

Mat-2.177 – Operaatiotutkimuksen projektityöseminaari
Loppuraportti, 25. huhtikuuta 2002

Velanhinnoittelumallin kalibrointi

Asiakas: Sampo Oyj

Projektityöryhmä:

Projektipäällikkö	Pekka Mild	50124C
	Jaakko Dietrich	47095R
	Juha Ojala	50132M
	Antti Punkka	48484T

Sisältö

1	Johdanto	2
2	Kalibrointimalli	2
2.1	Luottoluokitukset ja niiden vaikutus lainan hintaan	2
2.2	Kalibrointimallin teoria	3
3	Credit Default Swapin hinnoittelun teoria	6
4	Ohjelman toiminta kokonaisuutena	7
4.1	Ohjelman rakenne	8
4.2	Lähtötietojen lukeminen ja tarkistaminen	9
4.3	Tulostus	13
5	Laskennan toteutus	13
5.1	Matriisilaskenta	13
5.2	Muuttujatyypit	14
5.3	calibrate-funktio	15
5.4	cdspricing-funktio	16
6	Yhteenveto	17
7	Lähteet	18

1 Johdanto

Luottoriskimallinnus on viimeisten vuosien aikana noussut kuumaksi tutkimus- ja sovellusalaaksi rahoitusmaailmassa. Tarve erilaisten johdannaisinstrumenttien ja sopimuskokonaisuuksien hinnoittelulle on kasvanut räjähdysmäisesti alan markkinoiden laajentumisen myötä. Perinteisestä historiadataan nojaavasta lähestymistavasta on luovuttu, ja jatkuvasti kehitetään uusia muuttuvaa markkinadataa hyödyntäviä malleja.

Projektimme tavoitteena oli tällaisen markkinoihin kalibroitavan velanhinnoittelumallin rakentaminen Excel-sovellukseksi. Mallin kalibroitiosion teoria oli valmiiksi annettu, ja tältä osin tehtävänä oli toimivan ja helppokäyttöisen makron kirjoittaminen. Microsoft Excel on asiakkaan organisaatiossa yleisimmin käytetty sovelluspohja, joten käyttöliittymä tuli rakentaa Exceliin. Projektin toisena osana oli yksinkertaisen luottojohdannaisen, *credit default swapin*, hinnoittelua tukevan työkalun rakentaminen. Hinnoittelun teoria perustuu kalibroitivaiheen tuloksiin, mutta tarkan mallin rakentaminen ja toteuttaminen jätettiin projektiryhmän tehtäväksi. Myös hinnoitteluosiosta toteutettiin helppokäyttöisellä käyttöliittymällä varustettu Excel-makro.

Projektin kokonaiskesto sijoittuu ajalle 30.01.2002 – 26.04.2002, josta varsinainen ohjelmointi- ja mallinnustyö toteutettiin helmi-maaliskuussa 2002.

Tämä projektin loppuraportti on ryhmitelty seuraavasti. Kappale 2 toimii johdantona käsiteltävään aiheeseen ja esittelee kalibroitiosion teorian. Kappaleessa 3 esitellään ja perustellaan hinnoitteluosiioon valitsemamme ratkaisumallin teoria. Kappale 4 kuvaa toteutetut ohjelmat kokonaisuutena. Kappale 5 käsittelee ohjelmointia Visual Basic for Applications -ohjelmointikielellä, sekä esittelee kappaleiden 2 ja 3 malleista rakennetut ohjelmasovellukset. Kappaleessa 6 luonnehditaan lyhyesti projektin etenemistä sekä arvioidaan saavutettuja tuloksia.

2 Kalibroitimalli

2.1 Luottoluokitukset ja niiden vaikutus lainan hintaan

Luottoluokituksilla mitataan yrityksen tai yhteisön velan takaisinmaksukykyä. Maailmanlaajuisesti tunnettuja luokituksia ylläpitävät mm. Moody's ja S&P. Yrityksen, jolla on korkea luokitus, katsotaan pystyvän suoriutumaan velvoitteestaan paremmin kuin yritysten, joiden luokitus on alhaisempi. Luokitus voi muuttua periodista toiseen, esim. jos yrityksen luokitus on vuonna k AA, voi se olla A vuonna $k+1$. Alhaisemman luottoluokituksen oletetaan tarkoittavan korkeampaa konkurssitodennäköisyyttä. Historiallisesta datasta on mahdollista estimoida todennäköisyyksiä luottoluokituksen muuttumiselle mistä tahansa määritellystä luokasta s mihin tahansa luokkaan s' hetkien k ja $k+1$ välillä yksinkertaisesti tutki-

mallalla todennäköisyyttä $P(\text{luokitus on } s' \text{ hetkellä } k+1 \mid \text{luokitus on } s \text{ hetkellä } t)$, joka voidaan laskea tilastoista frekventistisen tulkinnan mukaan. Näiden todennäköisyyksien perusteella voidaan luokituksen muuttumisen dynamiikkaa mallintaa Markovin ketjuna. Ketjun tiloina ovat luottoluokitusluokat ja siirtymätodennäköisyydet määritellään kuten edellä. Ketjun absorboivana tilana on konkurssi.

Lainan hinnoittelussa arvioidaan asiakkaan takaisinmaksukyky lainan eräännytyessä. Lainanantajan näkemykseksi asiakkaan maksukykyvystä voidaan ajatella asiakkaan luottoluokitus, joka edellä esitetyn Markovin ketjun mielessä määrää asiakkaan maksukykyyn eräpäivänä. Asiakas siis ajatellaan maksukykyiseksi, jos se ei ole konkurssissa. Täten korkeamman luottoluokituksen yrityksille ovat lainat edullisempia eli korot matalampia, alhaisemmasta konkurssitodennäköisyydestä (luottoriskistä) johtuen.

