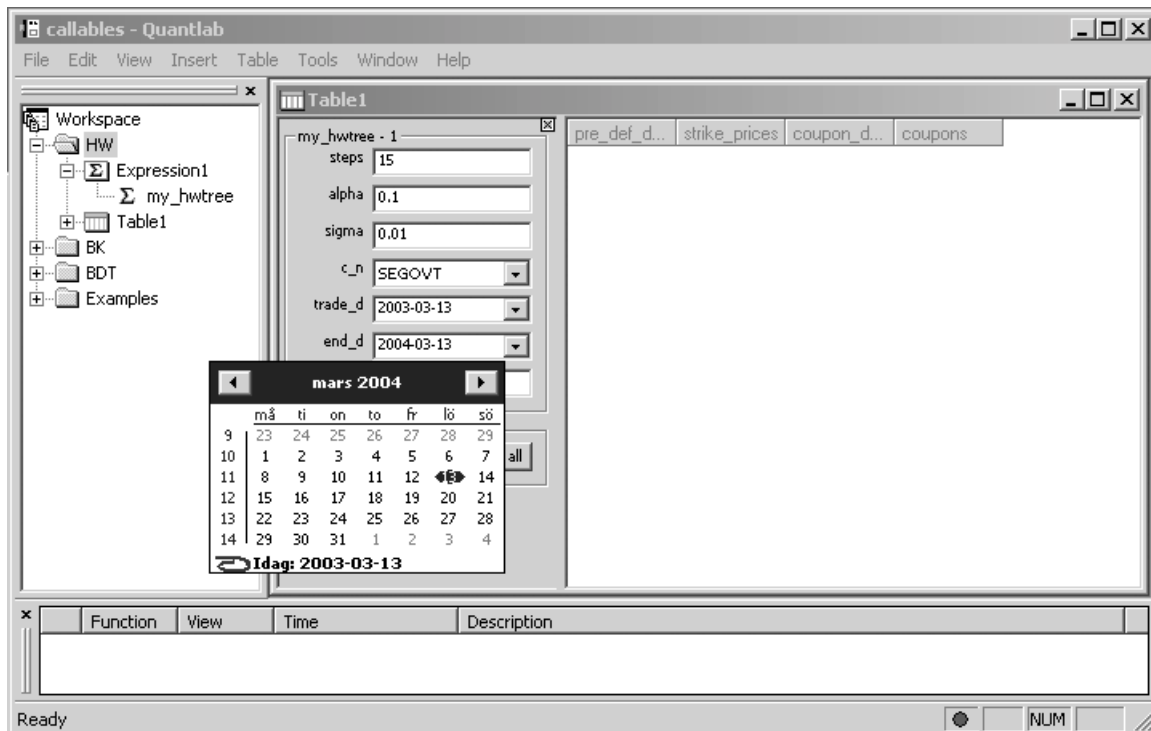


Figur 8. Exempel på användning i Quantlab

Efter att ha valt räntemarknad ska sedan start samt slutdatum för modelleringen väljas. Detta sker med hjälp av Quantlabs kalenderfunktion och illustreras i figur 9.

Nästa steg är för användaren att mata in datum för möjlig tidig inlösen samt priser till vilket dessa kan ske. På samma sätt anges sedan datum för kupongbetalningar samt dess storlekar. Dessa parametrar anges i respektive kolumn i den tabell för inmatning som presenteras i dialogrutan. Genom att lägga till rader i kolumnerna för inmatning av dessa parametrar styr användaren också antalet datum för möjlig tidig inlösen respektive utdelningar.

Efter att ha valt samtliga parametrar ovan påbörjas modelleringen. Denna sker efter den realtidsdata som Quantlab har tillgång till och modelleringen kommer att göras om med förutbestämda intervall enligt uppdaterade data. Möjligheten finns också att via Quantlab ändra i de instrument som ligger till grund för den aktuella räntekurvan för att se hur detta påverkar modelleringen.



