

Title	BWR-vaikutusalojen laskentajärjestelmä
Author(s)	Rantamäki, Karin; Rätty, Antti
Citation	Tutkimusraportti : VTT-R-00778-13 VTT, 2013, pages 26 s.
Rights	This report may be downloaded for personal use only.

VTT
<http://www.vtt.fi>
P.O. box 1000
FI-02044 VTT
Finland

By using VTT Digital Open Access Repository you are bound by the following Terms & Conditions.

I have read and I understand the following statement:


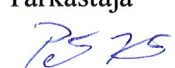
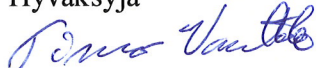
This document is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of this document is not permitted, except duplication for research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered for sale.



BWR-vaikutusalojen laskentajärjestelmä

Kirjoittajat: Karin Rantamäki ja Antti Rätty

Luottamuksellisuus: julkinen

Raportin nimi BWR-vaikutusalojen laskentajärjestelmä		
Asiakkaan nimi, yhteyshenkilö ja yhteystiedot VYR	Asiakkaan viite 5/2012SAF	
Projektin nimi CRISTAL2012/Safir2014	Projektin numero 73800	
Raportin laatija(t) Karin Rantamäki ja Antti Räty	Sivujen/liitesivujen lukumäärä 26/6	
Avainsanat vaikutusalat, BWR, BWRXS, CASMO-4E, CAXMAN, CRFIT, HEXBU-3D, TRAB-3D, APROS	Raportin numero VTT-R-00778-13	
Tiivistelmä <p>Sydänsimulaattorit ja transienttikoodit käyttävät vaikutusaloja kuvaamaan polttoaineen käyttäytymistä mm. lämpötilojen, tiheyksien ja palaman funktiona. Myös systeemikoodit ovat viime aikoina alkaneet siirtyä pistekinetiikasta 1D- tai 3D mallinnukseen, koska näin niihin saadaan tarkempi kuvaus reaktorin käyttäytymiselle erilaisissa tilanteissa.</p> <p>VTT:llä on jo pitkään laskettu materiaalivakioita VVER-440 reaktoreille. Laskenta on pohjautunut CASMO-4/4E-nippupalamaohjelmaan. Materiaalivakioiden laskemista varten on aiemmin kirjoitettu apuohjelmia, joiden avulla vakiot voidaan tuottaa haluttuun muotoon.</p> <p>Tässä työssä nämä ohjelmat on koottu paketiksi. Työssä on keskitytty erityisesti kiehumisvesilaitokselle laskettaviin vaikutusaloihin. Laskentajärjestelmää varten kirjoitettiin perl-skripti BWRXS.pl, joka toimii paketin pääohjelmana. Se huolehtii laskentaketjun läpiviemisestä sekä apuohjelmien syöttötiedostojen tekemisestä. Näin voidaan vähentää käsityötä, kun jokaista välivaihetta ei tarvitse erikseen tehdä käsin.</p> <p>Paketti tarvitsee pohjakseen CASMO:n syöttötiedostot. Jokaiselle nipputyypille pitää olla oma peruspalamalaskun tiedosto. Lisäksi tarvitaan yksi yhteinen kerroinlaskutiedosto. Näistä tiedostoista skripti tekee varsinaisen syöttötiedoston CASMOlle. Lopputuloksena syntyy kullekin nipulle tiedosto, jossa on tarvittavat takaisinkytkentäkertoimien sovituskertoimet HEXBU-muodossa.</p>		
Luottamuksellisuus	julkinen	
Espoo 6.2.2013 Laatija  Karin Rantamäki, erikoistutkija	Tarkastaja  Petri Kotiluoto, johtava tutkija	Hyväksyjä  Timo Vanttola, teknologiapäällikkö
VTT:n yhteystiedot VTT, PL 1000, 02044 VTT		
Jakelu (asiakkaat ja VTT) SAFIR2014/TR3, TK5011		
VTT:n nimen käyttäminen mainonnassa tai tämän raportin osittainen julkaiseminen on sallittu vain VTT:ltä saadun kirjallisen luvan perusteella.		

Sisällysluettelo

1	Johdanto	3
2	Laskentapaketti	4
2.1	BWRXS.pl –skripti	4
2.2	Caxman8.f.....	6
2.3	Crfit8.f	8
3	Käyttöohje	11
3.1	Inputit	11
3.1.1	BWRXS-skriptin input.....	11
3.1.2	CASMOOn inputit.....	12
3.1.3	Caxman8n input	15
3.1.4	Crfiti8n inputit	15
3.2	Rajoitukset/vaatimukset	17
4	Yhteenveto	19
Liite A	BWRXS.pl skripti	21

1 Johdanto

Sydänsimulaattoreissa ja transienttikooodeissa käytetään vaikutusaloja kuvaamaan polttoainepippujen käyttäytymistä mm. lämpötilojen, tiheyksien ja palaman funktiona. Vaikutusalat toimivat tällöin lähtötietona näille kooodeille. Vakioiden käytön etuna on, että näissä kooodeissa käytetään varsinaisia reaktorifysiikkakooodeja yksinkertaisempaa neutroniikkamallia, joka perustuu yksi- tai kolmedimensioiselle noodimallille. Yleisimmin käytettäneen kaksiryhmäisiä vakioita, jotka lasketaan nopealle ja termiselle ryhmälle erikseen. Myös systeemikoodit ovat siirtymässä pistekinetiikasta 1D- tai 3D mallinnukseen, koska näin niihin saadaan tarkempi kuvaus reaktorin käyttäytymiselle erilaisissa tilanteissa.

VTT:llä on jo pitkään laskettu Loviisan sydänsimulaatioita varten materiaalivakioita. Vaikutusalat lasketaan nippukohtaisesti CASMO-4/4E:llä [1,2]. Aikanaan VTT:llä on kirjoitettu apuohjelmat caxman [3] ja crfit [4,5], jotka lukevat ja muokkaavat CASMO-4/4E:n tulostiedosta tarvittavat tiedot. Tuloksena syntyvät materiaalivakiot kirjoitetaan muotoon, jota sydänsimulaattori HEXBU-3D [6,7] ja VTT:llä kirjoitettu dynamiikkaohjelma TRAB-3D [8] ymmärtävät.

Loviisan materiaalivakioiden laskentaa varten tehtiin VTT:llä joitain vuosia sitten paketti, joka helpottaa vakioiden laskemista VVER-440-reaktoreille. Viime aikoina on ollut kasvavaa tarvetta vaikutusaloille myös APROS:n [9] suunnalta, johon on lisätty HEXTRANin [10] sydänmalli. Vaikutusaloja generointia tarvitaan myös BWRn 3D-dynamiikkalaskentaa varten. Tästä syystä on päädytty tekemään paketti kiehutusvesilaitoksen materiaalivakioiden laskemista varten. Työn tarkoituksena oli luoda nykyisessä ohjelmaympäristössä toimiva, helppokäyttöinen laskentaketju sovituskertoimien laskemiseksi kiehutusvesireaktorille.

Pakettia varten kirjoitettiin skripti, joka huolehtii laskentaketjun läpi viemisestä. Siihen kerättiin erityisesti kiehutusvesilaitosta varten kirjoitetut versiot Caxmanista ja Crfitistä. Skripti tekee laskut alkaen CASMOsta, jota varten pitää olla syöttötiedostot. Apuohjelmien syöttötiedostot kirjoitetaan laskennan kuluessa. Lopputuloksena on kullekin polttoainepipulle tehty tiedosto, joka sisältää HEXBU-muotoiset materiaalivakiot. Paketin etuna on, että vakioiden tekeminen on melko helppoa, kun käsityötä on vähennetty, eikä jokaista välivaihetta tarvitse ajaa käsin.

Tässä raportissa on kuvattu laskentapaketin toiminta luvussa 2. Myös työssä käytetyt ohjelmat on kuvattu. Luvussa 3 on annettu skriptin käyttöohjeet. Sekä skriptin että sen käyttämien ohjelmien syöttötiedostojen kuvaukset ja vaatimukset on esitetty tässä luvussa. Lopuksi luvussa 4 on lyhyt yhteenveto työstä.

2 Laskentapaketti

Tämä laskentapaketti on kehitetty nimenomaan kiehutusvesireaktorin nippujen materiaalivaikoiden tuottamiseen TRAB-3Dlle [8] tai Aprosille [9]. Se koostuu yhdestä perl-skriptistä BWRXS.pl ja kolmesta ohjelmasta CASMO [1,2], Caxman8 [3] ja Crfit8 [4]. CASMOa lukuun ottamatta nämä osat on kuvattu ja dokumentoitu tässä luvussa. Paketti on kehitetty VTT:n Linux-klusterilla Ahjo (esptt3m003.ad.vtt.fi).

CASMOlla tarkoitetaan tässä raportissa joko CASMO-4:ää [1], tai sen laajennettua versiota CASMO-4E:tä [2]. Oletusarvoisesti käytetään CASMO-4E:tä. Caxman8:sta tehtiin tämän työn yhteydessä laajennettu versio caxman8rods. Kumpaakin voidaan käyttää, mutta oletusarvoisesti käytetään caxman8rodsia. Yksinkertaisuuden vuoksi tässä raportissa käytetään caxman8-ohjelmasta nimitystä caxman riippumatta versiosta.

2.1 BWRXS.pl –skripti

BWRXS-skripti tekee APROSille tai TRAB-3Dlle vaikutusalat HEXBU-3D muodossa lähtien CASMO-inputeista. Skripti käyttää hyväkseen caxmania ja crfit8aa, jotka on kuvattu luvuissa 2.2 ja 2.3. Tämän työn yhteydessä caxman ja crfit8 on käännetty Linux-pohjaisessa Ahjo-klusterissa Intelin f77-kääntäjällä. Skripti sekä caxman ja crfit8 löytyvät hakemistosta /home/prokmr/bin/XSohjelmia/. Ohjelmalistaus on annettu liitteessä A.

Skripti ottaa argumenttina syöttötiedoston nimen. Syöttötiedoston rakenne on kuvattu luvussa 3.1. Tiedostosta luetaan muun muassa laskettavien nippujen lukumäärä ja niiden perustiedostojen nimet sekä Crfit8:n tarvitsemien kerroinlaskujen kortit sisältävän tiedoston nimi. Näitä tiedostoja kutsutaan tässä raportissa perusnipputiedostoiksi ja kerrointiedostoksi. Lisäksi luetaan laskettavien säätösauvojen määrä sekä laskettavien aukko-osuuksien arvot ja niitä vastaavat tiheydet.

Käytetyt ohjelmat ja niiden versiot määritellään skriptin alussa kohdassa Ohjelmamäärittelyt, jossa annetaan siis ajokelpoiset versiot ohjelmista. Samassa kohdassa määritellään myös polku, josta ne löytyvät. Tässä tapauksessa polku on /home/prokmr/bin/XSohjelmia.