Riskillisiä velkakirjoja voidaan siten hinnoitella luottoluokituksiin perustuen. Koska velkojen takaisinmaksuaika on tyypillisesti pitkä, täytyy hinnoittelussa huomioida myös luokituksen muutosprosessi velan juoksuaikana. Muutoksen mallintamiseen käytetään Markovin ketjua. Jotta Markovin ketju toimisi luotettavana mallina, tulee ketjun siirtotodennäköisyyksien pyrkiä vastaamaan todellista tilannetta. Historiadatasta estimoidut todennäköisyydet kuvaavat pitkän ajan keskiarvoja, joissa erilaiset suhdannevaihtelut tms. trendit ovat tasoittuneet. Reaaliaikainen markkinakaupankäynti kuitenkin reagoi jatkuvasti vallitseviin olosuhteisiin ja lisäksi heijastaa markkinoiden näkemystä tulevasta. Luokitusprosessin parametreja saadaankin päivitettyä kalibroimalla näitä vallitsevan markkinatilanteen mukaisesti.

2.2 Kalibrointimallin teoria

Kehittämämme Microsoft Excel -sovellus perustuu Jarrow-Lando-Turnbull -malliin (JLT) (Jarrow, Lando ja Turnbull, 1997) sekä sen laskennalliset ongelmat ratkaiseeseen versioon (Kijima ja Komoribayashi, 1998) (KK). JLT:ssä esitetään malli Markovin ketjuna mallinnetun luottoluokitusprosessin parametrien estimointiin markkinoilla vaihdettujen joukkovelkakirjojen avulla.

Luottoluokitusprosessin Markovin ketjun (yhden periodin aikahomogeneiseen) siirtotodennäköisyysmatriisiin viitataan Q :lla, jonka jokainen vaakarivi vastaa luottoluokitusta tarkastellulla ajanhetkellä (siis lähtötilaa). Riveistä ylin vastaa korkeinta luottoluokitusta ja alin konkurssitilaa. Konkurssi on myös ketjun absorboiva tila. Pystyivät vastaavat todennäköisyyksiä, millä ko. luottoluokitukseen (tilaan) päädytään. Merkitään lisäksi luottoluokitusten ja konkurssitilan muodostamaa joukkoa $N = \{1, \dots, K + 1\}$, jotka siis tässä ovat Markovin ketjun tiloja. Siirtymätodennäköisyysmatriisi voidaan siis kirjoittaa

$$Q = \begin{pmatrix} q_{11} & q_{12} & \dots & q_{1K} & q_{1,K+1} \\ q_{21} & q_{22} & \dots & q_{2K} & q_{2,K+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ q_{K1} & q_{K2} & \dots & q_{KK} & q_{K,K+1} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \quad (1)$$

missä siis

$$q_{ij} = P(X_{t+1} = j | X_t = i), \quad i, j \in N, \quad t = 0, 1, 2, \dots$$

ja luottoluokituksia on K kappaletta. Tila $K+1$ vastaa siis konkurssia. Q :n alkiot ovat tyypillisesti historiasta estimoituja ja askelpituutena käytetään yhtä vuotta.

Koska tavoitteena on hinnoitella riskillisiä joukkovelkakirjoja, on edellä esitetty stokastinen luokitusprosessi määriteltävä modernin arbitraasiteorian mukaan riskineutraalin todennäköisyysmitan $\tilde{P}$ alla. Koska tämä prosessi ei välttämättä ole aikahomogeeninen, on tarkasteltava siirtymätodennäköisyyksiä ajan funktiona. Siirtymätodennäköisyyksiksi saadaan siis

$$\tilde{q}_{ij}(t, t+1) = \tilde{P}(\tilde{X}_{t+1} = j | \tilde{X}_t = i), \quad i, j \in N, \quad t = 0, 1, 2, \dots,$$

missä $\tilde{X}$ on siis tarkasteltava stokastinen prosessi riskineutraalien todennäköisyyksien alla. Riskitöntä korkoa merkitään $r(t)$. Siirtymätodennäköisyyksiä $\tilde{q}_{ij}(t, t+1)$ vastaavaa, ajasta riippuvaa yhden periodin transitiomatriisia merkitään $\tilde{Q}(t, t+1)$. Ajasta riippuva ratingprosessi ja riskittömän koron muutosprosessi oletetaan toisistaan riippumattomiksi riskineutraalin todennäköisyysmitan $\tilde{P}$ alla.

Merkitään riskittömän, hetkellä T maturoituvan, joukkovelkakirjan markkinahintaa hetkellä t $v_0(t, T)$. Vastaavasti merkitään luottoluokitusluokkaan j , $j \in N$, kuuluvan joukkovelkakirjan hintaa hetkellä t $v_j(t, T)$. Olkoon δ palautustaso, eli se osuus lainasta (%), jonka lainanantaja uskoo saavansa takaisin konkurssipesästä.

JLT:n mukaan riskillisen ja riskittömän joukkovelkakirjan hinnoille saadaan yhteys

$$v_j(t, T) = v_0(t, T)[\delta + (1 - \delta)(1 - \tilde{q}_{j,K+1}(t, T))], \quad (2)$$

missä $j \in N$. Yhtälön (2) termi $(1 - \tilde{q}_{j,K+1}(t, T))$ on siis Markovin ketjun *survival probability*, eli todennäköisyys että tilasta j ei päätytä konkurssiin aikavälillä (t, T) .

Siirtymätodennäköisyydet yleisessä muodossa ovat

$$\tilde{q}_{ij}(t, t+1) = \pi_{ij}(t)q_{ij}, \quad i, j \in N, \quad (3)$$

missä q_{ij} :t siis vastaavat alkuperäisen aikahomogeenisen Markovin prosessin siirtymätodennäköisyyksiä. Positiiviset muuntokertoimet $\pi_{ij}(t)$:t ovat riskipreemiosopeutuksia, jotka ovat tulkittavissa riskin markkinahinnoiksi. Nämä kalibrointikertoimet lasketaan markkinoilla vaihdettujen velkakirjojen hintoihin pohjautuen. JLT:ssä esitetään riskipreemioiden ratkaiseminen yhtälöistä (2) ja (3), mutta tämän menettelytavan käyttäminen johtaa laskennallisiin vaikeuksiin korkean luokituksen (ja siten alhaisen kaatumistodennäköisyyden) lainoja tarkasteltaessa. KK:ssa esitetään kerrointen laskemiseen toinen menettelytapa, jota projektissammekin käytettiin.