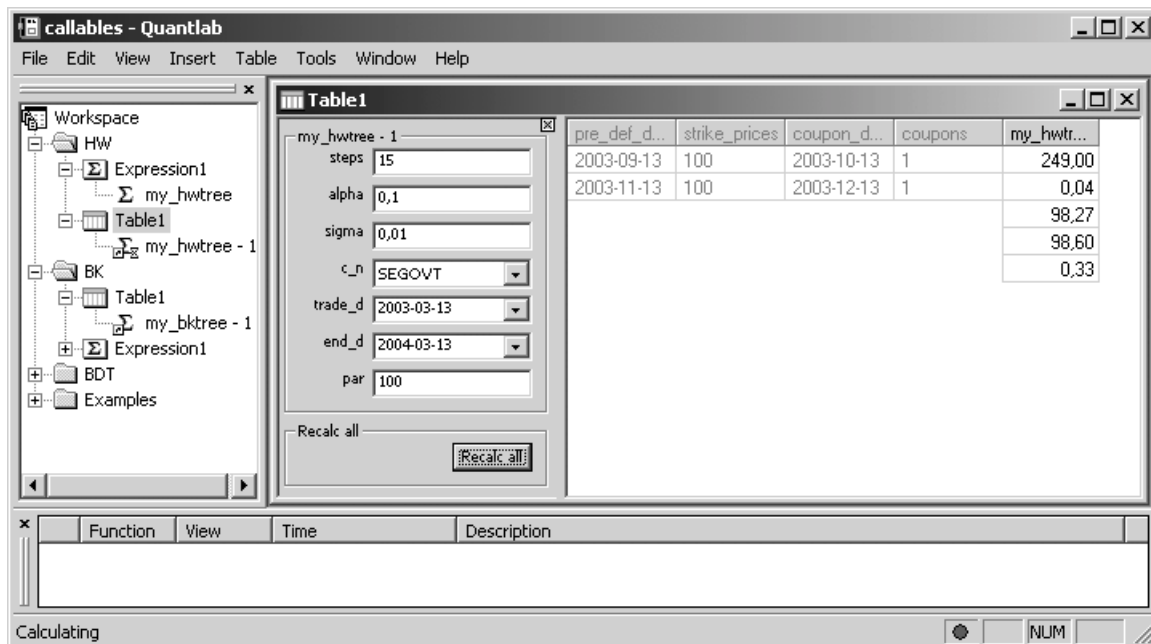
Figur 9. Exempel på användning i Quantlab, fortsättning

Denna redovisning av användande baseras på programexemplet i avsnitt 4.4. I programmeringssteget finns det en stor flexibilitet för användaren att utnyttja modellen på det sätt som önskas.