Alussa määritellään muutama CASMO-laskuun tarvittava kortti. Ne on koottu taulukkoon 1. CASMO-laskut tehdään kolmelle aukko-osuudelle, joten VOI kortilla pitää olla nämä arvot annettuna. Lisäksi tällä kortilla määritellään caxmanin inputiin tarvittava aukko-osuus. Tästä syystä VOI-kortin muodon on oltava taulukossa annetun muotoinen. PUN-kortin tulee olla annetun mukainen, jotta *.cax-tiedostoon kirjoitetaan keskimääräinen vuo kussakin mikroalueessa. WRI-kortilla vältetään ylimääräinen *.out-tiedostoon kirjoittaminen.

Taulukko 1: CASMO-laskuun lisättävät kortit ja niiden muodot. Skripti lisää nämä kortit automaattisesti poistaen muut mahdolliset samannimiset kortit.

VOI 0,40,80	* History depletion void (base=40%)
PUN 6*0 1	
WRI 100/'OUT'	

Skripti yhdistää CASMO-ajoa varten kullekin ajettavalle nipulle kerrointiedoston perusnipputiedostoon ja tekee inputin, joka on nimeltään NippuXXXX.inp. XXXX on nelinumeroitu luku, jolla niput identifioidaan. Kaksi ensimmäistä numeroa otetaan perusnipputiedoston nimestä.

Tästä syystä perusnipputiedostojen nimissä pitää olla jaksoluku, jossa on kaksi numeroa esim. e15testi.inp. Tässä e:n perässä oleva luku kuvaa jaksoa. Kolmas numero nipputiedostossa otetaan tiedostojen järjestyksestä. Esimerkiksi jakson 15 kolmelle nipulle syntyvät tiedostot ovat siten Nippu1510.inp, Nippu1520.inp ja Nippu1530.inp.

Perusnipputiedostosta luetaan polttoaineen lämpötila TFU, joka pyöristetään lähimpää kokonaislukuun ja korvataan nipputiedostoon. TFU:n pyöristys kokonaislukuun johtuu siitä, että ketju ei tällä hetkellä mene läpi, jos lämpötila on annettu desimaalilukuna. Syytä tähän ei tässä vaiheessa tiedetä. Epäilynä on, että kyseessä on liukulukuihin liittyvä ongelma joko caxmanissa tai crfit8ssa. Laskentaketjun läpiviemisen varmistamiseksi päädyttiin tässä vaiheessa pyöristämään polttoaineen nominaalitalan lämpötila CASMO-laskuissa lähimpään kokonaislukuun. Skripti hoitaa tämän automaattisesti, joten käyttäjän ei tarvitse huolehtia siitä erikseen.

Kerroinlaskujen tiedostosta luetaan kerroinlaskujen alin ja ylin arvo polttoaineen lämpötilalle. Siksi kyseisellä kortilla pitää olla sana ”Doppler”. Lisäksi tiedostosta luetaan kerroinlaskujen palama-arvot. Arvot luetaan riviltä, joka seuraa ensimmäistä riviä, jolla lukee ”BRANCHES”.

Yhdistettyyn nippu-tiedostoon päivitetään polttoaineen lämpötilojen arvot joka kohtaan. Kerroinlaskujen palamapisteiden arvoja ei kuitenkaan tarkasteta eikä päivitetä. Käyttäjän on itse huolehdittava, että kaikilla kerroinlaskuilla on tarvittavat palamapisteet. Sen sijaan mahdolliset PUN, VOI, WRI kortit korvataan laskujen tarvitsemien mukaisilla korteilla ja perusnipputiedostossa olevat STA ja END kortit poistetaan ennen tiedostojen yhdistämistä.

Syntyneestä nipputiedostosta luetaan nominaalitalan jäähdytteen lämpötila TMO, aukko-osuus ja boori. Tätä varten VOI-kortin pitää olla taulukossa 1 esitetyn muotoinen. Aukko-osuus luetaan kohdan ”base=” perästä. Näitä arvoja tarvitaan caxmanin syöttötiedostoa varten.

Seuraavaksi ajetaan CASMO-ajo ja syntynyt *.cax-tiedosto linkitetään caxmania varten nimelle caxman.cax. Skripti ei kirjoita olemassa olevien CASMO-tiedostojen päälle. Jos tiedostot ovat olemassa, jää ajo roikkumaan, koska CASMO odottaa käyttäjän vastausta kysymykseen tiedostojen käsittelyyn. Tästä syystä kaikki Nippu*. * tiedostot on syytä poistaa hakemistosta ennen laskentapaketin ajamista.

Caxmania varten kirjoitetaan nominaaliarvot syöttötiedostoon. Mikäli tarkoituksena on ajaa vain yksi säätösauvatapaus, skripti toimii myös alkuperäisellä Caxman8-versiolla, mutta oletuksena skripti käyttää myös useamman tapauksen mahdollistavaa Caxman8rods-versiota. Ajon jälkeen skripti huolehtii tulostiedostojen kopioimisesta oikeille nimille, joissa käytetään samaa nimeämiskäytäntöä kuin CASMO:n nipputiedostojen nimissä. Caxmanin input on nimeltään cxXXXX.in, eli esim. cx1510.in ja tulostiedosto vastaavasti cx1510.dat

Lopuksi kirjoitetaan vielä Crfit8:aa varten input. Input on nimeltään crXXXX.in, missä tuo numero on sama kuin aiemmin. Ensin kirjoitetaan perustiedot ja sitten blokit kullekin säätösauvalle ja kullekin aukko-osuudelle. Inputin rakenne on kuvattu tarkemmin luvussa 3.1. Crfit8:n ajon jälkeen kopioidaan tulokset tiedostosta fort.9 perustapauksen mukaan numeroituun tiedostoon crXXXX.mat. Kaikki materiaalivekiot ovat siten *.mat-tiedostoissa.

Skripti ei tarkasta missään vaiheessa, ovatko ajot menneet onnistuneesti läpi. Ellei koko ketju ole mennyt läpi, ei myöskään materiaalivekiot sisältäviä *.mat-tiedostoja synny. Välivaiheena syntyviä tiedostoja ei poisteta, joten epäonnistumisen syyn selvittämisen pitäisi näiden avulla

onnistua. Kun kaikki tiedostot ovat olemassa, pitäisi laskennan edetä kuten taulukossa 2 on esitetty.

Taulukko 2: Laskentaketju pelkistetyssä muodossaan erikseen tehtynä.

```
> cas4e_lib.pl Nippu1510.inp e4l1b70
> ln -s Nippu1510.cax caxman.cax
> caxman <cx1510.in >cx1510.out
> cp caxman.dat fort.8
> crfit8.x <cr1510.in >cr1510.out
> cp fort.9 cr1510.mat
```

2.2 Caxman8.f

Caxman on ohjelma, jolla voidaan lukea CASMO-ohjelmalla laskettuja vaikutusaloja ja kirjoittaa ne Crfit-ohjelman vaatimaan muotoon. Muutama yksikkömuunnos poislukien Caxman ei muokkaa lukemiaan arvoja, ainoastaan järjestää ne uudelleen. Caxman on siis vain apuohjelma, joka lukee CASMO:n tulostiedostoa ja poimii Crfitille tarpeelliset tiedot. Jotta kaikki tarvittavat kerroinlaskut tulevat tehtyä ja tarpeellinen tieto saadaan luettua, pitää CASMO:n syöttötiedosto olla tietyllä tavalla koostettu. Tarkemmat ohjeet on annettu luvussa 3.1.2.

Caxman8 on erityisesti kiehutusvesireaktorille kirjoitettu versio Caxmanista. Se lukee Crfit8:n tarvitsemat suuret CASMO:n tuottamasta *.cax-korttiedostosta ja kirjoittaa ne binääritiedostoon. Caxman8 tarvitsee erillisen syöttötiedoston, josta se lukee ainoastaan nominaaliarvot polttoaineen lämpötilalle T_{f0} , hidastimen lämpötilalle T_{c0} , joka on samalla jäähdytteen lämpötila, jäähdytteen aukko-osuuden V ja boorikonsentraation C_{B0} . Inputin tarkempi kuvaus on luvussa 3.1.3. Tässä työssä Caxman8-ohjelmaa muokattiin siten, että se pystyy lukemaan useamman säätösauvatyyppin arvoja samasta *.cax-tiedostosta. Muokattu versio on nimeltään Caxman8rods.

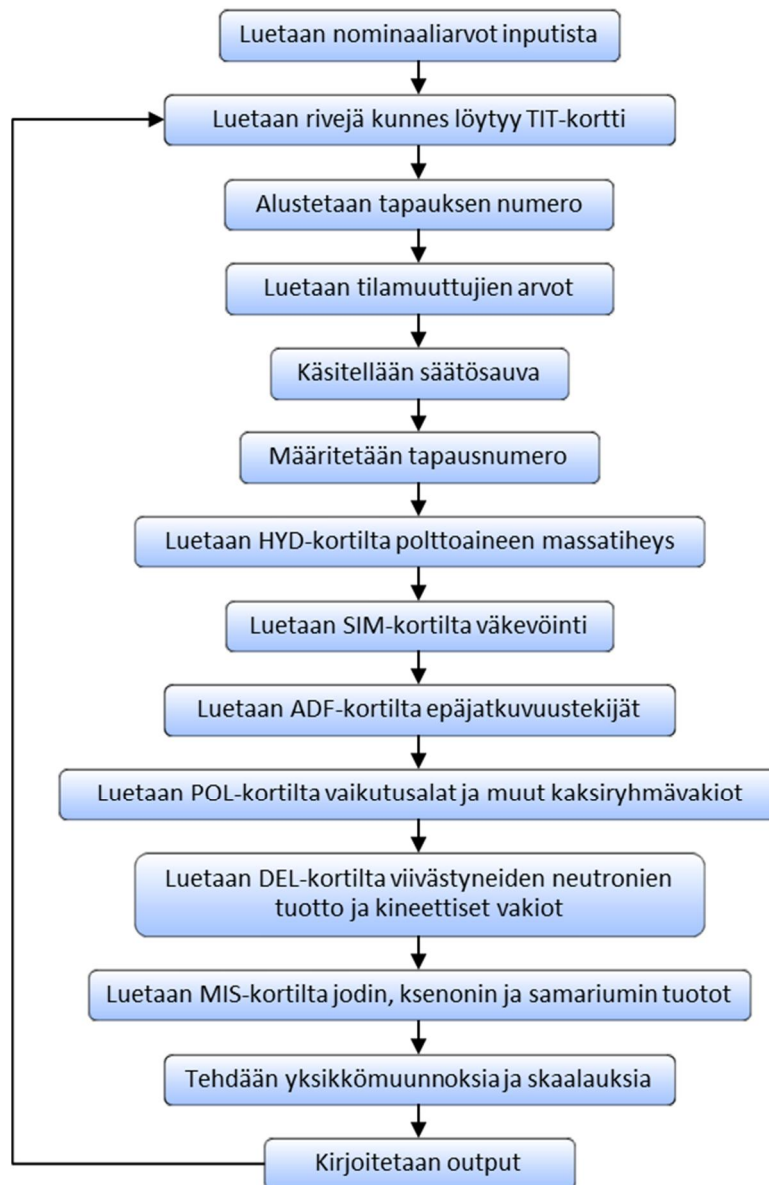
CASMO-4E:n tulostiedoston lukeminen tapahtuu yksityiskohtaisesti kuvan 1 mukaisesti. Kuvassa on esitetty caxman8rodsin pääohjelman toimintakaavio.