Oletetaan, että

$$\pi_{ij} = l_i(t), \quad (4)$$

kaikille $j \neq K + 1$ ja $l_i(t)$:t ovat deterministisiä ajan funktioita. Näin sopeutuskerroin riippuu vain lähtötilasta i ja ajanhetkestä t . Nyt voidaan määritellä uudestaan Markovin ketju $\tilde{X}$, jonka siirtymätodennäköisyydet ovat

$$\tilde{q}_{ij}(t, t + 1) = \begin{cases} l_i(t)q_{ij}, & j \neq K + 1 \\ 1 - l_i(t)(1 - q_{i,K+1}), & j = K + 1 \end{cases}. \quad (5)$$

Alkuperäinen, historiasta estimoitu aikahomogeeninen siirtymätodennäköisyysmatriisi Q voidaan kirjoittaa kanoniseen muotoon

$$Q = \begin{pmatrix} A & R \\ O' & 1 \end{pmatrix}, \quad (6)$$

missä A sisältää siirtymätodennäköisyydet ei-absorboivien tilojen välillä, O' on vaakavektori, jossa on K nollaa ja R sisältää absorbtiotodennäköisyydet (kaatumistodennäköisyydet) kustakin tilasta. Lisäksi on huomioitava, että Markovin ketjun transitiomatriiseissa rivisummat ovat 1, jolloin väistyvien tilojen matriisin A j :s rivisumma voidaan laskea erotuksena $(1 - q_{j,K+1})$. Yhtälön (5) mukaan modifioidulle ketjulle $\tilde{X}$ saadaan matriisimuodossa

$$\tilde{A}(t, t + 1) = L_D(t)A$$

$$\tilde{R}(t, t + 1) = e - L_D(t)Ae, \quad (7)$$

missä $L_D(t)$ on diagonaalimatriisi, jonka diagonaalialkiot ovat $l_j(t)$:t ja e on pystyvektori, jonka alkiot ovat ykkösiä. Markovin ketjuissa useamman askeleen kumulatiiviset siirtotodennäköisyydet voidaan laskea seuraavasti

$$\tilde{A}(0, t + 1) = \tilde{A}(0, t)\tilde{A}(t, t + 1), \quad t = 0, 1, 2, \dots, \quad (8)$$

Nyt voidaan yhtälöiden (2), (5), (7) ja (8) avulla kalibrointikertoimet $l_j(t)$ laskea riskillisten joukkovelkakirjojen markkinahinnoista $v_j(0, t)$, $j \in 0 \cup N$,

$$l_j(t) = \frac{1}{1 - q_{j,K+1}} \sum_{k=1}^K \tilde{q}_{jk}^{-1}(0, t) \frac{v_k(0, t) - \delta v_0(0, t)}{(1 - \delta)v_0(0, t)}, \quad j \in N, \quad (9)$$

missä $\tilde{q}_{ij}^{-1}(0, t)$:t ovat matriisin $\tilde{A}^{-1}(0, t)$ komponentteja. Matriisin $\tilde{A}^{-1}(0, t)$ oletetaan siis olevan kääntyvä kaikilla ajanhetkillä ja $\tilde{A}(0, 0)$ on identiteettimatriisi. Erityisesti ajanhetkelle $t = 1$ määritellään

$$l_j(0) = \frac{1}{1 - q_{j,K+1}} \frac{v_k(0, 1) - \delta v_0(0, 1)}{(1 - \delta)v_0(0, 1)}.$$

Matriisimuodossa riskipreemiosopeutukset saadaan vektorin $L(t) = L_D(t)e$ avulla.

$$L(t) = R_D^{-1} \tilde{A}^{-1}(0, t)b(t+1), \quad t = 0, 1, 2, \dots, \quad (10)$$

missä R_D on diagonaalimatriisi, jonka alkiot ovat $a_{jj} = 1 - q_{j,K+1}$ ja $b(t) = \tilde{A}(0, t)e$.

3 Credit Default Swapin hinnoittelun teoria

Credit Default Swap on luottojohdannainen, jossa johdannaisen ostaja efektiivisesti vakuuttaa antamansa lainan. Merkitään lainan hankkineen yrityksen tilaa

$$DS(t) \equiv \begin{cases} \overline{D}, & \text{jos ei konkurssissa hetkellä } t, \\ D, & \text{jos konkurssi on tapahtunut viimeistään hetkellä } t. \end{cases} \quad (11)$$

Credit Default Swapin ostajan kassavirrat, $F(t)$, ovat (Cheng, 2001)

$$F(t) = \begin{cases} -S & \text{jos } DS(t) = \overline{D}, \\ 1 - \delta_n & \text{jos } DS(t) = D \text{ ja } DS(t-1) = \overline{D}, \\ 0 & \text{muuten,} \end{cases} \quad (12)$$

missä $1 \leq t \leq T$. T on viimeinen hetki, johon liittyy kassavirta, ja maturiteetille T' pätee $T' \geq T$. Maksu S toimii siis swapin ostajalle toistuvana vakuutusmaksuna, ja konkurssin tapahtuessa swapin myyjä sitoutuu korvaamaan ostajalle konkurssista aiheutuvat kustannukset.