4.6 Resultat i Quantlab

Hull och Whites modell

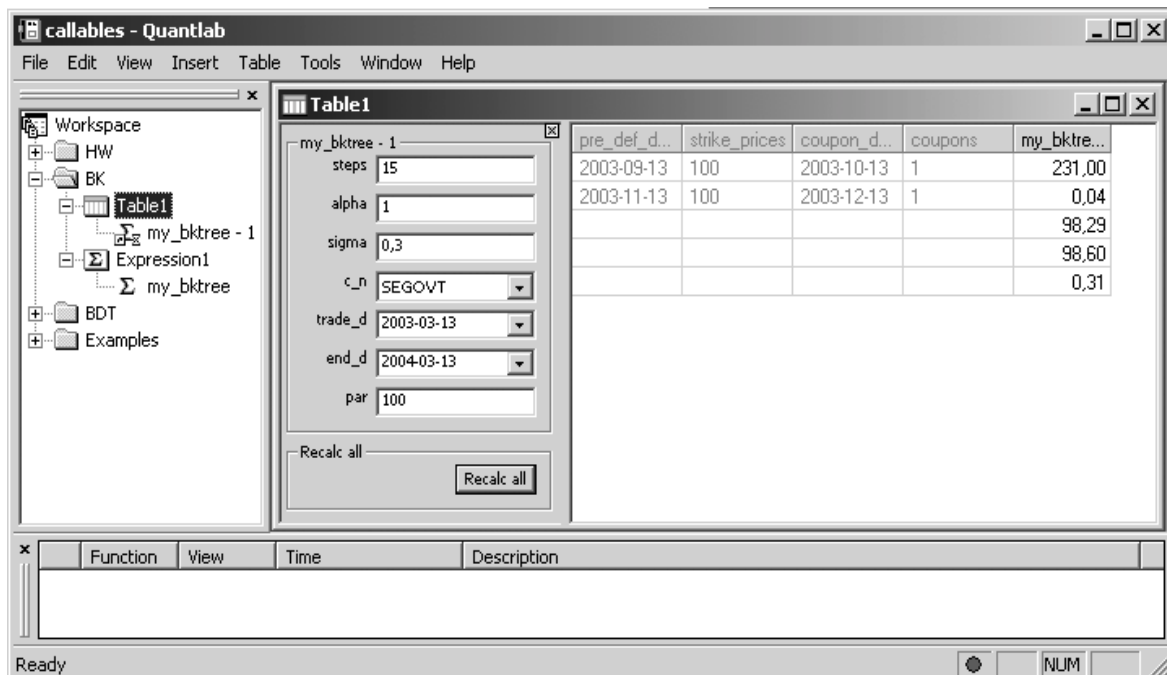
Resultatet från det exempel som redovisats i avsnitt 4.4 och 4.5 visas i figur 10 nedan. Enligt det program som presenterades i avsnitt 4.4 redovisas de returnerade parametrarna i den femte kolumnen i tabellen, med rubriken `my_hwtree`. Resultatet redovisas i den ordning som parametrarna returnerats från programmet. Översta raden anger antalet noder i trädet, andra raden räntan i den första noden, tredje raden priset för en callable, fjärde raden priset för motsvarande vanliga obligation och i femte raden redovisas slutligen differensen mellan de två instrumenten. Den redovisade differensen visar, som behandlats tidigare, värdet på den inbyggda option som finns i instrument av typen callable.



Figur 10. Resultat, Hull och Whites modell

Black och Karsinskis modell

Användande av Black och Karsinskis modell i den färdiga implementeringen påminner till mycket stor del av användande av Hull och Whites modell. I figur 11 nedan visas resultatet av denna modell. Inmatande av parametrarna sker på samma sätt som i Hull och Whites modell.



Figur 11. Resultat, Black och Karsinskis modell

Det program, vars resultat redovisas i figuren, är skrivet på samma sätt som det som redovisas för Hull och Whites modell i kapitel 4.4. Den väsentliga skillnaden sker vid markering 3 i figur 7 ovan. Istället för att här skapa objekt av typen hwtree används nu istället typen bktree. I övrigt är programmeringen och användandet detsamma.

Black, Derman och Toys modell

Motsvarande exempel för Black, Derman och Toys modell redovisas i figur 12 nedan. Även i detta fall har programmeringen skett på liknande sätt som i figur 7. I det program som ligger till grund för detta resultat har dock objekt av typen `bdttree` använts istället för som tidigare `hwtree` respektive `bktree`. Ytterligare en skillnad i detta exempel utgörs av att användaren får mata in diskreta värden ur volatilitetsfunktionen samt dess derivata, något som tas som inparametrar vid trädkonstruktionen. Resultatredovningen sker på samma sätt som tidigare.

The screenshot shows the QuantLab software interface. The main window is titled 'callables - QuantLab'. On the left is a workspace tree with folders for 'HW', 'BK', 'BDT', 'Expression1', 'Table1', and 'Examples'. The 'Table1' window is active, showing parameters for 'my_bdttree - 1':

- steps: 6
- c_n: SEGOVT
- trade_d: 2003-03-13
- end_d: 2004-03-13
- par: 100

Below the parameters is a 'Recalc all' button. To the right is a table with the following data:

pre_def_ti...	strike_pric...	coupon_d...	coupons	vol	volprim	my_bdttr...
2003-09-13	100	2003-09-13	1	0.3	0.5	32,00
2003-11-13	100	2003-11-13	1	0.4	0.5	0,03
				0.5	0.5	98,25
				0.6	0.5	98,60
				0.7	0.5	0,35
				0.8	0.5	

At the bottom of the window, there is a status bar with 'Calculating' and a 'NUM' button.