Aluksi luetaan syöttötiedostosta nominaalitalan arvot polttoaineen lämpötilalle yksikkönä [K], jäähdytteen lämpötilalle [K], jäähdytteen aukko-osuus [%] ja boorikonsentraatio [ppm]. Sen jälkeen luetaan CASMO:n tulostiedoston TIT-kortilta laskun otsikot ja tapausnumerointi, jolla merkitään, minkä muuttujan mukaan jaksotettuna tietoja luetaan. Seuraavaksi luetaan tilamuuttujien arvot. Tilamuuttujia ovat palama, aukko-osuus, aukkohistoria, polttoaineen ja jäähdytteen lämpötilat sekä boorikonsentraatio. Mikäli laskussa on mukana säätösauvoja, tallennetaan säätösauvan numero. Tässä työssä ohjelmaa muokattiin siten, että säätösauvatyyppiä voi olla useampia. Jokainen säätösauvatyyppin kerroinlasku tallennetaan eri säätösauvanumerolla. Seuraavaksi luetaan HYD-kortilta uraanin massatiheys pitoisuusyksikköä kohden [g/cm] ja SIM-kortilta uraanin väkevöinti [%]. ADF-kortilta luetaan epäjatkuvuustekijät nipun eri sivuilta ja POL-kortilta kaksiryhmäiset vaikutusalat:

- Diffuusiovakiot D_1, D_2
- Siirtymävaikutusala nopeasti termiseen ryhmään Σ_{12}
- Absorptiovaikutusalat Σ_{a1}, Σ_{a2}
- Neutronituotot $\nu\Sigma_1, \nu\Sigma_2$
- Xe:n ja Sm:n termisen ryhmän mikroskooppiset vaikutusalat $\sigma_{Xe,2}$ ja $\sigma_{Sm,2}$
- Fission energiantuotto κ
- Kasvutekijä k_∞

- Kupevuustekijä B_m^2
- Neutronien keskimääräinen lukumäärä fissiota kohti, ν_1 ja ν_2 , missä 1 on nopea ja 2 on terminen ryhmä.

Seuraavaksi luetaan DEL-kortilta viivästyneiden neutronien tuotot ja kaksiryhmäiset neutronien nopeuksien käänteisarvot. Lopuksi luetaan vielä MIS-kortilta jodin, ksenonin ja samariumin tuotot.



Kuva 1: Caxman8rodsin toimintakaavio.

Kun kaikki tiedot on luettu, suoritetaan yksikkömuunnokset, joissa lämpötilat muutetaan Celsius-asteiksi. Lisäksi lasketaan fissiovaikutusala neutronituoton ja ν :n avulla sekä ksenon- ja samariumvuot yhtälöstä [11, s.320]

$$\phi_X = \frac{\lambda_X}{\sigma_X}, \quad (1)$$

missä alaindeksi X on alkuaine Xe tai Sm, λ_X on alkuaineen hajoamisvakio ja σ mikroskooppinen vaikutusala.

Lopuksi luetut ja muunnetut tiedot kirjoitetaan output-tiedostoon ja palataan takaisin tilamuuttujien arvojen lukemiseen seuraavalla tapausnumerolla. Output-tiedoston tarkka kuvaus on annettu viitteessä [4].

2.3 Crfit8.f

Crfit on ohjelma, joka lukee caxmanin tuottamia binäärisiä tiedostoja ja laskee niistä nodaalikoodien vaatimat ryhmävakioiden parametrisoinnin kertoimet. Crfit8 on erityisesti kiehutusvesireaktoreille kirjoitettu versio ohjelmasta. Sovituksissa käytetyt tapaukset ja parametrit on esitetty taulukossa 3. Tilaparametrien arvot saadaan Caxman-ohjelman tuottamasta binääritiedostosta. Kertoimet lasketaan vain ohjelman syöttötiedostossa annetuille palamapisteille.

Taulukko 3: Sovituskertoimien laskennassa käytetty tapaukset ja niitä vastaavat parametrien muutokset [5]. Crfit8:ssa käytetyt tapaukset on korostettu sinisellä.

<i>Tapaus</i>	<i>Polttoaineen lämpötila</i>	<i>Jäähdytteen lämpötila</i>	<i>Jäähdytteen tiheys</i>	<i>Boori-konsentraatio</i>
0	T_f	T_c	n_c	C_B
1	$T_f + \Delta T_f$	T_c	n_c	C_B
2	$T_f - \Delta T_f$	T_c	n_c	C_B
3	T_f	T_c	n_c	0
4	T_f	$T_c + \Delta T_c$	$n_c - \Delta n_c$	0
5	T_f	$T_c - \Delta T_c$	$n_c + \Delta n_c$	0
6	T_f	T_c	$n_c + \Delta n_c$	0
7	T_f	T_c	$n_c - \Delta n_c$	0
8	T_f	$T_c + \Delta T_c$	n_c	C_B
9	T_f	$T_c - \Delta T_c$	$n_c + \Delta n_c$	$C_B + \Delta C_B$
10	T_f	$T_c + \Delta T_c$	$n_c - \Delta n_c$	$C_B + \Delta C_B$
11	T_f	T_c	n_c	$C_B + \Delta C_B$

Nodaalikoodissa tilaparametrien T_f , T_c , n_c ja C_B muutokset vaikutusaloihin toisen asteen polynomisovituksena. Merkitään tilaparametrin muutosta

$$\Delta x_i = x_i - x_0, \quad (2)$$

missä x_0 on tilaparametrin nominaaliarvo ja $x_{i=1,2}$ on kerroinlaskujen parametrin arvo. Näiden avulla vaikutusalat Σ sovitetaan tilaparametreihin kaavalla

$$\Delta \Sigma_i = c_1 \Delta x_i + c_2 (\Delta x_i)^2, \quad (3)$$

missä $\Delta \Sigma$ on vuorollaan kunkin tapauksen vaikutusalan poikkeama nominaaliarvostaan. Käytetyt vaikutusalat ovat nopean ja termisen ryhmän diffuusiokertoimet $D_{i=1,2}$ absorptiovaikutusalat Σ_{a1} ja Σ_{a2} , neutronintuottot $\nu \Sigma_{f1}$ ja $\nu \Sigma_{f2}$ sekä sirontavaikutusala nopeasta termiseen ryhmään Σ_{j2} . Tässä yhteydessä vaikutusalojen alaindekseissä käytetyt numerot 1 ja 2 viittaavat nopeaan ja termiseen ryhmään, vastaavasti. Laskettavat sovituskertoimet ovat c_1 ja c_2 .

Kahden kerroinlasku avulla voidaan kertoimet c_1 ja c_2 ratkaista yhtälöstä (3)

$$c_1 = \frac{\Sigma_1 - \Sigma_0}{x_1 - x_0} + \frac{x_1 - x_0}{x_1 - x_2} \left(\frac{\Sigma_2 - \Sigma_0}{x_2 - x_0} - \frac{\Sigma_1 - \Sigma_0}{x_1 - x_0} \right) \quad (4)$$

$$c_2 = \frac{\frac{\Sigma_2 - \Sigma_0}{x_2 - x_0} - \frac{\Sigma_1 - \Sigma_0}{x_1 - x_0}}{x_2 - x_1}. \quad (5)$$

Toisen asteen sovitetta käytetään polttoaineen lämpötilalle ja jäähdytteen tiheydelle. Jäähdytteen lämpötilalle käytetään ensimmäisen asteen yhtälöä, jolloin kerroin c_2 asetetaan nolaksi. Polttoaineen lämpötilan takaisinkytkentä lasketaan lämpötilan neliöjuuresta, jolloin $x_i = (T_{f,i})^{1/2}$ ja jäähdytteen tiheyden takaisinkytkentä lasketaan käyttäen diffuusiovakion käänteisarvoa $x_i = 1/D_{i=1,2}$.

Boorikonsentraation takaisinkytkennässä korjauksena otetaan huomioon myös boorikonsentraation riippuvuus hidasteen tiheydestä:

$$\Delta\Sigma = c_1\Delta x + c_2(\Delta x)^2 + c_{11}\gamma_B\Delta z \quad (6)$$

Yhtälössä (6) $\Delta x = \Delta\rho_B$ on booritiheyden ja $\Delta z = \Delta\rho_c$ hidastimen massatiheyden poikkeama nominaaliarvostaan. Tiheyksien välille oletetaan lineaarinen riippuvuus $\rho_B = C_B\rho_c$, missä C_B on hidastimen boorikonsentraatio. Koska hidastimen tiheyden muutos kytkeytyy sen lämpötilan muutokseen, oletetaan kaavan (6) sovituskertoimista c_{11} laskettaessa, että laskut tehdään yhtäaikaista myös hidastimen lämpötilalla. Tarkkaan ottaen mahdollinen boorikonsentraation vaikutus on voimassa vain silloin, kun käytetään yhdistettyä tiheyden ja lämpötilan kerroinlaskua. Boorikertoimia ei tosin lasketa kiehutusvesilaitokselle, mutta ohjelmassa ne on kuitenkin esitetty.