Swap voidaan hinnoitella käyttämällä riskineutraaleja todennäköisyyksiä. On siis laskettava maturiteettiin asti kaatumistodennäköisyydet jokaisen periodin lopussa. Yleisesti todennäköisyys, että konkurssi tapahtuu ajanhetkien $t-1$ ja t välissä, kun tila alussa on i , on

$$P_i(\text{DS}(t) = D | \text{DS}(t-1) = \overline{D}) = \sum_{j=1}^K q_{ij}^{t-1} q_{j,K+1},$$

missä q_{ij}^k on matriisin Q^k alkio (i, j) . Koska kuitenkin olemme kiinnostuneita riskineutraaleista todennäköisyyksistä, käytämme edelliselle lausekkeelle muotoa

$$P_i(t) = \tilde{P}(\text{DS}(t) = D | \text{DS}(t-1) = \overline{D}) = \tilde{q}_{i,K+1}(0, t) - \tilde{q}_{i,K+1}(0, t-1), \quad (13)$$

missä $\tilde{q}_{i,j}(0, t)$ on kalibroituvaiheessa lasketun matriisin $\tilde{Q}(0, t)$ alkio. Koska yhtälön (8) mukaan $\tilde{Q}(0, t_n) = \tilde{Q}(0, t_1)\tilde{Q}(t_1, t_2) \cdots \tilde{Q}(t_{n-1}, t_n)$, saadaan $\tilde{q}_{ij}(0, t_s)$ ratkaistua halutulle t_s kalibroitukertomien l ja matriisin Q avulla (ks. yhtälö (7)). Hinnoittelussa ollaan kiinnostuneita riskineutraalista odotusarvosta, joka siis lasketaan kassavirtojen odotusarvona aikaperiodien yli diskontaten riskittömällä korolla $r(t)$. Jos konkurssi tapahtuu hetkellä t , ovat tähän tapahtumaan kohdistuvat kassavirrat $-S$ hetkellä $1, \dots, t-1$ ja $1-\delta$ hetkellä t . Täten odotusarvon nykyarvoksi saadaan

$E(\text{kassavirtojen nykyarvo}) =$

$$\sum_{t=1}^T P_i(t) \left[(1-\delta) \frac{1}{(1+r(t))^t} - \sum_{j=1}^{t-1} S \frac{1}{(1+r(j))^j} \right] - P_i(T+1) \sum_{j=1}^T S \frac{1}{(1+r(j))^j} \quad (14)$$

Kaavassa (14) keskimmäisen summan summausindeksi kulkee arvosta 1 arvoon 0, kun $t = 1$. Ajatellaan tämä termi nollassi. Lisäksi $P_i(T+1)$ tarkoittaa todennäköisyyttä, jolla kaatumista ei ole tapahtunut hetkeen T mennessä. Arbitraasivapaassa riskineutraalissa hinnoittelussa S valitaan hinnoittelun helppouden vuoksi siten, että kassavirtojen nykyarvon odotusarvo on nolla, joten yhtälöstä (14) saadaan velkakirjalle, jonka lähtöluokitus on i ja maturiteetti T

$$S_{iT} = \frac{(1-\delta) \sum_{t=1}^T \left[\frac{P_i(t)}{(1+r(t))^t} \right]}{\sum_{t=2}^{T+1} \left[P_i(t) \sum_{j=1}^{t-1} \left[\frac{1}{(1+r(j))^j} \right] \right]}. \quad (15)$$

Näin kaavan (15) avulla saadaan laskettua $(k \times T)$ hintamatriisi, jossa on hinnat kullekin lähtöluokitukselle ja kullekin maturiteetille.

4 Ohjelman toiminta kokonaisuutena

Projektitehtäväksi asetettiin kappaleiden 2 ja 3 mallien mukaisten sovellusten rakentaminen Microsoft Exceliin. Kalibroitiosion sovellukseen määriteltiin tarkemmin, että

1. Sovellus ottaa parametreinä aikahomogeenisen tilansiirtomatriisin Q , korkojen termiinirakenteen kullekin lähtötilalle (siis yksi matriisi, dimensiot k ja T , riskittömän korkorakennevektorin r_0 sekä palautustaso-olettaman δ).
2. Käyttäjä syöttää lähtödatan Excelin työkirjan soluihin, ja tulokset palautetaan työkirjan soluihin.
3. Sovellus on käsiteltävässä kontekstissa dimensiovapaa, eli syötteenä voi antaa erikokoisia matriiseja (vaihteleva määrä luottoluokituksia ja maturiteettitietä). Kontekstissa relevantit ylärajat dimensioille ovat luokkaa 30×30 .

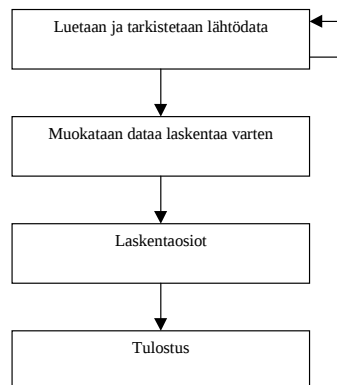
Kalibrointiosion tulee tulostaa kaavan (10) mukainen korjauskerroinmatriisi $L(t)$. Projektin edetessä asetettiin tulostettavaksi myös kalibroidut kumulatiiviset transitiomatriisit $\tilde{Q}(0, t)$, $t = 1, \dots, T$. Hinnoitteluosiosta tulostetaan kaavan (15) avulla muodostuva hintamatriisi S .

Sovelluksen käyttöliittymäksi oli siis asiakkaan puolelta määritelty Microsoft Excel. Exceliin integroitu ohjelmointikieli Visual Basic for Applications (VBA) otettiin käyttöliittymän toteutuksen lähtökohdaksi. Myös laskentaosioden ohjelmakoodit päädyttiin toteuttamaan VBA:lla (Kappale 5.1 käsittelee matriisilaskentaa VBA-ohjelmointikiellä). Näin kaikki ohjelmakoodi saadaan sisältymään yhteen Excel-tiedostoon.

Projektin tuloksena asiakkaalle toimitettava ohjelma on yksi Excel-tiedosto, joka sisältää molempien osioiden makrot (käynnistettävä erikseen) sekä valmiiksi nimetyt tarvittavat tulostusalueet.

4.1 Ohjelman rakenne

Ohjelman molempien osioiden perusrakenne esitetään kuvassa 1.



Kuva 1: Ohjelman toiminta

Lähtöarvot luetaan ja tarkistetaan soluviittauksina. Hyväksynnän jälkeen soluviittausten osoittama data muunnetaan laskentaa varten matriisimuuttujiksi. Laskentaosiot käsittelevät ja palauttavat matriisimuuttujia, joiden sisältämä tieto puretaan tulostusvaiheessa laskentataulukon soluihin.