Figur 12. Resultat, Black, Derman och Toys modell

4.7 Jämförelser mellan de olika modellernas resultat

I resultatredovisningarna ovan framgår att de tre modellerna ger närliggande resultat. Detta gäller under förutsättning att parametrarna väljs på det sätt som gjorts. Observera att volatilitetsparametrarna enligt teorikapitlet i de tre modellerna har olika innebörd. Ingen av modellerna uppvisar någon systematisk skillnad i värdering gentemot någon av de andra modellerna. Detta förutsätter att modellerna anpassas till samma räntestruktur och att samma uppsättning instrument används vid kalibreringen.

Vid kalibreringen har dock modellerna lite olika egenskaper. Tidigare studier av modellerna i deras standardutförande visar att modeller som antar normalfördelade räntor uppvisar en något snabbare och effektivare kalibrering vid konstanta volatilitetsparametrar. Med tidsberoende parametrar är dock inte resultatet lika entydigt. En slutsats som däremot då kan dras är att kalibreringen kommer att leda till att parametrarna för modeller som antar lognormalfördelade räntor kommer att variera i mycket större grad än motsvarande parametrar för modeller med normalfördelade räntor.

4.8 Exempel på ytterligare användning

De modeller som skapats kan användas på en mängd olika sätt enligt användarens önskemål. I detta avsnitt visas hur trädkonstruktion för Hull och Whites modell kan användas för att få en uppskattning av olika typer av riskmått. Figur 13 nedan visar resultatet från ett program som syftar till att uppskatta riskmått hos en callable. Inmatningen känns till stor del igen från

exemplena ovan. I detta exempel används dock även en parameter sigmadiff för att få fram ett värde för ett vegamått för instrumentet. Detta mått visar hur differensen mellan en callable och motsvarande vanliga obligation beror på förändringar i den underliggande volatiliteten. Måttet uppskattas i exemplet genom att modelleringen görs om med två olika värden på sigma. I den femte kolumnen med namnet hw_vega presenteras i den första respektive den andra raden värdet på differensen för de två modelleringarna med en skillnad i volatilitet enligt parametern sigmadiff. Differenserna avrundas vid presentationen i Quantlab och får därför samma värde enligt tabellen. Internt lagras dock dessa värden med större noggrannhet och i den tredje raden visas sedan kvoten mellan skillnaden mellan de två differenserna och skillnaden i volatilitet, sigmadiff. På så sätt erhålles en uppskattning av hur differensen, och därmed premien för den inbyggda optionen, beror på volatiliteten.

Ett annat sätt att uppskatta riskmått ur träd är att använda sig av värdena i de olika noderna i trädet. Exempel med detta har genomförts och dess resultat redovisas i den sjätte kolumnen, simple_hw_greeks. Den första raden i denna kolumn visar en tolkning av delta för en callable. I detta fall har delta tolkats som den inbyggda optionens värde beroende av en vanlig obligations värde. Måttet som presenteras är därför skillnaden mellan optionspremien i två av noderna i trädets första steg dividerat med skillnaden mellan obligationsvärdet i samma noder. Andra raden är ett mått som kan sägas utgöra en uppskattning av instrumentets rho. Även detta mått fås direkt ur trädets noder. I detta fall beräknas riskmålet som en kvot mellan differensen av optionsvärdet i två av det första stegets noder och räntan i dessa noder. Detta mått är dock inte helt rättvisande eftersom det bygger på de korta räntor som modelleras i trädet.