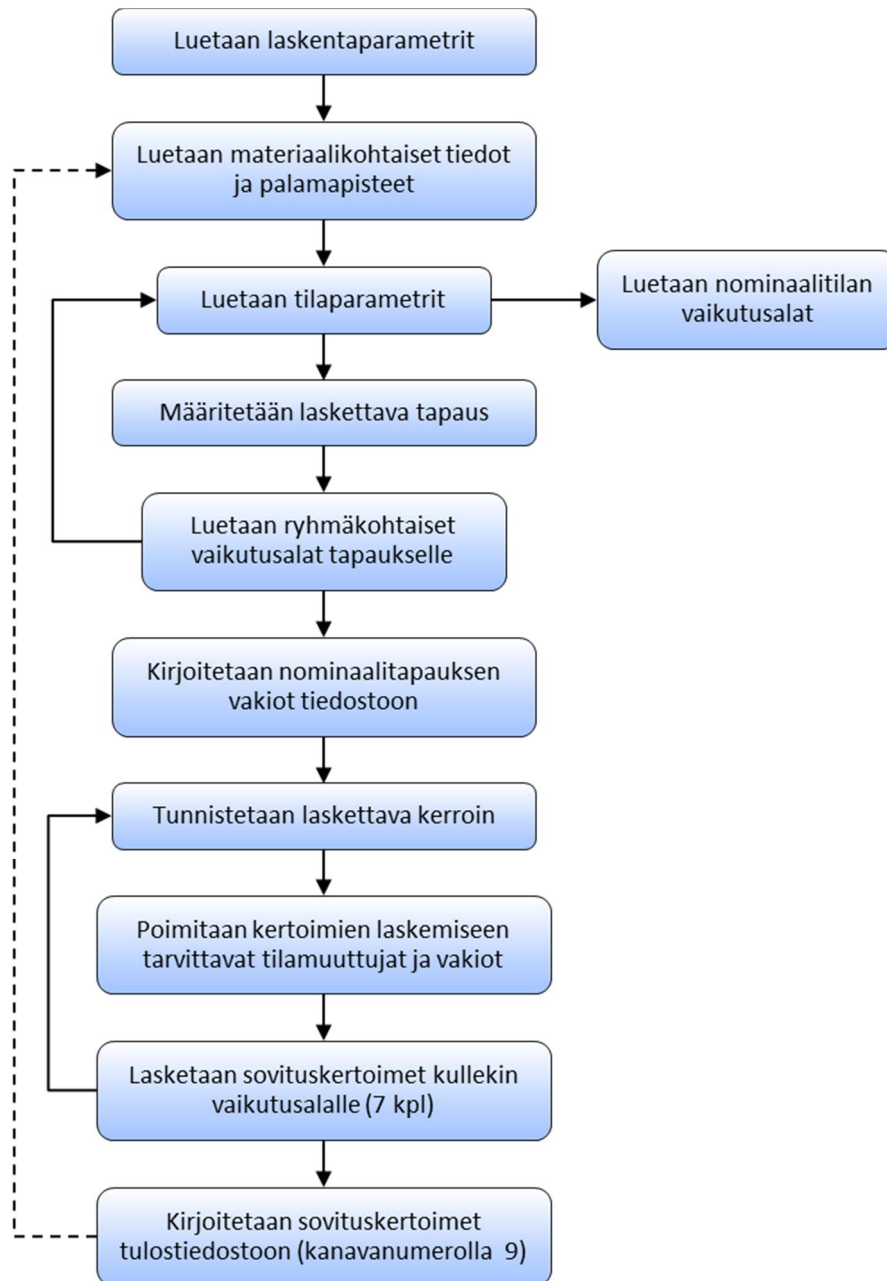
Crfit8-ohjelman toimintakaavio on annettu kuvassa 2. Aluksi ohjelma lukee Caxman-koodin kirjoittamasta tiedostosta laskentaparametrit sekä materiaaliakohtaiset tiedot: uraanin massa, polttoaineen lämpötila, väkevöinti, aukko-osuudet ja säätösauvatyyppit. Lisäksi luetaan kerroinlaskuissa käytettävät palamapisteet.

Seuraavaksi ohjelma poimii vaikutusalojen laskemista varten tilaparametrien arvot ja lukee kaksiryhmäiset vaikutusalat: kupevuus, kasvutekijä, diffuusio kertoimet, siirtymävaikutusala, absorptiovaikutusalat ja fissiotuotot. Lukemista jatketaan kunnes kaikki syöttötiedostossa annetut, taulukon 3 mukaiset tapaukset on luettu.

Seuraavaksi ohjelma kirjoittaa nominaalitalan vakiot tiedostoon, minkä jälkeen tunnistetaan laskettava kerroin. Laskettavalla kertoimella tarkoitetaan tässä yhteydessä doppler-, moderaattori- ja boorikertoimia. Moderaattorikertoimia ovat jäähdytteen lämpötila, tiheys sekä näiden yhteisvaikutus. Yhdistettyä lämpötilan ja tiheyden vaikutusta ei lasketa kiehutusvesireaktorille, kuten ei myöskään boorikertoimia.

Kun laskettava kerroin on tunnistettu, poimitaan kyseisen kertoimen laskemiseen tarvittavat vakiot, sekä tilaparametrit että kerroinlaskujen vaikutusalat. Sitten lasketaan näiden avulla sovituskertoimet c_1 ja c_2 kullekin seitsemälle vaikutusalalle. Lopuksi sovituksien tulokset kirjoitetaan outputtiin tarvittavassa muodossa.

Käyttäjän niin halutessa voidaan saatua ääretöntä kasvutekijää k_∞ verrata alkuperäiseen CAS-MOn kasvutekijään. Tätä varten on ohjelmassa oma aliohjelma, jota ei kuitenkaan ole kuvan 2 sisällytetty. Ero kirjoitetaan ohjelman tulostukseen, mutta siitä ei seuraa automaattisesti mitään toimenpiteitä.



Kuva 2: Crfit8:n pääohjelman toimintakaavio. Laskettava tapaus viittaa taulukossa 3 annettuihin tapauksiin. Laskettava kerroin on doppler-, moderaattori- tai boorikerroin.

3 Käyttöohje

3.1 Inputit

3.1.1 BWRXS-skriptin input

Skripti ottaa syöttöparametrina input-tiedoston nimen. Input-tiedosto on rakennettu korttiperi-aatteella. Rivin alussa annetaan kortin nimi tai tunnus, jonka perusteella skripti osaa lukea oikeat tiedot. Kortit ja luettavan tiedon kuvaus on annettu taulukossa 4. Tyypillisesti tarvittava tieto annetaan seuraavalla rivillä. Kuitenkin esim. luettavien parametrien lukumäärä annetaan samalla rivillä kortin nimen kanssa. Tästä syystä inputin formaatti on hyvin säännelty ja sen on syytä olla samanlainen kuin taulukossa 5 annettu esimerkki. Aukko-osuuksia vastaavat tiheydet on annettava aukkojen pareina, ja ne on laskettava tapauskohtaisesti kulloistakin morderaattorilämpötilaa vastaavasti. Tässä esimerkissä ne on laskettu arvolle $T_{co}=559$ K. Polttoaineen väkevöinti tarvitaan Crfitin [4,5] inputia varten, mutta sitä ei varsinaisesti käytetä.

Taulukko 4: BWRXS-skriptin inputin korttien kuvaus.

OTIT	työn otsikko
CKORTIT	kerroinlaskujen CASMO input (määrätty sisältö)
NIPUT	laskettavien tiedostojen määrä (arvo samalla rivillä), seuraavilla riveillä CASMO-tiedostojen nimet
CLIB	CASMO:n kirjasto
ROD	säätösauvojen määrä (arvo samalla rivillä)
HVOI	laskettavien aukkojen lkm (arvo samalla rivillä), seuraavilla riveillä pareittain aukko ja veden tiheys kullekin aukolle
ENR	nipun väkevöinti (arvo samalla rivillä)

Taulukko 5: Esimerkki BWRXS-skriptin syöttötiedostosta.

OTIT	BWR Stabiilisuusbenchmark, crfit8 20.6.2012
CKORTIT	crkortit2.inp
NIPUT	3 e15testi1.inp e15testi2.inp e15testi3.inp
CLIB	e41blb70
RODS	6
HVOI	3 0 0.739605 40 0.458426 80 0.177247
ENR	3.0

3.1.2 CASMO:n inputit

CASMO-inputeja tarvitaan kahta tyyppiä. Kullekin mallinnettavalle nipulle tarvitaan ns. perusnipputiedosto. Lisäksi tarvitaan yksi yhteinen kerroinlaskuinput. Näistä kahdesta inputista skripti yhdistää ja muokkaa paketin tarvitseman varsinaisen CASMO-inputin kullekin nipulle.

Perusnipputiedosto on aivan tavallinen peruspalamalasku ilman minkäänlaisia kerroinlaskuja. Se alkaa TTL-kortilla ja päättyy STA-korttiin, jota mahdollisesti seuraa vielä END-kortti. END-kortti ei ole välttämätön, mutta se voi olla mukana. Jos tiedosto on saatu valmiina, pitää kaikki kerroinlaskut poistaa. Lisäksi on huolehdittava, että TFU- ja TMO-kortit ovat omilla riveillään. VOI-kortti voi tiedostossa olla, mutta se korvataan paketin tarvitsemalla kortilla, joka on annettu taulukossa 1. Myös mahdolliset PUN ja WRI kortit korvataan. Jos näitä kolmea korttia (VOI, PUN ja WRI) ei ole, ne lisätään yhdistettyyn nipputiedostoon taulukossa 1 annetussa muodossa.

Perusnipputiedoston nimen on oltava sellainen, että siinä on kahden numeron mukainen ns. jaksoluku. Ensimmäinen kaksinumeroinen luku tulkitaan jaksoluvuksi ja sitä käytetään tapausten identifioivan luvun luomiseen. CASMO-inputin ns. nippuinputin ensimmäiset kaksi numeroa ovat nämä jaksonumerot. Nippuinputin kolmas numero otetaan perusnipputiedostojen järjestyksestä. Neljäs on aina 0. Perusnipputiedosto voi siis olla nimeltään esim. e15testi1.inp tai test32.inp. Näissä tapauksissa jaksoluku on 15 tai 32 ja nippua kuvaava luku vastaavasti 1510 tai 3210, kun kyseessä ovat ensimmäiset jakson tiedostot. Kolmas numero on järjestystä kuvaava numero ja neljäs 0. Näin jakson ensimmäisen nipun materiaaliluku päättyy aina lukuun 10, toisen lukuun 20, kolmannen lukuun 30 jne. Perusnipputiedoston nimessä ei suositella käytettäväksi sanaa ”*Nippu*”, koska skripti nimeää yhdistetyt nipputiedostot nimellä NippuXXXX.inp, missä XXXX on nipun identifioiva luku.

Kerroinlaskuinput voi olla minkä niminen tahansa, mutta samaa tiedostoa käytetään kaikille nipuille. Inputin muoto on sen sijaan hyvin tarkkaan säännelty. Esimerkki inputista on annettu taulukoissa 6 ja 7. Selvyyden vuoksi ne on tässä raportissa jaettu kahteen eri taulukkoon. Oikeasti niiden on oltava samassa tiedostossa.

Alussa oleva kerroinlasku Doppler-kertoimille määrittää kaikissa Doppler-kerroinlaskuissa käytettävät lämpötilat. Skripti lukee nämä kaksi lämpötilaa riviltä, jolla lukee Doppler. Muilla riveillä ei saa lukea sanaa Doppler. Kerroinlaskujen palama-arvot luetaan riviltä joka seuraa aukko-osuuksien kerroinlaskujen TTL-korttia. Tällä TTL rivillä lukee siis ”*BRANCHES*” ja palamat ovat seuraavalla rivillä. Jokaisen kerroinlaskun TTL-kortin pitää olla samanmuotoinen kuin näissä taulukoissa on annettu. Caxman määrittää näiden perusteella, mitä kerrointapaus-ta se käsittelee.

Tässä mallissa on muutamiin kerroinlaskuihin lisätty MI2-kortti. Tätä materiaalia ei käytetä mihinkään, mutta se on välttämätön tällä hetkellä käytössä olevilla ohjelmilla, koska CASMO:n nykyiset versiot eivät laske redundantteja tapauksia. Esimerkiksi VOID-kerroinlasku hetkellisellä aukko-osuudella 0 ja aukkohistorialla 0 jää nykyisellä versiolla laskematta, koska se on identtinen peruslaskun kanssa. Nykyiset versiot Caxmanista ja crffitista ovat kuitenkin riippuvaisia siitä, että myös nämä tapaukset lasketaan. Ne eivät osaa ottaa tietoa peruspalamalaskun osasta. MI2-kortilla CASMO ”huijataan” laskemaan myös nämä ns. redundantit tapaukset. Voisi olla järkevää päivittää Caxman ja Crfit toimimaan myös nykyisen CASMO-version kanssa ilman ylimääräisiä kortteja.