Tehtävän toteutus sisältää siis kaksi erillistä makroa, jotka ovat kuitenkin toiminnaltaan varsin samanlaisia. Lisäksi hinnoittelumakro `cal_cdspricing` vaatii syötteekseen kalibrointimakron `cal` tuottaman korjauskerroinmatriisin. Objektien kutsuhierarkia on kuvattu taulukossa 1 (varsinaisesti tarkistusfunktioita kutsuu `UserForm` -komponentti). Objekteja on käsitelty tarkemmin viimeisen sarakkeen osoittamassa kappaleessa.

Taulukko 1: Makrojen kutsurakenne

Makro (aliohjelma)	Kutsuu	Kutsuttavan kuvaus	Kappale
cal	InputForm-komponentti	Käyttöliittymän dialog-box	4.2
	tarkistusfunktio	Syötteen tarkistus	4.2
	rngToArray-funktio	Tietotyyppimuunnos	5.2
	calibrate-funktio	Kalibroinnin laskenta	5.3
cal_cdspricing	InputFormCdspr	Käyttöliittymän dialog-box	4.2
	tarkistusfunktio	Syötteen tarkistus	4.2
	rngToArray-funktio	Tietotyyppimuunnos	5.2
	cdspricing-funktio	Hinnoittelun laskenta	5.4

Viitaten kuvaan 1, vaiheet 1-3 toteutetaan itsenäisillä (makrosta kutsuttavilla) kokonaisuuksilla. Tulostusvaihe sisältyy suoraan ajettavaan makroon. Yhteensä ohjelmassa on koodia noin 800 riviä.

4.2 Lähtötietojen lukeminen ja tarkistaminen

Lähtötiedot syötetään suoraan taulukon "lähtödata" soluihin. Lähtötietojen lukemiseen käytetään `UserForm` -käyttöliittymäkomponenttia. Molempien osioiden `UserForm`it ovat toiminnaltaan samanlaisia. Kuvassa 2 esitetään kalibrointiosion `UserForm`in ulkoasu.

Ohjelma toimii seuraavasti.

1. Syötetään lähtötiedot taulukon "lähtödata" soluihin.
2. Käynnistetään makro `Tools`-valikosta, jolloin `UserForm` -komponentti tulee näkyviin.
3. Viedään kursori `UserForm`in kenttään, jonka solun arvo aiotaan seuraavaksi osoittaa (esim. "Tilatodennäköisyysmatriisi").

Kuva 2: Tietojen syöttäminen

4. Maalataan laskentataulukosta osoitettava solualue hiirellä. Maalattu alue ilmestyy soluviittauksena UserFormin edellä aktivoituun kenttään (esim. \$B\$2:\$K\$10).
5. Palataan vaiheeseen 3, kunnes kaikki lähtötiedot on osoitettu.
6. Kun kaikki lähtötiedot on osoitettu, painetaan OK-painiketta. UserFormista voidaan poistua (ja makron ajo keskeyttää) painamalla Peruuta-painiketta.

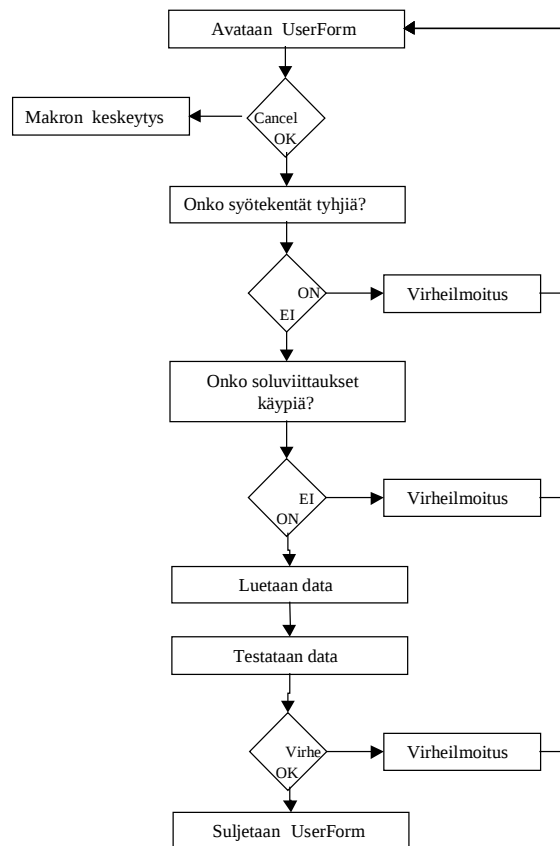
Jos kaikki syötteet ovat käypiä, ei käyttäjältä vaadita muita toimenpiteitä, jolloin OK-painikkeen painaminen ajaa makron loppuun, ja tulokset ilmestyvät näkyviin (katso kappale 4.3).

Taustalla pyörivä makro etenee kuitenkin datan tarkistusvaiheeseen. Tarkistuksella osoitetaan käyttäjälle lähtötiedoissa olevat mahdolliset huolimattomuusvirheet tai puutteet, ja toisaalta estetään ohjelmakoodin kaatumiseen johtavan virheellisen datan pääseminen laskentavaiheeseen.

Ensisijaisesti syötteestä tarkistetaan, että se on tehtävän kannalta relevanttia (katso taulukko 2). Lähtötietoja syöttäessään käyttäjän tulee huomioida, että a) tilatodennäköisyysmatriisiin tulee olla täydellinen, eli myös konkurssitilan rivi ja sarake tulee sisältyä osoitettavaan solualueeseen, b) transitiotodennäköisyydet sekä kaikki korot annetaan *prosentteina*, s.e. korko 1% = 100 bps syötetään lukuna 1.00, ja siirtotn. 95% lukuna 95.00, c) riskitön korkokanta syötetään vaakavek-

torina ja d) palautustaso-olettama syötetään työkirjan soluun ja osoitetaan UserFormissa soluviittauksena.

Syötteen testauksen toimintalogiikka esitetään kuvassa 3.