Exercise o...	Strike prices	Coupon d...	Coupons	hw_veg...	simple_...
2003-09-13	100	2003-10-13	1	0,35	0,51
2003-11-13	100	2003-12-13	1	0,35	-24,69
				9,71	

Figur 13. Riskmått för modellering med Hull och Whites modell

Ett mer rättvisande mått för rho skulle kunna fås om man på motsvarande sätt som med vega ovan gjorde om modelleringen för olika räntor. Det bästa skulle då vara att skjuta hela räntekurvan med ett visst mått och därefter göra om modelleringen och beräkna rho.

5 Slutord och förslag till framtida studier

I detta kapitel avslutar författaren rapporten. Han summerar här det genomförda arbetet och innehållet i denna rapport. Vidare uttrycker han sina egna åsikter om områden som skulle kunna lämpa sig för fortsatta studier.

Författarens förhoppning är att denna rapport har gett läsaren inblick i och förståelse för det utförda examensarbetet. Han hoppas vidare att läsaren funnit det genomförda arbetet intressant och givande och att rapporten visat att syftet med examensarbetet uppnåtts.

Arbetet och dess omfattning har till stor del överensstämmt med de förväntningar som författaren hade vid arbetets start. Tidsåtgången missbedömdes dock något och arbetet har tagit en dryg månad längre än den ursprungliga planeringen. En viss del av förklaringen till detta utgörs dock av att författaren under drygt två veckor var tvungen att göra uppehåll i arbetet till förmån för andra studier.

Under examensarbetets gång har författaren använt de förkunskaper som han hade innan arbetet påbörjades. Dessa har dock endast utgjort en grund och en stor del av examensarbetet har kommit att bestå i att inhämta ny kunskap. Detta har varit nödvändigt för att kunna genomföra modelleringen såväl som implementeringen av modellerna.

Ett område inom vilket författaren vid arbetets start helt saknade förkunskaper var inom programmering i C++. Den del av implementering som gjorts i C++ är därför av grundläggande karaktär. Med större erfarenhet inom det aktuella programmeringsspråket skulle implementeringen eventuellt kunnat genomföras effektivare och resulterat i programkod som bättre följer programmeringspraxis.

Examensarbetet har gett författaren en god insikt i finansiell modellering, implementering av modeller samt praktisk användning av desamma. Vidare har arbetet inneburit att han fått använda och studera en programvara för finansiella marknader, Quantlab. Efter att ha genomfört detta examensarbete anser sig författaren ha en god grund för att kunna genomföra liknande arbetsuppgifter i framtiden.

De nedan presenterade förslagen på fortsatta studier är bara två exempel på ämnen som skulle kunna resultera i intressanta studier. Området är omfattande och man kan tänka sig ett antal olika ämnen som kan behandlas.

En naturlig utvidgning av det genomförda arbetet är en implementering av de föreslagna kalibreringarna i Quantlab. Trädoobjekten kunde då utökas med funktioner för kalibrering. Användaren skulle sedan kunna välja de instrument som skulle fungera som kalibreringsinstrument och kalibreringen skulle genomföras med hjälp av någon optimeringsfunktion. Även detta skulle kunna styras av val från användaren. I samband med detta skulle också kalibreringens effektivitet kunna studeras. Arbetet kunde vidare undersöka hur val av modellernas volatilitetsparametrar som tidsberoende eller konstanta påverkar kalibreringens utfall och omfattning. Intressant skulle då också vara att studera hur olika val av parametrisering av de ingående parametrarna påverkar resultatet.

En annan intressant studie skulle vara en undersökning av ett segment av de finansiella marknaderna. En sådan studie skulle, med detta examensarbete och andra källor som grund, kunna fokusera på att studera hur väl marknadspriserna överensstämmer med de modellerade