Taulukko 6: CASMO:n kerroinlaskujen inputin osa säätösauvattomille tapauksille.

<pre> *----- * Hot branches, all histories COE,,0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 * RES points TFU 559 975 *--- Doppler *----- </pre>	Doppler-kerroinlaskut kullekin HVOIDille
<pre> * Hot branches, base history TTL *+CRFIT TYPE 3 NOMINAL RES '/0001' 0 0.5 1.5 2.5 5 7.5 10 12.5 15 17.5 20 25 30 35 40 45 50 TMO 559.0001 VOI 40 STA </pre>	Nominaalitila VOID-kerroinlaskua varten HVOID=0
<pre> TTL *+CRFIT TYPE 6 7 VOID BRANCHES RES '/0001' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 TMO 559.0001 VOI 0,80 MI2 8.2 18.E-6 /347=100 STA </pre>	VOID-kerroinlaskut HVOID=0
<pre> TTL *+CRFIT TYPE 3 NOMINAL RES '/0002' 0 0.5 1.5 2.5 5 7.5 10 12.5 15 17.5 20 25 30 35 40 45 50 TMO 559.0001 VOI 40 MI2 8.2 18.E-6 /347=100 STA </pre>	Nominaalitila VOID-kerroinlaskua varten HVOID=40
<pre> TTL *+CRFIT TYPE 6 7 VOID BRANCHES RES '/0002' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 TMO 559.0001 VOI 0,80 STA </pre>	VOID-kerroinlaskut HVOID=40
<pre> TTL *+CRFIT TYPE 3 NOMINAL RES '/0003' 0 0.5 1.5 2.5 5 7.5 10 12.5 15 17.5 20 25 30 35 40 45 50 TMO 559.0001 VOI 40 STA </pre>	Nominaalitila VOID-kerroinlaskua varten HVOID=80
<pre> TTL *+CRFIT TYPE 6 7 VOID BRANCHES RES '/0003' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 TMO 559.0001 VOI 0,80 MI2 8.2 18.E-6 /347=100 STA </pre>	VOID-kerroinlaskut HVOID=80
<pre> TTL *+CRFIT TYPE 8 MOD TEMP RES '/0001' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 TMO 539.0001 COO .739605,,539.0001 MOD .739605,,539.0001 STA </pre>	TM-kerroinlaskut HVOI=0
<pre> TTL *+CRFIT TYPE 8 MOD TEMP RES '/0002' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 TMO 539.0001 COO .739605,,539.0001 MOD .739605,,539.0001 STA </pre>	TM-kerroinlaskut HVOI=40
<pre> TTL *+CRFIT TYPE 8 MOD TEMP RES '/0003' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 TMO 539.0001 COO .739605,,539.0001 MOD .739605,,539.0001 STA </pre>	TM-kerroinlaskut HVOI=80

Säätösauvalliset kerroinlaskut poikkeavat hieman sauvattomista tapauksista. Kullekin aukko-historialle lasketaan nominaalitila, joka on yhteinen sekä Doppler-kertoimille että aukko-kertoimille. Doppler-kertoimien laskut pitää näissä tapauksissa kuitenkin olla listattuna kullekin aukko-historialle. Aukkohistoriat määräytyvät RES-kortin ensimmäisen parametrin mukaan. Skripti huolehtii siitä, että nämä historiat lasketaan peruslaskussa, lisäämällä tarvittavan VOI-

kortin taulukon 1 mukaisesti. Säätösauvaosioita voi olla useampi, kun käytetään caxman8-rods-ohjelmaa. Tällöin kullekin perusnipputiedostossa määritellylle sauvalle tehdään oma osio, jossa ROD-kortilla spesifioidaan, mikä sauva on sisällä.

Taulukko 7: CASMOn kerroinlaskujen inputit säätösauvallisille tapauksille.

<pre> **-----** ** Ekaa säätösauvaa varten tarvittavat kortit ** **-----** TTL *+CRFIT TYPE 3 NOMINAL CONTROL ROD INSERTED RES '/0001' 0 0.5 1.5 2.5 5 7.5 10 12.5 15 17.5 20 25 30 35 40 45 50 ROD 'CRD' VOI 40 STA </pre>	Nominaalitila säätösauvalli- selle HVOI=0
<pre> TTL *+CRFIT TYPE 2 3 1 DOPPLER RES '/0001' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' TFU 559 775 975 STA </pre>	Doppler- kerroinlaskut HVOID=0
<pre> TTL *+CRFIT TYPE 6 7 VOID RES '/0001' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' VOI 0,80 STA </pre>	VOID-kerroin- laskut HVOID=0
<pre> TTL *+CRFIT TYPE 3 NOMINAL CONTROL ROD INSERTED RES '/0002' 0 0.5 1.5 2.5 5 7.5 10 12.5 15 17.5 20 25 30 35 40 45 50 ROD 'CRD' VOI 40 STA </pre>	Nominaalitila HVOID=40
<pre> TTL *+CRFIT TYPE 2 3 1 DOPPLER RES '/0002' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' TFU 559 775 975 STA </pre>	Doppler- kerroinlaskut HVOID=40
<pre> TTL *+CRFIT TYPE 6 7 VOID RES '/0002' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' VOI 0,80 STA </pre>	VOID-kerroin- laskut HVOID=40
<pre> TTL *+CRFIT TYPE 3 NOMINAL CONTROL ROD INSERTED RES '/0003' 0 0.5 1.5 2.5 5 7.5 10 12.5 15 17.5 20 25 30 35 40 45 50 ROD 'CRD' VOI 40 STA </pre>	Nominaalitila HVOID=80
<pre> TTL *+CRFIT TYPE 2 3 1 DOPPLER RES '/0003' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' TFU 559 775 975 STA </pre>	Doppler- kerroinlaskut HVOID=80
<pre> TTL *+CRFIT TYPE 6 7 VOID RES '/0003' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' VOI 0,80 STA </pre>	VOID-kerroin- laskut HVOID=80
<pre> TTL *+CRFIT TYPE 8 MOD TEMP RES '/0001' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' TMO 539 COO .739605,,539 MOD .739605,,539 STA </pre>	TM-kerroin- laskut HVOI=0

<pre>TTL *+CRFIT TYPE 8 MOD TEMP RES '/0002' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' TMO 539 COO .739605,,539 MOD .739605,,539 STA</pre>	TM-kerroin- laskut HVOI=40
<pre>TTL *+CRFIT TYPE 8 MOD TEMP RES '/0003' 0 0.5 1.5 2.5 5 7.5 10 15 20 30 40 50 ROD 'CRD' TMO 539 COO .739605,,539 MOD .739605,,539 STA</pre>	TM-kerroin- laskut HVOI=80

3.1.3 Caxman8n input

Caxmanin input on hyvin yksinkertainen. Siinä annetaan yhdellä rivillä nominaalitalan parametrit: polttoaineen ja moderaattorin lämpötilat T_{f0} ja T_{m0} , aukko-osuus V ja boorikonsentraatio C_{B0} . Lämpötilat annetaan Kelvineinä, aukko-osuus prosentteina ja boorikonsentraatio yksiköissä ppm. BWRXS-skripti tekee inputin nimellä cxXXXX.in, missä XXXX on nipun identifioiva luku. Esimerkki Caxman8n inputista on annettu taulukossa 8.

Taulukko 8: Caxmanin syöttötiedoston esimerkki

826.0	559.0	40.0	0.0
-------	-------	------	-----

Varsinaisen inputin lisäksi caxman tarvitsee CASMO:n korttitiedoston *.cax. Se pitää kopioida tai linkittää nimelle caxman.cax, josta caxman lukee tarvitsemansa tiedot. Skripti hoitaa kopioinnin automaattisesti.

3.1.4 Crfit8n inputit

Crfit8:n input on selkeästi monimutkaisempi. BWRXS-skripti tekee kullekin laskettavalle nipulle inputin automaattisesti. Esimerkki on annettu taulukossa 9.

Taulukko 9: Crfit8n inputin esimerkki yhdellä säätösauvalla.

<pre>'BWR Stabiilisuusbenchmark, crfit8 20.6.2012' 1 0 0 0 2 8 1 2 2 7*2 7*0 7*0 7*2 7*1 7*0 0.177247 0.458426 0.739605 286. 0.739605 0.0 0</pre>
<pre>1510 0.0 1 3.00 0.0 0.739605 0.739605 0.0 1 12 0. 500. 1500. 2500. 5000. 7500. 10000. 15000. 20000. 30000. 40000. 50000. -1/</pre>
<pre>1510 0.0 1 3.00 0.0 0.458426 0.458426 40.0 1 12 0. 500. 1500. 2500. 5000. 7500. 10000. 15000. 20000. 30000. 40000. 50000. -1/</pre>
<pre>1510 0.0 1 3.00 0.0 0.177247 0.177247 80.0 1 12 0. 500. 1500. 2500. 5000. 7500. 10000. 15000. 20000. 30000. 40000. 50000. -1/</pre>
<pre>1511 0.0 1 3.00 0.0 0.739605 0.739605 0.0 2 12 0. 500. 1500. 2500. 5000. 7500. 10000. 15000. 20000. 30000. 40000. 50000. -1/</pre>
<pre>1511 0.0 1 3.00 0.0 0.458426 0.458426 40.0 2 12 0. 500. 1500. 2500. 5000. 7500. 10000. 15000. 20000. 30000. 40000. 50000. -1/</pre>
<pre>1511 0.0 1 3.00 0.0 0.177247 0.177247 80.0 2 12 0. 500. 1500. 2500. 5000. 7500. 10000. 15000. 20000. 30000. 40000. 50000. -1/</pre>
<pre>0/</pre>

Ensimmäisellä rivillä on tapaukselle annettu otsikko. Toisella rivillä annetaan laskentaa ohjaavat parametrit. Ne on koottu taulukkoon 10.

Taulukko 10: Crfitin inputin toisen rivin parametrit.