Kuva 3: Ohjelman suorittaman virheentarkistuksen toiminta

Ensimmäisenä siis testataan, että kaikki vaadittava syöte on osoitettu ja että se on teknisesti oikeassa muodossa. Tämän jälkeen data luetaan "sisään" testausvaiheeseen. Datan testaukseen käytetyt funktiot esitetään kutsumisjärjestyksessä taulukossa 2. Jos jokin testifunktio löytää syötteestä virheen, annetaan käyttäjälle virheilmoitus eikä jäljellä olevia testejä suoriteta. Virheilmoituksen yhteydessä UserForm tulee näkyviin ja virheellinen solualue aktivoituu. Näin käyttäjän on helppoa korjata virhe osoittamalla virheellinen solualue uudelleen tai muuttamalla osoitetun solualueen sisältöä. Kun virhe on korjattu, käyttäjä painaa OK-painiketta, ja datan testaus lähtee alusta liikkeelle.

Kun syötedata on läpäissyt kaikki testit, ohjelma etenee laskentavaiheeseen (katso kappale 5).

Taulukko 2: Lähtötietojen testausfunktiot

UserForm Testifunktio	Kuvaus
<i>InputForm</i>	
IsQOk	Testaa osoitetusta transitiomatriisista Q , että 1) solualue on yhtenäinen 2) Q on neliömatriisi 3) matriisialkiot ovat ei-negatiivisia numeroita 4) rivisummat kuuluvat välille $[99, 101]$ 5) konkurssitila on absorboiva Antaa virheen löytyessä virhetilaa kuvaavan virheilmoituksen
IsTSOk	Testaa osoitetusta bondien korkorakenteesta, että 1) solualue on yhtenäinen 2) luottoluokitusten lukumäärä vastaa matriisin Q dimensiota 3) matriisialkiot ovat positiivisia numeroita Antaa virheen löytyessä virhetilaa kuvaavan virheilmoituksen
Isr0Ok	Testaa osoitetusta riskittömän korkokannan vektorista, että 1) solualue on yhtenäinen 2) vektorin dimensio on $1 \times n$, missä n vastaa bondien korkorakenteen aikajännettä Antaa virheen löytyessä virhetilaa kuvaavan virheilmoituksen
IsdOk	Testaa osoitetusta palautustasosta, että 1) solualue viittaa yksittäiseen soluun 2) annettu syöte on luku väliltä $[0, 1]$ Antaa virheen löytyessä virhetilaa kuvaavan virheilmoituksen
<i>InputFormCdspr</i>	
IsQOk	ks. yllä
IsLOk	vastaa funktiota IsTSOk
Isr0Ok2	vastaa funktiota Isr0Ok
IsdOk	ks. yllä

4.3 Tulostus

Ohjelman laskentaosiot palauttavat tuloksenaan matriisimuuttujia, joiden tiedot (alkiot) täytyy purkaa laskentataulukon soluihin käyttäjän luettavaksi. Käytännössä taulukkoon tulostus on toteutettu s.e. 1) osoitetaan tulostustaulukosta tulostusalue soluviittauksena, ja 2) kirjoitetaan matriisimuuttujien tiedot alueen soluihin alkio kerrallaan For-silmukoiden avulla. Siis tulostusalueen kuhunkin soluun kirjoitetaan laskennan tuloksena saadun matriisin vastaavan alkion arvo. Ohjelmakoodina sijoitus on muotoa:

```
Dim alue as Range
Set alue = Range("soluviittaus")
alue.Cells(i,j).value = L(i,j)
```

Tulostusalueiden koot on asetettu siten, että ohjelma pystyy käsittelemään tilanteet, joissa luottoluokitusten ja eri maturiteettien tulo ei ylitä arvoa 10000 Tulostuksen(kin) puolesta ohjelma täyttää asetetun dimensiovapausvaatimuksen.

Käyttöliittymä tulostaa korjauskerroinmatriisin L yhteen taulukkoon ("korjauskertoimet") ja kaikki kumulatiiviset tilansiirtomatriisit allekkain yhteen taulukkoon ("transitiomatriisit"). Hinnoittelumakron tuottama hintamatriisi tulostetaan jälleen erilliseen taulukkoon ("CDShinnat"). Näin ohjelmassa on yhteensä neljä erillistä laskentataulukkoa, joista kolme toimii tulostusalueina.

5 Laskennan toteutus

Tässä kappaleessa käydään läpi ohjelmoinnin toteutuksen kannalta oleellisimpia yksityiskohtia. Tällaisia ovat matriisilaskennassa käytetyt Excelin työkirjafunktiot (WorksheetFunction) sekä muuttujien nimeämisessä ja tyyppien valinnassa käytetyt periaatteet. Lisäksi esitellään ohjelman laskentafunktioiden toimintaperiaate.

5.1 Matriisilaskenta

Koska VBA:ssa ei ole varsinaisia matriisifunktioita, käytetään matriisilaskentaan Excelin työkirjafunktioita. Näitä voidaan kutsua objektin Application.WorksheetFunction kautta. Laskentaosioden ohjelmoinnissa tarvitaan kahden matriisioperaatiota, käänteismatriisin laskemista sekä matriisituloa. Nämä voidaan suorittaa seuraavilla käskyillä:

```
Application.WorksheetFunction.MInverse(A)
Application.WorksheetFunction.MMult(A,B)
```

Funktioiden argumentit voivat olla joko Range- tai Variant-tyyppisiä taulukoi-
ta. Mikäli matriisioperaatioita tarvitaan peräkkäin useampia, on kätevää käyttää
seuraavanlaista rakennetta:

```
With Application.WorksheetFunction
    .MInverse(A)
    .MMult(A,B)
End With
```

Tehtävän kontekstissa relevantin kokoisilla matriiseilla laskentanopeus on yo.
rakenteita käytettäessä erinomainen. Näin tarvetta laskennan suorittamiseen
matemaattisesti tehokkaammalla kielellä ei ole.

5.2 Muuttujatyypit

Desimaaliluvuille käytettiin aina Double-tyyppisiä muuttujia parhaan mahdol-
lisen laskentatarkkuuden varmistamiseksi. Kokonaisluvuille käytettiin luonnollis-
esti Integer-tyyppisiä muuttujia. Matriiseille jouduttiin useimmiten käyttämään
Variant-tyyppisiä muuttujia, joihin voidaan tallentaa mm. Double-tyyppisiä
taulukkoita. Variant-tyyppiä jouduttiin käyttämään, koska se on ainoa muuttu-
jatyyppi, johon matriiseja käsittelevien työkirjafunktioiden ulostulot voidaan si-
joittaa suoralla sijoituksella. Variant-muuttujaan voidaan myös sijoittaa mieliv-
altaisen kokoinen matriisi; ohjelman käsittelemien matriisien kokoja ei haluttu
kiinnittää ohjelman joustavuuden takaamiseksi.