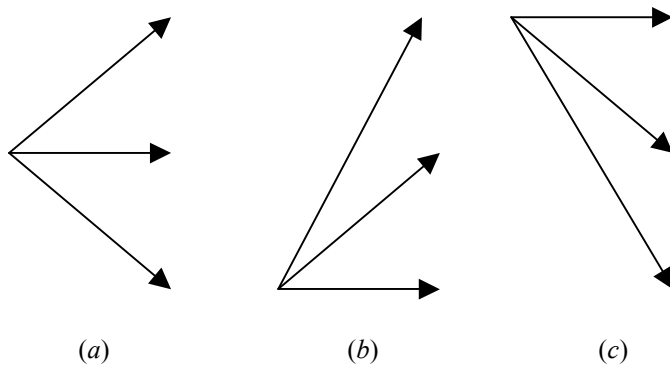
priserna. I en sådan studie kunde också jämförelser mellan olika marknader genomföras för att se om någon av de studerade modellerna stämmer bättre överens på någon viss marknad än på någon annan. Om differens föreligger mellan de modellerade priserna och marknadspriserna skulle det vara intressant att ingående utreda varför denna skillnad uppstår och hur den skiljer sig i storlek mellan olika marknader.

Referenser

- [1] Algorithmica Research AB, 2002, Manual för Quantlab.
- [2] Björk, Tomas, 1995, *Stokastisk kalkyl och kapitalmarknadsanalys – del II Specialstudier*, Institutionen för matematik, KTH.
- [3] Black, F., Derman, E. & Toy, W., 1990, A one-factor model of interest rates and its application to treasury bond options, *Financial Analysts Journal*, February, 33-39.
- [4] Black, F. & Karsinski, P., 1991, Bond option pricing when short rates are lognormal, *Financial Analysts Journal*, July-August, 52-59.
- [5] Boyle, Phelim P., Tan, Ken Seng & Tian, Weidong, 2000, Calibrating the Black, Derman and Toy model: some theoretical results, *Applied Mathematical Finance* 8, 27-48.
- [6] Brigo, Damiano & Mercurio, Fabio, 2001, A deterministic-shift extension of analytically tractable and time-homogeneous short-rate models, *Finance and Stochastics* 5, 369-387.
- [7] Clewlow, Les & Strickland, Chris, 1998, *Implementing Derivatives Models*, John Wiley & Sons.
- [8] Fabozzi, Frank J., 2002, *Interest rate, term structure and valuation modeling*, John Wiley & Sons.
- [9] Gibson, R., Lhabitanat, F.-S. & Talay, D., 2001, Modeling the term structure of interest rates: a review of the literature, *European Finance Review* 4, 69-90.
- [10] Hull, John C. & White, Alan, 1990, Pricing interest rate derivative securities, *Review of Financial Studies*, Vol. 3.
- [11] Hull, John C. & White, Alan, 1994, Numerical procedures for implementing term structure models I: Single-factor models, *Journal of Derivatives*, Vol. 1.
- [12] Hull, John C. & White, Alan, 1995, Using Hull-White interest rate trees, *Journal of Derivatives*, Vol. 3.
- [13] Hull, John C. & White, Alan, 1996, *Hull-White on Derivatives*, Risk Publications.
- [14] Hull, John C., 2000, *Options, Futures & Other Derivatives*, Prentice-Hall, Inc.
- [15] Lippman, Stanley B. & Lajoie, Josée, 1998, *C++ Primer*, Addison Wesley.
- [16] Rebonato, Riccardo, 1998, *Interest-rate option models*, John Wiley & Sons.

Bilaga 1

Sannolikhetsberäkningar i Hull och Whites standardalgoritm



Figur 1. Möjliga metoder i ett trinomialträd

Sannolikheterna väljs så att de motsvarar den väntade förändringen av R^* och dess varians under följande tidsperiod, Δt . Vidare måste sannolikheternas summa, som vanligt, uppgå till ett. Dessa tre förhållanden leder till tre ekvationer som måste vara uppfyllda för sannolikheterna. Tre obekanta och tre ekvationer resulterar i ett lösbart system och de olika sannolikheterna kan beräknas. I fallet då ”standardmetoden” enligt figur 1 används blir ekvationerna:

$$p_u \Delta R - p_d \Delta R = -aj \Delta R \Delta t \quad \text{Ekvation 1}$$

$$p_u \Delta R^2 + p_d \Delta R^2 = \sigma^2 \Delta t + a^2 j^2 \Delta R^2 \Delta t^2 \quad \text{Ekvation 2}$$