	<i>Arvo</i>	<i>Kommentti</i>
1.	1	lasketaan parametrit tehollisina
2.	0	mielivaltainen crfit8lle, eli ei käytetä
3.	0	materiaalikupevuus data
4.	0	ei huomioida Samariumia Σ_{a2} :n laskennassa
5.	2	kirjoitetaan Xe ja Sm parametrit
6.	8	epäjatkuvuustekijöiden lkm (neljä per energiaryhmä)
7.	1	sisällytetään pistekineettiset parametrit

Kolmannella rivillä annetaan ensin Xe ja Sm parametrien lukumäärä (2 kummallekin). Jos tarvitaan vaikutusaloja Ksenon-transienttia varten, pitää Xe:lle käyttää arvoa 3 rivin ensimmäisenä parametrina. Sitten annetaan laskettavien sovituskertoimien määrät. Sovituskertoimet lasketaan seitsemälle vaikutusalalle D_1 , Σ_{a1} , $v\Sigma_{f1}$, Σ_{12} , D_2 , Σ_{a2} ja $v\Sigma_{f2}$. Siitä tulee tuo ”7*” minkä jälkeen annetaan sovituskertoimien määrät. Kertoimet lasketaan seuraaville parametreille: Doppler-kertoimille (2kpl), booritiheydelle (0 kpl), yhdistetylle jäähdytteen lämpötilalle ja tiheydelle (0 kpl), jäähdytteen tiheydelle (2kpl), jäähdytteen lämpötilalle (1kpl) ja materiaalikupevuudelle (0kpl). Suluissa on annettu esimerkissä oleva lukumäärä. Tämä rivi on sama, kuin HEXBU-3D:n inputissa käytettävä FITCO-kortti

Neljännellä rivillä annetaan jäähdytteeseen liittyviä parametreja. Ensinnä annetaan kolme kerroinlaskuissa käytettyä tiheyttä, alin, nominaali- ja ylin arvo. Sitten annetaan jäähdytteen lämpötila polttoainekanavan ulkopuolella [°C] sekä sitä vastaava tiheys. Nämä ovat yleensä samat kuin nominaaliarvot. Kuudes parametri on kanavan ulkopuolisen veden tilavuusosuus ja viimeinen indikoi moderaattorisovituksen tyyppiä.

Seuraavat kolme riviä menevät aina ryhmittäin. Ne on toistettava kullekin aukko-osuudelle eli jäähdytteen tiheydelle ja sitten myös kullekin säätösauvatapaukselle. Näistä ensimmäisen rivin parametrit on annettu taulukossa 11. Toisella rivillä annetaan ensin kerroinlaskujen palamapisteiden lukumäärä ja sitten yhtä monta palamapistettä. Palamapisteeset annetaan yksikössä MWd/tU ja niiden täytyy olla samat kuin Caxmanin tiedostossa. Kolmas rivi lopettaa aina blokin ja aloittaa uuden laskennan. Tämä rivi on ”-1”.

Taulukon 11 ensimmäinen parametri on materiaalinumero. Samaa numeroa käytetään identifioimaan polttoainetyyppi HEXBU:n MATER-kortilla. Skriptissä materiaalinumero muodostuu tuosta tiedostossa käytetystä numerosta siten, että neljäs numero kuvaa säätösauvatapauksia. 0 on perustapaus ilman säätösauvoja ja säätösauvalliset tapaukset menevät juoksevasti yhdestä ylöspäin.

Parametrit 6-9 ovat erityisesti Crfit8:n parametreja, joita tarvitaan poimimaan tarvittavat tiedot Caxmanin tuottamasta tiedostosta. Tiedostojen numeroinnista poiketen säätösauvan indeksi on 1 sauvattomalle tapaukselle ja kasvaa juoksevasti kullekin säätösauvatyyppille. Modifioitu caxman8rods päivittää kullekin säätösauvatapaukselle tätä arvoa, koska CASMO:n korttitiedostossa ei ole kyseistä arvoa. Näin tiedot löytyvät caxman8rodsilla tehdystä caxman.cax-tiedostosta. Jos säätösauvoja ei ole kuin yksi, voidaan käyttää myös muokkaamatonta caxman8aa.

Taulukko 11: Crfitin aukko-osuudelle eli jäähdytteen tiheydelle ja kullekin säätösauvalle annettavat parametrit.

<i>Paramaterin kuvaus</i>	<i>Kommentti</i>
1. materiaalinumero	
2. uranimassa	Ei käytetä crfit8:ssa, vaan luetaan caxmanin tuotamasta tiedostosta. Voidaan antaa arvo 0.
3. polttoaineen lämpötilalle käytetyn sovitteen tapausnumero	Löytyy taulukosta 3.
4. Polttoaineen väkevöinti	Ei käytetä crfit8:ssa, vaan luetaan caxmanin tuotamasta tiedostosta. Voidaan antaa noin arvo.
5. vanhentunut	Voidaan antaa esim. 0.0.
6. Nominaalitilan tiheyden historia-arvo	
7.	Pitää olla sama kuin edellä.
8. Aukko-osuuden historia-arvo	
9. Säätösauvan indeksi	

Crfit8ssa käytetään aukko-osuuden historia-arvoa sekä säätösauvan indeksiä määrittämään tietty tapaus. Jos crfitin tulostustiedostoon halutaan lisäksi vaikutusalojen vertailu CASMO:n ja crfitin välille, pitää crfitin inputiin laittaa kunkin blokin perään ennen blokin päättävää ”-1/” niiden kerrointausten numero, joille vertailu halutaan tehdä. Numero saadaan taulukosta 3 ja rivi päätetään merkillä ”/”.

3.2 Rajoitukset/vaatimukset

Periaatteessa BWRXS-skriptin pitäisi pystyä hoitamaan koko laskentaketju läpi. Se myös ottaa huomioon tietyt rajoitukset ja vaatimukset laskennalla. Taulukkoon 12 on kuitenkin koottu tärkeimpiä näistä vaatimuksista myös sitä silmällä pitäen, että laskentaa tehdään ilman skriptiä.

Taulukko 12: Rajoitukset tai vaatimukset sekä niiden ratkaisut

	<i>Ongelma/Vaatus</i>	<i>Ratkaisu</i>	<i>Ohjelma</i>
TFU ¹	Polttoaineen lämpötila pitää antaa kokonaislukuna	Skripti hoitaa pyöristyksen	CASMO, Caxman
Kerroinlaskukortit	Kerroinlaskut pitää tehdä tietyllä tavalla. Lisäksi kaikki laskut tulee tehdä.	Lisätty M12-kortti, niihin tapauksiin, jotka ovat identtisiä peruspalamalaskun kanssa	CASMO
Useampi säätösauva	CASMO:n korttitiedostossa ei numeroida säätösauvoja juoksevasti	Modifioitu caxman numeroi säätösauvat juoksevasti. Käytetään caxman8rodia	Caxman
Jäähdytteen tiheydet	Jäähdytteen tiheydet laskettava kaikille aukko-osuksille ennen paketin ajoa	Annetaan skriptin inputissa	Crfit
PUN-kortti	PUN kortti oltava muotoa PUN 6*0 1	Skripti korvaa mahdolliset PUN kortit	CASMO
VOI-kortti	Tarvitaan aukko-historiat 0 40 ja 80	Skripti hoitaa VOI-kortin tarvitsemaansa muotoon	CASMO
WRI-kortti	Suositellaan tulostuksen rajoittamiseksi WRI-kortille muotoa WRI 100/'OUT'	Skripti korvaa mahdolliset WRI-kortit	CASMO

¹ Johtunee liukulukualivuodosta tms., joten saattaa päteä myös muille lämpötiloille. Voisi olla järkevää päivittää caxman8 ja crfit8 niin, ettei tätä rajoitusta tarvita.

4 Yhteenveto

Tässä työssä on koottu laskentapaketti kiehutusvesilaitoksen vaikutusalojen laskemiseksi. Työssä erilliset ohjelmat on koottu yhteen helppokäyttöiseksi paketiksi. Sen tavoitteena on helpottaa ja yksinkertaistaa kiehutusvesilaitoksen materiaalivakioiden laskemista vähentämällä laskentaan tarvittavaa käsityötä. Täysin käsityötä ei voida poistaa, koska kullekin polttoainenipulle pitää aina tehdä CASMO:n [1,2] syöttötiedostot.

Työn kuluessa on kirjoitettu perl-skripti BWRXS.pl, joka toimii paketin pääohjelmana. Tämä skripti huolehtii laskentaketjun läpi viemisestä. Pakettiin kuuluvat CASMO:n lisäksi myös apuohjelmat caxman8 [3] ja crfit8 [4,5]. Näistä on tässä paketissa käytössä nimenomaan kiehutusvesilaitokselle tehdyt versiot. Caxman8 lukee CASMO:n tulostiedostosta tarvittavat tiedot ja kirjoittaa ne tiedostoon, jota crfit8 pystyy lukemaan. Crfit8 puolestaan laskee annetuille takaisinkytkentäkertoimille tarvittavat sovituskertoimet. Työn kuluessa caxman8 muokattiin siten, että se huomioi useamman säätösauvatyyppin käytön.

Paketti tarvitsee perustakseen CASMO:n syöttötiedoston kullekin laskettavalle nipulle sekä yhden kerroinlaskutiedoston. Näistä tehdään varsinainen ajettava syöttötiedosto. Syöttötiedostojen yksityiskohtaiset ohjeet rajoituksineen on annettu tässä raportissa. Apuohjelmien syöttötiedostot skripti tekee kullekin nipulle automaattisesti laskennan aikana. Niidenkin kuvaukset on kuitenkin annettu tässä raportissa, jotta ne tarvittaessa voidaan tehdä paketista riippumatta.

Laskentajärjestelmän lopputuloksena syntyy kullekin nipulle tiedosto, jossa on tarvittavat takaisinkytkentäkertoimien sovituskertoimet. Kertoimet on kirjoitettu ns. HEXBU-muodossa, jota käyttävät sekä sydänsimulaattori HEXBU-3D [6,7] että dynamiikkaohjelma TRAB-3D [8]. Myös APROS:n [9] uusi sydänmalli käyttää tätä muotoa materiaalivakioiden käsittelyssä.

Paketti on tällä hetkellä hyvin tarkka CASMO:n kerroinlaskuihin käytettävän syöttötiedostojen muodosta. Sen pitää olla tarkalleen tässä raportissa annetun muotoinen, jotta kaikki kerroinlaskut tulevat laskettua. Nykyinen versio CASMO:sta ei laske kerroinlaskua, jos se on identtinen peruslaskun kanssa. Koska apuohjelmat ovat riippuvaisia siitä, että kaikki laskut tehdään, on tässä paketissa annetussa kerroinlaskutiedostossa tavallaan huijattu ohjelma tekemään myös nämä laskut käyttämällä ylimääräistä MI2-korttia. Ylimääräisen kortin välttämiseksi pitäisi apuohjelmat kirjoittaa uudelleen, mitä ei kuitenkaan tämän työn puitteissa katsottu mahdolliseksi.

Paketti tuottaa ns. perinteiset vaikutusalat kolmessa dimensiossa. Systeemikoodit käyttävät usein yksinkertaisempaa aksiaalisesti yksidimensioista neutroniikkamallia, jolloin tarvitaan yksidimensioiset vakiot. Kondensointia ei kuitenkaan vielä tässä vaiheessa pakettiin liitetty, mutta tarvittaessa se lienee kuitenkin tehtävissä myöhemmin.