Excelin työkirjasta soluviittauksena (Range) luettu lähtödata muutetaan
Variant-tyyppiseksi matriisimuuttujiksi itse kirjoitetulla funktiolla rngToArray.
Funktiossa solujen arvo muutetaan Double-tyyppiseksi käyttämällä VBA:n funk-
tiota CDBl. Koska funktio on lyhyt ja kuitenkin melko oleellinen, se esitetään
seuraavassa:

```
Public Function rngToArray(rng As Range) As Variant

    Dim R As Integer 'rivien lukumäärä
    Dim C As Integer 'sarakkeiden lukumäärä
    R = rng.Rows.Count
    C = rng.Columns.Count

    ReDim arr(1 To R, 1 To C) As Variant
    For i = 1 To R
        For j = 1 To C
            'arr(i, j) = Val(rng.Cells(i, j).Value)
            arr(i, j) = CDBl(rng.Cells(i, j).Value)
        Next j
    Next i

    rngToArray = arr
```

End Function

Muuttujien nimeämisessä pyrittiin mahdollisimman paljon noudattelemaan kappaleissa 2.2 ja 3 käytettyä notaatiota. Lisäksi luokitustasojen lukumäärää merkittiin muuttujalla k ja eri maturiteettien lukumäärää muuttujalla T . Riskineutraaleista todennäköisyyksistä käytetty merkintä $\sim$ korvattiin koodissa kirjaimella m (esim. $\tilde{A} \rightarrow Am$). Käänteismatriiseiden $^{-1}$ -merkintä korvattiin kirjaimella i (esim. $R^{-1} \rightarrow Ri$).

5.3 calibrate-funktio

Funktio `calibrate` suorittaa korjauskerroinmatriisin L laskemisen soveltamalla kaavaa (10). L -matriisin lisäksi funktio palauttaa riskineutraalit kumulatiiviset tilansiirtomatriisit ajanhetkestä nolla hetkeen T . Funktion ulostulomatriisissa on siis rinnakkain matriisit L sekä T kappaletta kumulatiivisia tilansiirtomatriiseja (joista puuttuu default-rivi). Funktion määrittely on seuraavanlainen:

```
Public Function calibrate(ByVal Q As Variant, ByVal TS As Variant,
ByVal r0 As Variant, ByVal d As Variant) As Variant

    'Q = tilansiirtomatriisi
    'TS = bondien korkorakenne
    'r0 = riskitön korkokanta, vaakavektori
    'd = palautustaso
```

Funktion toiminta yksinkertaistettuna on seuraavanlainen:

- Riskitöntä korkokantaa $r0$ ja *vuosittaista koron laskua* käyttäen lasketaan vaakavektoriin $v0$ diskonttokertoimet eri maturiteeteille.
- Termiinirakennetta TS ja *vuosittaista koron laskua* käyttäen lasketaan matriisiin V riskilliset diskonttokertoimet eri maturiteeteille.
- Muodostetaan (pysty-)yksikkövektori e .
- Muodostetaan kanonisen esitysmuodon matriisi A matriisista Q .
- Lasketaan matriisi R_D ja sen käänteismatriisi $R_D i$, katso kaava (10).
- Muodostetaan identiteettimatriisi $Am(0, 0)$.
- Muodostetaan L -matriisi rekursiivisesti. Oleellisesti tässä sovelletaan kaavoja (7), (8), (9) ja (10). Tämä on funktion oleellisin kohta, joten sen koodi esitetään seuraavassa:

```

ReDim b(1 To k, 1 To 1) As Double ReDim L(1 To k, 1 To T) As
Variant ReDim apu2(1 To k, 1 To 1) As Variant ReDim apu3(1 To k, 1
To k) As Double

With Application.WorksheetFunction

For u = 1 To T
  For i = 1 To k
    b(i, 1) = (V(i, u) - d(1, 1) * v0(1, u)) /
      ((1 - d(1, 1)) * v0(1, u)) ' KK, kaava (19)
  Next i
  apu2 = .MMult(Ri, .MMult(.MInverse(Am), b))
  For i = 1 To k
    L(i, u) = apu2(i, 1)
    apu3(i, i) = apu2(i, 1)
  Next i
  Am = .MMult(Am, .MMult(apu3, A)) ' KK, kaavat (17) ja (18)
Next u
End With

```

- Jokaiselta iteraatiokierrokselta poimitaan matriisi $A(0, u)$, ja lasketaan tästä $R(0, u)$. Matriisi $A(0, u)$ ja vektori $R(0, u)$ liitetään rinnakkain apumatriisiin.
- Kootaan ulostulomatriisi $\text{calibrate} = [L, A_m(0,1), R(0,1), A_m(0,2), R(0,2), \dots, A_m(0,T), R(0,T)]$, missä vektorit R ($k \times 1$) kuvaavat kumulativisia kaatumistodennäköisyyksiä. Calibrate-matriisissa on siis k riviä ja $T + T \cdot (k + 1)$ saraketta.
- Palautetaan matriisi calibrate .