$$p_u + p_m + p_d = 1 \quad \text{Ekvation 3}$$

Genom att använda ekvation 4 från uppsatsens huvuddel fås $\Delta R^2 = 3\sigma^2 \Delta t$. Ekvationssystemet ovan löses med hjälp av detta och lösningen blir:

$$p_u = \frac{1}{6} + \frac{a^2 j^2 \Delta t^2 - aj \Delta t}{2} \quad \text{Ekvation 4}$$

$$p_m = \frac{2}{3} - a^2 j^2 \Delta t^2 \quad \text{Ekvation 5}$$

$$p_d = \frac{1}{6} + \frac{a^2 j^2 \Delta t^2 + aj \Delta t}{2} \quad \text{Ekvation 6}$$

På motsvarande sätt erhålles ekvationer för de andra möjliga metoderna i figur 1. Lösningarna för ekvationerna om metoden enligt figur 1b används ger följande sannolikheter:

$$p_u = \frac{1}{6} + \frac{a^2 j^2 \Delta t^2 + aj \Delta t}{2} \quad \text{Ekvation 7}$$

$$p_m = -\frac{1}{3} - a^2 j^2 \Delta t^2 - 2aj\Delta t \quad \text{Ekvation 8}$$

$$p_d = \frac{7}{6} + \frac{a^2 j^2 \Delta t^2 + 3aj\Delta t}{2} \quad \text{Ekvation 9}$$

Om metoden enligt figur 1c istället används ges följande lösningar:

$$p_u = \frac{7}{6} + \frac{a^2 j^2 \Delta t^2 - 3aj\Delta t}{2} \quad \text{Ekvation 10}$$

$$p_m = -\frac{1}{3} - a^2 j^2 \Delta t^2 + 2aj\Delta t \quad \text{Ekvation 11}$$

$$p_d = \frac{1}{6} + \frac{a^2 j^2 \Delta t^2 - aj\Delta t}{2} \quad \text{Ekvation 12}$$

Bilaga 2

Sannolikhetsberäkningar i den anpassade algoritmen

Som i fallet med standardalgoritmen väljs sannolikheterna så att de motsvarar den förväntade förändringen och dess varians under följande tidsperiod. I detta fall används ränteförändringens förväntade värde under en period som ger upphov till ekvation 1 nedan samt förändringen under två perioder som ger upphov till ekvation 2. De två ekvationerna leder tillsammans med det faktum att sannolikheternas summa ska vara ett till ekvationssystemet bestående av ekvation 1 –3 nedan.

$$j\Delta x_i + M = k\Delta x_{i+1} + (p_u - p_d)\Delta x_{i+1} \quad \text{Ekvation 1}$$

$$V + (j\Delta x_i + M)^2 = k\Delta x_{i+1}^2 + 2k(p_u - p_d)\Delta x_{i+1}^2 + (p_u - p_d)\Delta x_{i+1}^2 \quad \text{Ekvation 2}$$

$$p_u + p_m + p_d = 1 \quad \text{Ekvation 3}$$

Genom att lösa ekvationssystemet erhålls ekvationer för de olika sannolikheterna enligt ekvation 4-6:

$$p_u = \frac{V}{2\Delta x_{i+1}^2} + \frac{\alpha^2 + \alpha}{2} \quad \text{Ekvation 4}$$

$$p_m = \frac{V}{2\Delta x_{i+1}^2} + \frac{\alpha^2 - \alpha}{2} \quad \text{Ekvation 5}$$

$$p_d = 1 - \frac{V}{\Delta x_{i+1}^2} - \alpha^2 \quad \text{Ekvation 6}$$

I ekvationerna ovan ges α ur ekvation 7 nedan.

$$\alpha = \frac{j\Delta x_i + M - k\Delta x_{i+1}}{\Delta x_{i+1}} \quad \text{Ekvation 7}$$