Lähdeviitteet

1. J. Rhodes and M. Edenius, "CASMO-4: A Fuel Assembly Burnup Program, User's Manual", Studsvik Scandpower Report SSP-01/400 Rev 4, 2004.
2. J. Rhodes, K. Smith and M. Edenius, "CASMO-4E: Extended Capability CASMO-4, User's Manual", Studsvik Scandpower Report SSP-01/401 Rev 2, 2004.
3. M. Anttila, "CAXMAN and related postprocessing codes. A provisional procedure to generate two-group diffusion parameters and fitting coefficients for HEXBU/ELSI calculations with the hexagonal version of the CASMO-4 fuel assembly program", VTT Energy, Technical report RDF-1/98, 1998.
4. E. Kaloinen, "CRFIT7&8, codes for creation of input file of two-group diffusion parameters and fitting coefficients for HEXBU-3D/MOD5, HEXTRAN and TRAB-3D," VTT Research Report No VTT-R-03225-06, VTT, 2006 12p.
5. E. Kaloinen, "CRFIT, A Code for Creation of Input File of Two-group Diffusion Parameters and Fitting Coefficients for HEXBU-3D/MOD5", VTT:n työraportti RFD-13/96, Espoo 1996.
6. E. Kaloinen, "HEXBU 3D/MOD5, a New Version of the HEXBU 3D Code", VTT Energy, Technical report RFD 26/92 (1992).
7. E. Kaloinen, "Input Instructions of HEXBU 3D/MOD5&6", VTT Processes, Project report PRO1/P1027/05 (2005).
8. E. Kaloinen and R. Kyrki-Rajamäki, "TRAB-3D, a new code for three-dimensional reactor dynamics", CD-ROM Proceedings of ICONE-5, 5th International Conference on Nuclear Engineering. "Nuclear Advances through Global Cooperation", May 26-30 1997, Nice, France, Paper ICONE5-2197.
9. E. Silvennoinen, K. Juslin, M. Hänninen, O. Tiihonen, J. Kurki & K. Porkholm, "The APROS Software for Process Simulation and Model Development", Research Reports 618, Technical Research Centre of Finland, Espoo 1989, ISBN 951-38-3463-8.
10. R. Kyrki-Rajamäki, "Three-dimensional reactor dynamics code for VVER type nuclear reactors", VTT Publications 246, Technical Research Centre of Finland, Espoo 1995. 51 p. + app. 80 p. Dr.Tech. Thesis.
11. J.R. Lamarsh, "Introduction to Nuclear Engineering", 2nd edition, Addison-Wesley 1983.

Liite A BWRXS.pl skripti

```

#! /usr/bin/perl
#
#-----#
# Tämä skripti tekee APROSille tai TRAB-3Dlle vaikutusalat HEXBU-3D #
# muodossa lähtien CASMO-inputeista. Se käyttää hyväkseen ohjelmia #
# caxman8 ja crfit8. Käytetyt ohjelmat määritellään skriptin alussa #
# kohdassa Ohjelmamäärittelyt. #
# Molemmat löytyvät hakemistosta /home/prokmr/bin/XSohjelmia #
# #
# Skripti ottaa syöttöparametrina input-tiedoston nimen. Input tie- #
# dostossa annetaan seuraavat kortit ja tieto seuraavalla rivillä: #
# OTIT - työn otsikko #
# CKORTIT - kerroinlaskujen CASMO-input (määrätty sisältö) #
# NIPUT - laskettavien tiedostojen määrä samalla rivillä #
# CASMO-tiedostojen nimet seuraavilla riveillä #
# CLIB - CASMON kirjasto #
# ROD - säätösauvojen määrä (arvo samalla rivillä) #
# HVOI - samalla rivillä laskettavien aukkojen lkm, #
# seuraavilla riveillä pareittain aukko ja veden #
# tiheys kullekin aukolle #
# ENR - nipun väkevöinti #
# #
# Esimerkki syöttötiedostosta: #
#-----#
# OTIT #
# BWR Stabiilisuusbenchmark, crfit8 20.6.2012 #
# CKORTIT #
# crkortit2.inp #
# NIPUT 3 #
# e15.p08.inp #
# e15.p10.inp #
# e15.red.inp #
# CLIB #
# e41blb70 #
# RODS 6 #
# HVOI 3 #
# 0 0.739605 #
# 40 0.458426 #
# 80 0.177247 #
# ENR 3.0 #
#-----#
# Skripti tekee annetuista tiedostoista CASMO-ajoa varten inputin, #
# joka nimetään NippuXXXX.inp. Kaksi ensimmäistä numeroa otetaan #
# tiedoston nimen alusta, kolmas numero otetaan tiedostojen järjes- #
# tyksestä. Näin em.tapauksessa inputit ovat Nippu1510.inp, #
# Nippu1520.inp ja ja Nippu1530.inp. Tästä syystä NIPUT-kortilla #
# annetuissa CASMO-tiedostoissa pitää olla esim. jaksoluku, jossa on #
# siis kaksi numeroa esim. e15testi.inp. #
# #
# Tiedostoista luetaan TFU, joka pyöristetään lähimpää kokonaislu- #
# kuun (korvataan). Kerroinlaskujen tiedostosta luetaan kerroinlas- #
# kujen alin ja ylin arvo. Nippu-teidostoon päivitetään nämä arvot #
# joka paikkaan. Myös kerroinlaskujen palamapisteiden arvot luetaan #
# tiedostosta. Mahdolliset PUN, VOI, WRI kortit korvataan laskujen #
# tarvitsemien mukaisilla korteilla ja perustiedostossa olevat STA #
# ja END kortit poistetaan, enne tiedostojen yhdistämistä. #
# Nippu-tiedostosta luetaan nominaalitilan TM, aukkoa ja boori. Tätä #
# varten VOI-kortilla pitää olla kerrottuna perustapauksen aukko #
# muodossa base=40%. Tästä syystä VOI-kortti korvataan. #
# CASMO-ajo ajetaan ja syntynyt cax-tiedosto linkitetään caxmania #
# varten nimelle caxman.cax #
# #
# CAXMANia varten kirjoitetaan nominaaliarvot syöttötiedostoon. Jos #
# on tarpeen ajaa kerralla useampi säätösauvatapaus pitää käyttää #
# modifioitua caxmania CAXMAN8RODS. Ajon jälkeen skripti huolehtii #
# tulostiedostojen kopioimisesta oikeille nimille. CAXMANin input #
# on nimeltään cxXXXX.in, eli esim. cx1510.in ja tulostiedosto vas- #
# taavasti cx1510.dat #
# #
# Lopuksi kirjoitetaan vielä CRFITiä varten input. Input on muotoa #
# crXXXX.in, missä tuo numero on sama kuin aiemmin. Materiaalinumero #
# muodostuu tuosta tiedostossa käytetystä numerosta siten, että nel- #
# jäs numero kuvaa säätösauva-tapauksia. 0 on perustapaus ilman #

```

```

# säätösauvoja ja säätösauvalliset tapaukset menevät juoksevasti yh- #
# estä ylöspäin. CRFITin ajon jälkeen kopioidaan tulokset perusta- #
# pauksen mukaisen numeron tiedostoon crXXXX.mat. Kaikki materiaali- #
# vakiot ovat siten *.mat-tiedostoissa. #
# #
# Skripti ei tarkasta missään vaiheessa ovatko ajot menneet onnistu- #
# neesti läpi. Elleivät ole menneet, ei myöskään materiaalivakiot #
# sisältävää *.mat-tiedostoa synny. Syntyviä tiedostoja ei poisteta #
# joten epäonnistumisen syytä pitäisi voida selvittää jälkikäteen. #
# Kun kaikki tiedostot ovat olemassa, pitäisi laskennan edetä #
# seuraavasti (esimerkkinä käytetty 3. e15-tapausta): #
# ----- #
# >> cas4e_lib.pl Nippu1530.inp e4l1blb70 #
# >> ln -s Nippu1530.cax caxman.cax #
# >> caxman8rods.x <cx1530.in >cx1530.out #
# >> cp caxman.dat fort.8 #
# >> crfit8.x <cr1530.in >cr1530.out #
# >> cp fort.9 cr1530.mat #
# ----- #
# #
# (c) Karin Rantamäki 2.7.2012 #
# #
#=====#

use strict;
use warnings;

#-----#
# Ohjelmamäärittelyt, eli mitä versioita käytetään #
my $path="/home/prokmr/bin/XSohjelmia/";
my $caxman=$path."caxman8rods.x";
my $crfit =$path."crfit8.x";
#-----#

my $Input = $ARGV[0];
if($#ARGV<0 || $Input eq 'h'){
  print "Anna inputfile\n";
  exit;
}
#-----#
# Muuttujamäärittelyt ja muuttujien alustus #
my (@niput,@hvoid,@dens,@bu);
my ($nippu,$ckortit,$cfile,$clib,$cxfile,$crfile,$tit,$enr);
my ($nrods);
my ($ii,$jj,$kk,$nn);
my (@items);
my ($Line,$TFLLine,$VOILine,$PUNLine,$WRILine);
my @Tfcoe;
my ($Tfnom,$Tmnom,$Vnom,$Bnom,$matnum,$jakso);
#-----#
# Nominaalitilan aukon määrittystä varten pitää aukko-rivillä #
# olle sana "base=", minkä perästä otetaan aukko #
#-----#
$jakso=0;
$Vnom=0;
$Bnom=0;
$VOILine=" VOI 0,40,80 \* History depletion void \((base=40%\))";
$PUNLine=" PUN 6\*0 1";
$WRILine=" WRI 100\/'OUT'";

#-----#
# Luetaan input-tiedostosta CRITin tarvitsemien CASMO-kortit sisältävä tiedosto #
# kohdasta CKORTIT sekä laskettavat nipputiedostot kohdasta NIPUT. NIPUT-sanan #
# perästä luetaan tiedostojen lukumäärä. Muuten tiedostojen nimet luetaan seu- #
# raavilta riveiltä. CLIB-kortilta luetaan käytettävä kirjasto ja RODS kortilta #
# laskettavien säätösauvojen määrä. HVOI kertoo laskettavien aukko-osuuksien #
# määrän ja reuraavilta riveiltä luetaan aukkoja vastaavat tieheydet. Lopuksi #
# luetaan ENR-kortilta väkevöinti. #
#-----#
open IN, "$Input" or die "Could not open file $Input: $! ";
$ii=0;
while ($Line=<IN>){
  chomp $Line; # Remove trailing newline
  $Line =~ s/s+//; # Remove space from beginning of line
  if($Line=~/^OTIT/){
    $tit=<IN>;
    chomp $tit; # Remove trailing newline
    $tit =~ s/^\s+//; # Remove space from beginning of line
    print "*****\n";
  }
}