5.4 cdspricing-funktio

Funktio `cdspricing` suorittaa credit default swapien hintojen (vuosittaisen koron) laskemisen käyttäen kaavaa (15). Laskeminen suoritetaan kaikille eri luottoluokituksille sekä maturiteeteille yhdestä T :hen. Arvonaan funktio palauttaa hinnat sisältävän matriisin. Funktion määrittely on seuraavanlainen:

```

Public Function cdspricing(ByVal Q As Variant, ByVal L As Variant,
ByVal d As Double, ByVal r0 As Variant) As Variant

  'Q = tilansiirtomatriisi
  'L = korjauskerroinmatriisi
  'd = palautustaso
  'r0 = riskitönkorkokanta, vaakavektori

```

Funktion toiminta yksinkertaistettuna on seuraavanlainen:

- Muodostetaan kanonisen esitysmuodon matriisi A matriisista Q .
- Lasketaan riskineutraalit kumulatiiviset kaatumistodennäköisyydet (todennäköisyydet että kaatuminen tapahtuu ennen ajanhetkeä t , $t \leq T$) eri luottoluokituksille ja ajanhetkille
- Lasketaan riskineutraalit kaatumistodennäköisyydet (todennäköisyydet että kaatuminen tapahtuu aikavälillä $[t - 1, t]$, $t \leq T$) eri luottoluokituksille ja ajanjaksoille.
- Riskitöntä korkokantaa r_0 ja vuosittaista koron laskua käyttäen lasketaan riskittömät diskonttokertoimet eri maturiteeteille.
- Lasketaan credit default swapin hinnat S eri maturiteeteille ja luottoluokituksille soveltaen kaavaa (15).
 - Lasketaan tulojen nykyarvon odotusarvo (eli kaavan (15) osoittajan arvo).
 - Lasketaan maksujen nykyarvon odotusarvo olettaen että maksun suuruus $S = 1$ (eli kaavan (15) nimittäjän arvo).
 - Lasketaan hinta S suorittamalla kaavan (15) jakolasku.
- Palautetaan hintamatriisi.

6 Yhteenveto

Kokonaisuutena projekti on ollut erittäin onnistunut. Asiakas on ollut tyytyväinen tuloksiin ja työ on edennyt sujuvasti ilman suurempia ongelmia.

Kalibrointivaiheen toteutuksessa haaste liittyi VBA-ohjelmointikielellä ohjelmointiin. Kappaleen 2.2 malli on saatu onnistuneesti ohjelmoitua Excel-sovellukseksi. Myös ohjelman antamat tulokset ovat erittäin suurella todennäköisyydellä oikein, vaikka täysin aukotonta validointia ei ole pystytty suorittamaan. Tulokset poikkeavat hieman KK:n paperissa esitetyistä, mutta luonnollinen selitys poikkeamalle on ero lähtötietojen riskittömässä korkokannassa. Lisäksi sovelluksemme käyttää vuosittaista koronlaskua jatkuvan sijaan. Ohjelmakoodi kuitenkin noudattaa tarkalleen KK:n mallia, ja ohjelman tuottamia tuloksia on testattu rakentamalla riskillisten velkakirjojen termiinirakenne "takaperin" ohjelman laskemien korjauskertoimien avulla. Uskallammekin sanoa, että kalibrointiosion tuottamat tulokset ovat oikeita.

Hinnoitteluosion laskentamallin ratkaiseminen kuului projektiryhmän tehtäviin. Ratkaisun teoria on esitetty kappaleessa 3, ja vastaava ohjelmatoteutus kappaleessa 5.4. Hinnoitteluosion tuloksia ei ole varsinaisesti päästy validoimaan, koska asiakkaan kanssa on sovittu, että hinnoittelun ratkaisuperiaatetta arvioidaan osin tämän loppuraportin avulla. Ohjelmakoodi noudattaa kappaleen 3 teoriaa, jonka arviointi on siis asiakkaan tehtävä. Projektiryhmämme oman asiantuntemuksen puitteissa uskomme hinnoittelumallin rakentamisen onnistuneen.

Ohjelman käyttöliittymään sekä asiakas että projektiryhmä ovat tyytyväisiä. UserForm -komponentti, ja siten hiirellä ohjailtava käyttöliittymä on helppokäyttöinen ja toimiva. Ohjelman toimintaa on testattu erikokoisilla matriiseilla, ja kaikkien testifunktioiden toiminta on varmistettu antamalla virheellistä syötettä. Tehokkaan lähtödatan testauksen johdosta ohjelma on toiminnaltaan erittäin vakaa. Virheilmoitukset pyrkivät kuvailemaan syntyneen virheen mahdollisimman tarkasti, ja lisäksi virhealue korostetaan aktiivimalla kyseinen alue. Näin ohjelman käyttö ja mahdollisten virheiden korjaaminen on helppoa.

Projektityö on edennyt varsin tarkasti projektisuunnitelman mukaisesti. Myös asetettu aikataulu piti pääosin paikkansa, kuitenkin niin että eri vaiheiden työt sijoittuivat ennakoitua tiiviimmin dead linen läheisyyteen. Alkuperäisessä suunnitelmassa asetimme sisäiseksi tavoitteeksi loppuraportille päivämäärän 19.04. Tähän tavoitteeseen ei päästy, mutta loppuraporttikin valmistui asiakkaan (kurssin) asettamaan päivämäärään 26.04. mennessä. Suunniteltuun työnjakoon verrattuna Juha Ojala ja Jaakko Dietrich tekivät ennakoitua enemmän ohjelmointityötä, ja vastaavasti Pekka Mild ja Antti Punkka kantoivat suurimman vastuun loppuraportin kirjoittamisesta. Kokonaisuutena tehtävät saatiin jaettua varsin järkevästi ja tasapuolisesti. Ennakoidut riskit olivat vähäisiä, eivätkä mitkään riskeistä varsinaisesti realisoituneet projektin kuluessa.

Projektitehtävä ja asetetut tavoitteet olivat kiinnostavia ja mielekkäitä. Myös toteutunut kokonaistyömäärä vastasi hyvin ennakoitua. Projektiryhmän puolesta projektin voidaan katsoa onnistuneen hyvin.

7 Lähteet

1. Cheng, W.-Y. (2001). Recent Advances in Default Swap Valuation. *The Journal of Derivatives*, vol. 9, pp. 8–17.
2. Jarrow, R.A., Lando, D. and Turnbull, S.M. (1997). A Markov Model for the Term Structure of Credit Spreads. *The Review of Financial Studies*, vol. 10, pp. 481–523.
3. Kijima, M., Komoribayashi, K. (1998). A Markov Chain for Valuing Credit Risk Derivatitives, *The Journal of Derivatives*, vol. 6, pp. 97–108.