```

```

    print "* $tit\n";
    print "*****\n";
  }
  elseif($Line=~/^KORTIT/){
    $Line=<IN>;
    chomp $Line;          # Remove trailing newline
    $Line =~ s/^\s+//;    # Remove space from beginning of line
    $kortit=$Line;
    print " CRFIT8n vaatimat lisäkortit tiedostossa $kortit!\n";
  }
  elseif($Line=~/^NIPUT/){
    chomp $Line;          # Remove trailing newline
    $Line =~ s/^\s+//;    # Remove space from beginning of line
    @items = split /\s+/, $Line;
    # print $Line, "\n";
    $nn=$items[1];
    print "Lasketaan $nn nippua\n";
    #-----#
    # Luetaan nippujen CASMO-tiedostojen nimet #
    for (my $jj=0;$jj<$nn;$jj++){
      $Line=<IN>;
      chomp $Line;          # Remove trailing newline
      $Line =~ s/^\s+//;    # Remove space from beginning of line
      $ninput[$jj]=$Line;
      printf " %i. CASMO-tiedosto on $ninput[$jj]\n", $jj+1;
    } # end for jj=0->nn
  } # end it $Line=s / ^NIPUT
  elseif ($Line=~/^CLIB/) {
    $Line=<IN>;
    chomp $Line;          # Remove trailing newline
    $Line =~ s/^\s+//;    # Remove space from beginning of line
    $clib=$Line;
    print "Kirjasto on $clib\n";
  }
  elseif ($Line =~ /^RODS/){
    chomp $Line;          # Remove trailing newline
    $Line =~ s/^\s+//;    # Remove space from beginning of line
    @items = split /\s+/, $Line;
    # print $Line, "\n";
    $nrods=$items[1];
    print "Lasketaan $nrods säätösauvaa\n";
  }
  elseif ($Line =~ /^HVOI/){
    chomp $Line;          # Remove trailing newline
    $Line =~ s/^\s+//;    # Remove space from beginning of line
    @items = split /\s+/, $Line;
    # print $Line, "\n";
    $nn=$items[1];
    print "Lasketaan $nn aukkoa\n";
    #-----#
    # Luetaan nippujen CASMO-tiedostojen nimet #
    for (my $jj=0;$jj<$nn;$jj++){
      $Line=<IN>;
      chomp $Line;          # Remove trailing newline
      $Line =~ s/^\s+//;    # Remove space from beginning of line

      @items = split /\s+/, $Line;
      # print $Line, "\n";
      $hvoid[$jj]=$items[0];
      $dens[$jj]=$items[1];
      printf " %i. aukolla $hvoid[$jj] tiheys on $dens[$jj]\n", $jj+1;
    } # end for jj=0->nn
  }
  elseif ($Line =~ /^ENR/){
    chomp $Line;          # Remove trailing newline
    $Line =~ s/^\s+//;    # Remove space from beginning of line
    @items = split /\s+/, $Line;
    # print $Line, "\n";
    $enr=$items[1];
    print "Väkevointi on $enr\n";
  }
} # end while
close IN;

```

```

#-----#
# Input-tiedosto on luettu ja laskenta-tapaukset tiedetään. #
#-----#
# Käsitellään kukin annetuista nipputapauksista. Luetaan perusinputista nomi- #
# naali Tf ja kerrointiedostosta laskujen alin ja ylin T. Yhdistetään tiedostot #
# materiaalinumeron mukaiseksi inputiksi ja päivitetään tarvittavat arvot. #
# Ajetaan CASMO, päivitetään tarvittavat inputit CAXMANia varten ja ajetaan se. #
# Päivitetään inputit CRFITiä varten ja ajetaan se. #
#-----#
foreach my $case (@niput) {
  print "\nKäsittelen tiedostoa $case!\n";
  #-----#
  # Lue perusinputista nominaalilämpötila #
  #-----#
  $Line = `grep TFU $case`;
  chomp $Line; # Remove trailing newline
  $Line =~ s/^\s+//; # Remove space from beginning of line
  @items = split /(=\s)\s*/, $Line;
  $Tfnom= sprintf "%.0f",$items[2]+0.5;
  print "Tfnom on $Tfnom\n";
  $TFLine="TFU $Tfnom";
  #-----#
  # Lue kerrointiedostosta kerroinlaskujen alin ja ylin lämpötila #
  #-----#
  $Line = `grep Doppler $ckortit`;
  chomp $Line; # Remove trailing newline
  $Line =~ s/^\s+//; # Remove space from beginning of line
  @items = split /\s+/, $Line;
  @Tfcoe=@items[1,2];
  #-----#
  # Luetaan kerrointiedostosta kerroinlaskujen palama-arvot #
  #-----#
  @items = `grep -ml -Al BRANCHES $ckortit`;
  @bu = split /\s+/, $items[1];
  shift @bu;
  shift @bu;
  printf "Käsitellään %d pistettä\n", $#bu+1;
  #-----#
  # Määritä materiaalinumero ja sen mukainen input-tiedosto #
  #-----#
  my $x=$case;
  $x =~ /(\d\d)/;
  $matnum=$1;
  if($matnum != $jakso){
    $ii=0;
    $jakso=$matnum;
  }
  $ii++;
  $matnum= $jakso*100+10*$ii;
  print "Materiaali on $matnum\n";
  # print "Numero on $matnum \n";
  $cfile='Nippu'. $matnum.'.inp';
  # print "CASMO-tiedosto on $cfile\n";
  `cp $case $cfile`;
  #-----#
  # Korvataan TFU-kortti pyöristetyllä arvolla. Poistetaan VOI-, #
  # PUN-, WRI-, STA- ja END-kortit CASMON input-tiedostosta. #
  # Kirjoitetaan tiedoston perään uudet VOI-, PUN-, WRI- ja STA- #
  # kortit. Liitetään vielä loppuun kerroinlaskujen kortit #
  # $ckortit-tiedostosta. #
  #-----#
  $Line = `grep TFU $cfile`;
  chomp $Line;
  system("perl -i -p -e 's/$Line/$TFLine/' $cfile");

  print "Poistetaan VOI,PUN,WRI,STA ja END kortit\n";
  `grep -v "PUN " $cfile | grep -v "WRI " | grep -v STA | grep -v END | grep -v VOI > foo `;
  `mv foo $cfile` ;

  open FILE1, ">>$cfile" or die "Could not open file $cfile: $! ";
  open FILE2, "$ckortit" or die "Could not open file $ckortit: $! ";
  print FILE1 $VOILine, "\n";
  print FILE1 $PUNLine, "\n";
  print FILE1 $WRILine, "\n";
  print FILE1 " STA\n";

  print FILE1 "*-----\n";
  print FILE1 "* Yhdistetään CRFITin tarvitsemat kortit\n";
  # print FILE1 "*-----\n";

```

```

while ($Line=<FILE2>){
  if($Line=~/^\\s*TFU\\s+(\\d{3,4}(\\s+|\\.\\d*))}{3}/) {
#     print " 3 TFU-lukua rivillä $Line";
    $Line= "TFU $Tfcoe[0] $Tfnom $Tfcoe[1] \\n";
#     print "uusi rivi on $Line\\n";
  } # end if $Line =~ TFU
print FILE1 $Line;
} # end while FILE2
close FILE1;
close FILE2;
#-----#
# Selvitetään CASMO-inputista nominaalitilan TM, aukko ja boori #
# Nominaalitilan aukon määrittystä varten pitää aukko-rivillä #
# olle sana "base=", minkä perästä otetaan aukko #
#-----#
$Line=`grep -m 1 TMO $cfile`;
chomp $Line; # Remove trailing newline
$Line =~ s/^\\s+//; # Remove space from beginning of line
@items= split /(=|\\s)\\s*/, $Line;
$Tmnom=$items[2];
print "Nominaalitilan TMO on $Tmnom, ";

$Line=`grep -m 1 base= $cfile`;
if($Line =~ /VOI/) {
  $Line =~ /=(\\d{2,3})/;
  $Vnom=$1;
} # end if Line =~ VOI
print "aukko on $Vnom";

$Line=`grep -m 1 BOR $cfile`;
if($Line =~ /BOR/) {
  $Line =~ / (\\d{2,4})/;
  $Bnom=$1;
} # end if Line =~ BOR
print " ja CB on $Bnom\\n";
#-----#
# Ajetaan CASMO halutulla kirjastolla ja linkitetään syntynyt #
# .cax tiedosto caxmania varten nimelle caxman.cax #
#-----#
# print ("cas4e_lib.pl $cfile $clib\\n");
print "Ajetaan CASMO tiedostolla $cfile\\n";
system("cas4e_lib.pl $cfile $clib >cas.out");
$cfile =~ s/.inp/.cax/;
`rm caxman.cax` if (-e "caxman.cax");
`ln -s $cfile caxman.cax`;
#-----#
# Kirjoitetaan nominaalitapauksen tiedot caxmanin inputiin ja #
# ajetaan caxman. Kopioidaan syntynyt .dat tiedosto sekä mate- #
# riaalinumeron mukaiseksi että crfitin tarvitsemaksi unit 8:n #
# tiedostoksi. Ahjolla fort.8. #
#-----#
open FILE, ">cx$matnum.in" or die "Could not open file cx$matnum.in: $! ";
printf FILE " %6.1f %6.1f %6.1f %6.1f \\n", $Tfnom, $Tmnom, $Vnom, $Bnom;
close FILE;
# print ("$caxman <cx$matnum.in >cx$matnum.out\\n");
print ("Ajetaan caxman");
system("$caxman <cx$matnum.in >cx$matnum.out ");
`cp caxman.dat cx$matnum.dat`;
`cp caxman.dat fort.8`;
#-----#
# Tee input crfitiä varten #
#-----#
$crfile="cr".$matnum;
open FILE, ">$crfile.in" or die "Could not open file $crfile.in: $! ";
print FILE "'$tit'\\n";
print FILE " 1 0 0 0 2 8 1 \\n";
#-----#
# Tf Cb (d+T)m dm Tm Bm #
print FILE " 2 2 7*2 7*0 7*0 7*2 7*1 7*0 \\n";
#-----#
printf FILE " %8.6f %8.6f %8.6f 286. $dens[0] 0.0 0\\n", $dens[2], $dens[1], $dens[0];
foreach ($kk=0;$kk<=$nrods;$kk++){
  foreach ($jj=0; $jj<@hvoid;$jj++){
    printf FILE "%5d 0.0 1 %5.2f 0.0 %8.6f %8.6f %5.1f %1d \\n",
      $matnum,$enr,$dens[$jj],$dens[$jj],$hvoid[$jj],$kk+1;
    printf FILE "%3d ", $#bu+1;
    foreach my $bu (@bu) {
      printf FILE "%5d. ", $bu*1000;
    } # End foreach bu
  }
}

```

```
        print FILE "\n";
        print FILE " -1/\n";
    } # end foreach @hvoid
    $matnum++;
} # end foreach $nrods
print FILE " 0/\n";
close FILE;
#-----#
# Aja CRFIT crfit8.x <$crfile.in >$crfile.out ja kopioi      #
# tulokset materiaalinumeron mukaiseen .mat -tiedostoon    #
#-----#
#   print (" $crfit <$crfile.in >$crfile.out\n");
#   print " ja Crfit \n";
#   system("$crfit <$crfile.in >$crfile.out ");
#   `cp fort.9 $crfile.mat`;
#   print "!-----\n";
#   print "! Vakiot tiedostossa $crfile.mat\n";
#   print "!-----\n";
} # end foreach case
#=====#
```